

Simultaneous Estimation of Pose and Motion at Highly Dynamic Turn Maneuvers

Alexander Barth¹, Jan Siegemund², Uwe Franke¹, and Wolfgang Förstner²

¹ Daimler AG, Group Research and Advanced Engineering,
Sindelfingen, Germany

² University of Bonn, Department of Photogrammetry,
Institute of Geodesy and Geoinformation, Bonn, Germany

Abstract. The (Extended) Kalman filter has been established as a standard method for object tracking. While a constraining motion model stabilizes the tracking results given noisy measurements, it limits the ability to follow an object in non-modeled maneuvers. In the context of a stereo-vision based vehicle tracking approach, we propose and compare three different strategies to automatically adapt the dynamics of the filter to the dynamics of the object. These strategies include an IMM-based multi-filter setup, an extension of the motion model considering higher order terms, as well as the adaptive parametrization of the filter variances using an independent maximum likelihood estimator. For evaluation, various recorded real world trajectories and simulated maneuvers, including skidding, are used. The experimental results show significant improvements in the simultaneous estimation of pose and motion.

1 Introduction

Detecting and tracking other traffic participants accurately is an essential task of future *intelligent vehicles*. Recent driver assistance and safety systems, such as the Adaptive Cruise Control (ACC) system, focus on tracking the leading vehicle in scenarios with relative low dynamics in lateral direction. Future collision avoidance systems, however, must be able to also track the oncoming traffic and to deal with a wide range of driving maneuvers, including turn maneuvers.

The complex dynamics of a vehicle are usually approximated by simpler motion models, e.g. constant longitudinal velocity and constant yaw rate (angular velocity) models [1] [2], or constant longitudinal acceleration and constant yaw rate models [3] [4]. These models are special cases of the well-known bicycle model [5] and restrict lateral movements to circular path motion since vehicles can not move sideward. Higher order derivatives, such as yaw acceleration, are modelled as zero-mean Gaussian noise.

At turn maneuvers, however, the yaw acceleration becomes a significant issue. Vehicles quickly develop a yaw rate if turning left or right at an intersection. A single (Kalman) filter, parametrized to yield smooth tracking results for mainly longitudinal movements, is often too slow to follow in such situations (see Fig. 1).

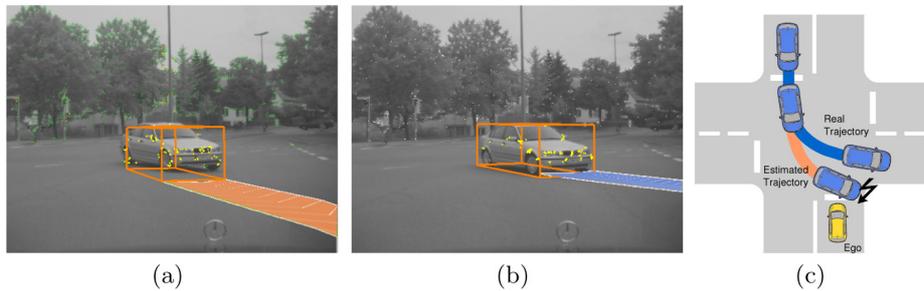


Fig. 1. (a) The filter can not follow a turning vehicle if parametrized for mainly longitudinal movements. (b) The same filter parametrized for turn maneuvers allows for accurate tracking. (c) Bird's eye view on the problem at hand.

On the other hand, making the filter more reactive in general increases the sensitivity to noise and outliers in the measurements.

Multi-filter approaches, such as the Interacting Multiple Models (IMM) method proposed by Bar-Shalom [6], tackle this problem by running several filters with different motion models in parallel. Kaempchen et al. [3] successfully applied the IMM framework to track leading vehicles in Stop&Go scenarios.

Alternative methods try to estimate the higher order terms, modelled as system noise, by an additional detector outside the filter. Chan et al. [7] introduced an input estimation scheme that computes the maximum likelihood (constant) acceleration of a moving target over a sliding window of observations, using least squares estimation techniques. The estimated acceleration is used to update the constant velocity model using the Kalman filter control vector mechanism. The drawback of this approach is that the control vector has a direct influence on the state estimate. Thus, errors in the input data directly affect the state estimate.

In [4], a stereo-vision based approach for tracking vehicles by means of a rigid 3D point cloud with a single Extended Kalman Filter (EKF) has been proposed. This approach yields promising results both for oncoming and leading vehicles at a variety of scenarios with mainly longitudinal movements and relatively slow turn rates. However, in general, approaches utilizing a single filter configuration suffer from the fact that the filter parametrization strongly depends on the expected dynamics.

In this contribution, we will extend the approach proposed in [4] and present three different solutions overcoming the problem of manual situation-dependent parameter tuning.

2 Object Model

An object is represented as rigid 3D point cloud attached to a local *object coordinate system*. The Z-axis of this system corresponds to the moving direction, i.e. the longitudinal axis for vehicles. The X- and Y-axis represent the lateral and

height axis respectively. The origin of the object coordinate system is defined at the center rear axis on the ground. In the same way, the *ego coordinate system* is defined for the ego vehicle. The following state vector is estimated:

$$\mathbf{x} = \left[\underbrace{{}^e X_0, {}^e Z_0, \psi}_{\Omega \hat{=} pose}, \underbrace{v, \xi, \dot{v}}_{\Phi \hat{=} motion}, \underbrace{{}^o X_1, {}^o Y_1, {}^o Z_1, \dots, {}^o X_M, {}^o Y_M, {}^o Z_M}_{\Theta \hat{=} shape} \right]^T \quad (1)$$

where the object pose w.r.t. the ego vehicle is given by position $[{}^e X_0, 0, {}^e Z_0]^T$, i.e. the object origin in the ego system, and orientation ψ . The motion parameters include the absolute velocity v and acceleration \dot{v} in longitudinal direction as well as the yaw rate ξ . Finally, ${}^o P_m = [{}^o X_m, {}^o Y_m, {}^o Z_m]^T$, $1 \leq m \leq M$, denote the coordinates of a point instance \mathcal{P}_m within the local object coordinate system. So the filter does not only estimate the pose and motion state, it also refines the 3D point cloud representing the object's shape. This is a significant difference to the original approach in [4], where the point positions have been estimated outside the filter due to real-time constraints. Furthermore, opposed to [4], the position of the center rear axis of a detected vehicle is assumed to be known sufficiently well at initialization for simplicity.

System Model: The following nonlinear equations define the system model (motion model) in time-discrete form:

$$\Delta \mathbf{x}(t) = \begin{bmatrix} \Delta {}^e X_0 \\ \Delta {}^e Z_0 \\ \Delta \psi \\ \Delta v \\ \Delta \xi \\ \Delta \dot{v} \\ \Delta {}^o P_1 \\ \vdots \\ \Delta {}^o P_M \end{bmatrix} = \begin{bmatrix} \int_t^{t+\Delta T} -v(\tau) \sin(\psi(\tau)) d\tau \\ \int_t^{t+\Delta T} v(\tau) \cos(\psi(\tau)) d\tau \\ \xi \Delta T \\ \dot{v} \Delta T \\ 0 \\ 0 \\ [0, 0, 0]^T \\ \vdots \\ [0, 0, 0]^T \end{bmatrix} \quad (2)$$

The scalar ΔT indicates the discrete time interval between current and last time step. The higher order terms are modelled as normal distributed noise, i.e. $\mathbf{x}(t + \Delta T) = \mathbf{x}(t) + \Delta \mathbf{x}(t) + \mathbf{w}(t)$ with $\mathbf{w}(t) \sim \mathcal{N}(0, \mathbf{Q}(t))$. The system noise matrix \mathbf{Q} , with $\mathbf{Q}(t) = \text{diag}(\sigma_{X_0}^2(t), \sigma_{Z_0}^2(t), \sigma_{\psi}^2(t), \dots, \sigma_{Z_M}^2(t))$, controls the reactivity of the filter. In the original approach, a constant system matrix is used for all time steps. For simplicity, we assume a stationary camera throughout this article, which is equivalent to an ideally ego-motion compensated scene, for example using the method proposed by Badino [8]. In practice, the estimated errors of the ego-motion have to be integrated into the system noise matrix.

Measurement Model: The measurement vector is of the form

$$\mathbf{z}(t) = [u_1(t), v_1(t), d_1(t), \dots, u_M(t), v_M(t), d_M(t)]^T \quad (3)$$

where $(u_m(t), v_m(t))$ represents a sub-pixel accurate image position of a given object point \mathcal{P}_m projected onto the image plane at time step t , and $d_m(t)$ the

stereo disparity at this position. The image positions are tracked using a feature tracker, e.g. [9], to be able to reassign measurements to the same object point over time. The nonlinear measurement equations used for measurement prediction directly follow from the projection equations (see [4] for details).

3 Filter Approaches

3.1 Interacting Multiple Model (IMM)

A set of r filters builds the basis of the IMM framework. Each filter represents a certain *mode*, e.g. non-maneuvering or maneuvering. One IMM filter cycle consists of three main parts: First, the *a posteriori* state estimates and covariance matrices of the previous discrete time step are *mixed* using a weighted sum (interaction step). Then, the filters are updated based on a common measurement vector (filtering step). Finally, a mode probability is computed for each filter (mode probability update step). The normalized residual, i.e. the deviation between prediction and measurements in terms of standard deviations, is used as indicator for mode likelihood. Furthermore, *a priori* mode transition probabilities must be provided. The weighting coefficients in the mixing step are derived from the mode likelihood and mode transition probability. Details can be found in [6].

We employ a two filter approach. Both filters use the same object, measurement, and motion model as proposed in Sec. 2. The different behaviour of the non-maneuvering filter ($\hat{=}$ mode 1) and maneuvering filter ($\hat{=}$ mode 2) is configured via the system noise matrices only, denoted as Q_{stat} and Q_{mnv} , respectively. It is possible to parametrize the system matrices in a way that the non-maneuvering filter corresponds to a constant acceleration, constant yaw rate motion model, while the maneuvering filter allows for larger changes in yaw rate and acceleration (see Sec. 4.1).

3.2 Oracle Measurements and Adaptive Noise (EKF-OR)

Given a set of k measurement vectors $\tilde{\mathbf{z}} = \{\mathbf{z}(T), \mathbf{z}(T - 1), \dots, \mathbf{z}(T - (k - 1))\}$, where T denotes the current discrete time step, we can estimate

$$\mathbf{y} = [\Theta, \Omega(T), \Omega(T - 1), \dots, \Omega(T - (k - 1))]^\top, \tag{4}$$

i.e. the shape parameters and a set of object poses, via a maximum likelihood estimation $\hat{\mathbf{y}} = \mathit{argmax}_{\mathbf{y}}(p(\tilde{\mathbf{z}}|\mathbf{y}))$, with $\hat{\mathbf{y}}$ denoting the estimated parameters. The idea of this approach is, using the known camera geometry, we directly obtain ${}^eP_m(t)$, i.e. the coordinates of the observed object points in the ego coordinate system at a given time step. From these point coordinates and the corresponding coordinates in the object system, we can derive the six parameter similarity transformation (no scale) between the ego and object system, which gives the object's pose (see [10]). Assuming the object point cloud to be rigid over time, we can simultaneously estimate the object point cloud (shape) and

the object poses in a least squares sense. As we assume Gaussian noise, this is equivalent to a maximum likelihood estimation.

In addition, the motion parameters can be easily derived from the estimated poses by numerical differentiation. We will refer to the estimation process outside the Kalman filter as *oracle* in the following. The oracle is purely data driven and not constrained by a motion model. Thus, the oracle approach is able to follow all movements describable in terms of a 3D rotation and translation.

The yaw rate estimate of the oracle $\hat{\xi}_{OR}$ gives a strong evidence for maneuvers. At non-maneuvering periods the yaw rate is assumed to be low in typical driving scenarios and may be constant for a longer time interval. On the other hand, a large yaw rate is very likely to change within a short time interval, since vehicles usually do not drive in a circle for very long. This leads to the idea of steering the system noise level based on the magnitude of the yaw rate estimate of the oracle (using the oracle's yaw acceleration estimate instead of the yaw rate turned out to be too noisy as reliable maneuver detector).

The yaw rate entry in the system noise matrix at a given time step t is set to $\Lambda(\hat{\xi}_{OR}(t))$, where Λ is a sigmoidal function that smoothly interpolates between a minimum and maximum noise level σ_{min} and σ_{max} , respectively, as can be seen in Fig. 2.

Next, we pass the yaw rate estimate of the oracle to the Kalman filter as additional measurement. The measurement vector $z(t)$ is extended to $z'(t) = [z(t), \hat{\xi}_{OR}]^T$. The measurement noise for the yaw rate estimate can be derived from the uncertainties of the estimated oracle poses.

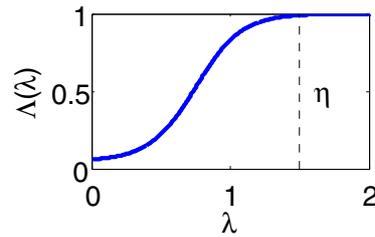


Fig. 2. Adaptive noise function as used in the experiments. The function converges to σ_{max} for $\lambda > \eta$ and equals $\sigma_{min} + \epsilon$ for $\lambda = 0$. The values of σ_{min} , σ_{max} , and η are design parameters and ϵ a small constant.

3.3 Yaw Acceleration Model (EKF-YA)

In the third solution, the state vector \mathbf{x} is augmented by an additional parameter for the yaw acceleration $\dot{\xi}$. The resulting system model corresponds to a constant yaw acceleration model, i.e. $\ddot{\xi} \sim \mathcal{N}(0, \sigma_{\ddot{\xi}})$. Accordingly, the linearized system matrix and system noise matrix \mathbf{Q} have to be adapted, while the measurement model remains unchanged.

4 Experimental Results

4.1 Filter Configuration

The different filter approaches have been parametrized with the following system noise throughout the experiments (given in standard deviations):

	σ_{X_0}	σ_{Z_0}	σ_ψ	σ_v	σ_ξ	$\sigma_{\dot{v}}$	$\sigma_{\dot{\xi}}$	σ_{X_m}	σ_{Y_m}	σ_{Z_m}
EKF	0.01	0.01	0.001	0.001	0.05	1.0	-	0.025	0.025	0.025
EKF+YA	0.01	0.01	0.001	0.001	0.0001	1.0	1.0	0.025	0.025	0.025
EKF+OR	0.01	0.01	0.001	0.001	$\Lambda(\hat{\xi}_{OR})$	1.0	-	0.025	0.025	0.025
IMM(Q_{stat})	0.01	0.01	0.001	0.001	0.0001	0.0001	-	0.025	0.025	0.025
IMM(Q_{mnv})	0.01	0.01	0.001	0.001	1.0	1.0	-	0.025	0.025	0.025

where $\Lambda(\cdot)$ represents the adaptive noise function defined in Sec. 3.2. The noise function has been parametrized with $\sigma_{min} = 0.05$, $\sigma_{max} = 1.0$, and $\eta = 1.5$. This means, for small $\hat{\xi}_{OR}$ the EKF-OR filter is configured as the standard EKF filter and for $|\hat{\xi}_{OR}| > 1.5$, the filter corresponds to the maneuvering mode of the IMM filter. A window size of $k = 5$ has been used for the oracle.

The mode transition probability matrix P_{trans} for the IMM filter has been chosen as

$$P_{trans} = \begin{bmatrix} 0.98 & 0.02 \\ 0.02 & 0.98 \end{bmatrix} \quad (5)$$

where the value at the i th row and j th column indicates the *a priori* probability that the filter switches from mode i to mode j ($i, j \in \{1, 2\}$), i.e. it is much more likely that the filter remains in the same mode. Diagonal entries close to 1 prevent frequent mode switching and result in more distinct mode probabilities.

4.2 Synthetic Ground Truth

In a simulation, a virtual vehicle, represented by a set of $M = 60$ rigid 3D points, is moved along a synthetic *ground truth* trajectory. The trajectory shows a turn maneuver that can not be followed well by the single EKF approach if parametrized for typical longitudinal or slow turn movements. At each time step, the pose and motion state of the virtual object is known. Measurements are generated by projecting the object points at a given time step onto the image planes of a virtual stereo camera pair. The measurements are disturbed by adding Gaussian noise with standard deviation $\sigma_u = 0.1$ pixel to the horizontal image coordinate in both images, i.e. $\sigma_d = \sqrt{(\sigma_u^2 + \sigma_u^2)}$ and $\sigma_v = 0$.

The vehicle starts approaching at 50 m distance with constant velocity of 10 m/s and a small yaw acceleration of 0.05 rad/s^2 . After 50 time steps, a sudden constant yaw acceleration of 2 rad/s^2 is generated for 10 time steps (0.4 s). Due to the random measurement noise, the simulation has been repeated 40 times. The mean estimated yaw rate for each filter is shown in Fig. 3(a), together with the $1-\sigma$ (standard deviation) error band. As can be seen, the single EKF approach (constant yaw rate assumption) can not follow the fast yaw rate increase, while the proposed extended versions approximate the ground truth much better. Differences are in the delay of the yaw rate increase, i.e. how fast does a given filter react to the yaw acceleration, and in *overshooting*.

The EKF-YA filter quickly follows the yaw rate increase by estimating the yaw acceleration. Fig. 3(b) shows the response to the rectangular yaw acceleration input. The resulting overshooting of the estimated yaw rate is about twice

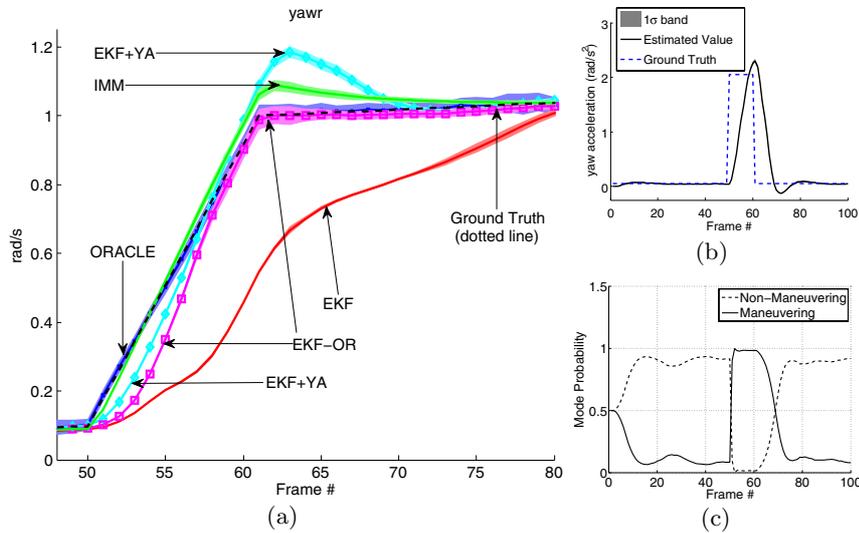


Fig. 3. (a) Error bands of yaw rate estimates based on mean and standard deviation over 40 monte carlo runs. All filter extensions approximate the ground truth much better than the original single EKF approach. (b) Estimated yaw acceleration of EKF+YA filter compared to ground truth. (c) IMM mode probabilities.

as large compared to the IMM approach, which is able to follow the yaw rate increase even faster by switching from non-maneuvering mode to maneuvering mode at frame 51 (see Fig. 3(c)). As the yaw acceleration maneuver ends at frame 60, the probability of the maneuvering mode starts to decrease. The oracle detects the yaw rate increase without delay and shows no overshooting at all. Since the oracle is model-free and unconstrained, the results show a larger standard deviation compared to the Kalman filter based methods. The resulting trajectories are quite unsteady. However, combining the oracle with a Kalman filter in the EKF-OR approach, yields both a very accurate yaw rate estimate and smooth trajectories with almost no overshooting due to the additional yaw rate measurements. The delay until the yaw rate starts to increase depends on the design of the adaptive noise control function (see Fig. 2).

4.3 Real World Ground Truth

The different filtering approaches have been also evaluated using a database of 57 real vehicle trajectories recorded from in-car sensors at different urban intersections, including 18 left turn maneuvers, 19 right turn maneuvers, and 20 straight crossing scenarios. Analog to the synthetic trajectory, a virtual point cloud object has been moved along the real trajectories.

The root mean squared error (RMSE) between estimated value and ground truth is taken as evaluation criteria. Fig. 4 shows the distribution of the RMSE for the different filters in terms of median and the 25th and 75th percentile (boxes). The whiskers indicate the first and (clipped) 99th percentile. As can

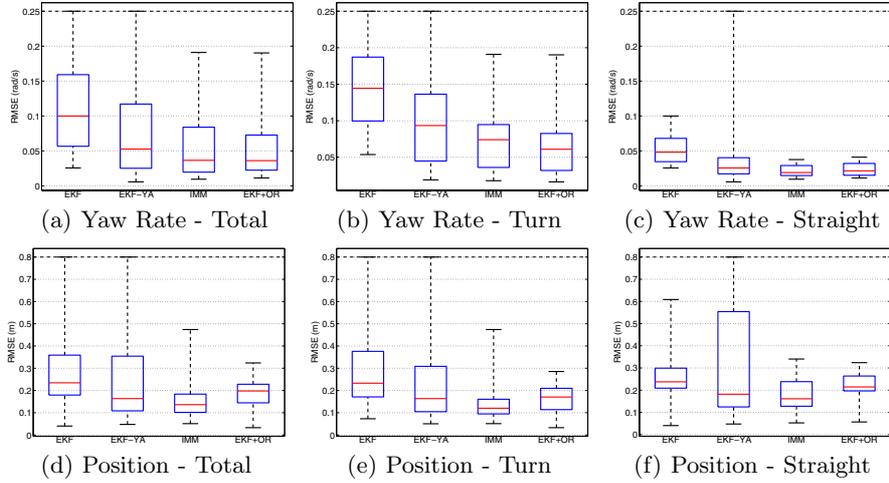


Fig. 4. Distribution of RMSE between estimated state and ground truth for yaw rate (top row) and position (bottom row). The boxes indicate the 25th and 75th percentile as well as the median.

be seen, the median RMSE of the yaw rate over all runs is significantly reduced by all proposed extensions compared to the original EKF approach (Fig 4(a)). The IMM and EKF+OR approach yield almost similar results and perform slightly better than the EKF-YA. The improvements in the estimated yaw rate directly affect the error in position, which is also decreased for all extensions (Fig 4(d)). Especially if only the subset of trajectories including a turn maneuver is considered for evaluation (second column). On the other hand, even for the straight motion trajectories (third column), improvements are achieved. The parametrization with much lower system variances at the non-maneuvering mode of the IMM filter is beneficial at low dynamic scenes as can be seen at Fig. 4(f). The EKF-OR filter equals the configuration of the EKF filter at straight motion, thus, the results do not differ much. Modelling of the yaw acceleration in the EKF-YA leads to larger deviations of the error at non dynamic scenes. The filter has to compensate for small errors in the orientation estimate by adapting the yaw acceleration, which typically leads to oscillation effects that increase the RMSE. We did not find a significant improvement in the shape estimate on the test data, since the initial noisy object point cloud is refined by the filters before the maneuvers start.

4.4 Filter Behaviour at Extreme Maneuvers

So far we have considered maneuvers that typically occur at urban intersections. In extreme situations, such as skidding on black ice, the assumption that the vehicle’s orientation is aligned with the moving direction is violated. We simulate a skidding trajectory by introducing an external force in terms of a lateral velocity component of 6 m/s, pushing the oncoming vehicle toward the observer

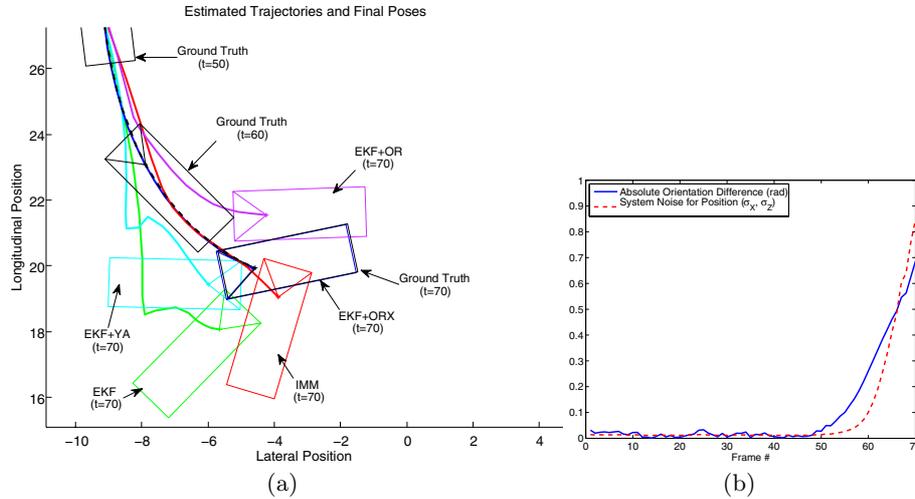


Fig. 5. (a) Estimation results of skidding trajectory. The EKF-ORX approach perfectly reconstructs the ground truth trajectory and final pose. (b) Absolute difference between object orientation and moving direction used for adaptation of the system noise.

while turning left with a constant yaw acceleration of $6 \text{ rad}/s^2$. As can be seen in Fig. 5(a), all filters discussed above fail to estimate the unexpected trajectory correctly since it does not agree with a circular path motion model. However, the model-free oracle is able to follow the sideslipping vehicle very accurately. Thus, we have adapted the idea of using the oracle not only to control the system noise for the yaw rate, but also for the position. The deviation between actual moving direction and object orientation, estimated by the oracle, is used as *skidding detector* (see Fig. 5(b)). Larger deviations indicate a violation of the circular path motion model. In this case, the system variances for position are increased using the same mechanism as introduced for the yaw rate. With large system variances for position, the vehicle is able to shift in arbitrary direction independent of the current orientation. The resulting final pose estimated by the so called EKF+ORX approach almost perfectly fits the ground truth, demonstrating the potential of this approach.

5 Conclusion

All of the proposed extensions have improved the tracking results of vehicles at typical turn maneuvers without losing performance at straight maneuvers and without any manual parameter tuning. The error in the estimate of yaw rate and position could be significantly reduced compared to the original single EKF approach. We found that the IMM approach yields the best compromise between tracking accuracy and computational complexity. Since the two IMM filters can be run in parallel on multi-processor architectures, the additional load reduces

to the mode mixing and probability update. The computation of the oracle is more costly compared to the Kalman filter, even if the sparseness in the involved matrices is exploited, and depends on the window size k . However, the skidding experiment has shown the potential of the oracle approach not only to improve the tracking at standard turn maneuvers, but also to detect extreme situations that are of special interest w.r.t. collision avoidance systems. The EKF-YA filter is an alternative if computation time is critical and only a single core processor is available. The additional computation time compared to the original EKF approach is negligible, while the results show a significant improvement at turn maneuvers.

The findings of this contribution are not restricted to image-based vehicle tracking approaches and could be directly applied to other sensors such as LIDAR sensors. Next steps will include to adapt the idea of the oracle to also control the noise level of the longitudinal acceleration and to reduce the oracle's complexity by restricting the rotation and translation from six to four degrees of freedom (in plane motion assumption).

References

1. Koller, D., Daniilidis, K., Nagel, H.: Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision* 10(3), 257–281 (1993)
2. Dellaert, F., Thorpe, C.: Robust car tracking using kalman filtering and bayesian templates. In: *Conference on Intelligent Transportation Systems* (1997)
3. Kaempchen, N., Weiss, K., Schaefer, M., Dietmayer, K.: IMM object tracking for high dynamic driving maneuvers. In: *Intelligent Vehicles Symposium*, pp. 825–830. IEEE, Los Alamitos (2004)
4. Barth, A., Franke, U.: Where will the oncoming vehicle be the next second? In: *Intelligent Vehicles Symposium*. IEEE, Los Alamitos (2008)
5. Zomotor, A.: *Fahrwerktechnik/Fahrverhalten*, 1st edn. Vogel (1987)
6. Bar-Shalom, Y., Rong Li, X., Kirubarajan, T.: *Estimation with Applications To Tracking and Navigation*. John Wiley & Sons, Inc., Chichester (2001)
7. Chan, Y., Hu, A., Plant, J.: A kalman filter based tracking scheme with input estimation. *IEEE Trans. on AES* 15(2), 237–244 (1979)
8. Badino, H.: A robust approach for ego-motion estimation using a mobile stereo platform. In: Jähne, B., Mester, R., Barth, E., Scharr, H. (eds.) *IWCM 2004*. LNCS, vol. 3417, Springer, Heidelberg (2004)
9. Tomasi, C., Kanade, T.: Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University (April 1991)
10. Horn, B.K.P., Hilden, H., Negahdaripour, S.: Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society A* 5(7), 1127–1135 (1988)