An Algorithm for Advanced Lateral Control of Autonomous Vehicles Applied to Car Following

Stefan K. Gehrig

Research Institute Daimler Benz AG
FT3/AB, HPC T 728
70546 Stuttgart
Germany

Phone: ++49-711-17-41484
Fax: ++49-711-17-47054
Gehrig@DBAG.Stg.DaimlerBenz.Com

Fridtjof J. Stein

Research Institute Daimler Benz AG
FT3/AB, HPC T 728
70546 Stuttgart
Germany

Phone: ++49-711-17-41880
Fax: ++49-711-17-47054
Stein@DBAG.Stg.DaimlerBenz.Com

Abstract

A crucial task for steering an autonomous vehicle along a safe path in a car following scenario is the lateral control. The sensory input of such a lateral control are the position coordinates of the leader vehicle. The following problem occurs: Due to the distance between the leader vehicle and the autonomous ego-vehicle, the lateral control has to interpolate a trajectory between the two vehicles. Using as a trajectory either a straight line or a curve of constant curvature causes the ego-vehicle to deviate from the leader vehicle's trajectory.

Given a system delivering 3D points of the leader vehicle with time tags, one has a handle to reconstruct the leader vehicle's trajectory. In addition, one has to compensate the motion of the ego-vehicle by using its motion parameters. Once this transformation is performed, the position coordinates of the leader vehicle are available in a coordinate system at rest. Knowing the position of the ego-vehicle in that coordinate system, one can select the trajectory point of the leader vehicle that is closest to the ego-vehicle as input to the lateral controller. This simple approach increases significantly the precision of car following systems.

The algorithm is applied successfully to an autonomous car for platooning at small velocities.

1 Introduction

In recent years an abundance of autonomous vehicle systems have been presented (e.g. [2, 4, 6, 11, 9, 12, 10]). In these presentations a popular approach to laterally guide the ego-vehicle was either lane following (e.g. [12]) or platooning (e.g. [5], [3]). In this paper we address platooning. For the lateral control there is one serious drawback: With only the current position information of the leader vehicle available, one has to interpolate the trajectory between the two vehicles. Both a straight line (tractor model) and a curve with constant curvature (arch) have been tried in previous applications (e.g.[3]). These interpolation approaches cause the autonomous vehicle to deviate from the leader vehicle's trajectory. The phenomenon scales with the distance to the leader vehicle. In platooning applications this behaviour can cause the autonomous vehicle to hit an obstacle such as the curb in a curve or a parking car. The deviation Δd of the trajectories is illustrated in Figure 1.

How is this paper organized ? Section 2 introduces a new algorithm that solves the problem explained above. In Section 3 a short mathematical treatment of the algorithm is provided. Lateral controller design issues are discussed briefly in Section 4. Section 5 contains both simulation results and results from our research vehicle. The final section summarizes the results and points out future work.

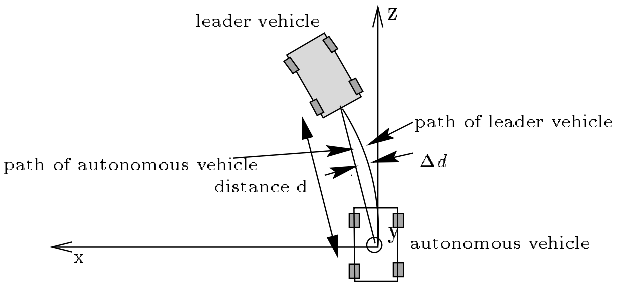

Figure 1: Trajectories of the autonomous car contrasted with the leader vehicle. The autonomous car uses a controller corresponding to the tractor model.

2 The Algorithm

For the sake of simplicity, we assume perfect sensor data and a very fast controller to explain the basic idea.

Unlike previous car following systems, the algorithm makes use of the time history associated with the leader vehicle. To determine the leader vehicle's trajectory it is sufficient to store the position coordinates of the leader vehicle and the motion parameters of the ego-vehicle over time. In order to transform the position coordinates into a coordinate system at rest, one has to compensate the motion of the ego-vehicle by using the ego-vehicle's velocity and steering angle. Knowing the position of the ego-vehicle in that coordinate system, one can select the trajectory point of the leader vehicle that is closest to the front of the ego-vehicle as input to the lateral control.

However, real lateral controllers have a certain delay to guarantee stability and compensate measurement errors of the sensor data. The next trajectory point to travel through must be a certain distance away from the ego-vehicle in order to allow the desired steering angle to have an impact on the ego-trajectory. To account for that, our algorithm always selects the trajectory point that exceeds a certain lookahead distance measured from the ego-vehicle. The mechanism is illustrated in Figure 2. Previous algorithms would have selected the trajectory point at t_n.

The algorithm explained above is referred to as the CUT algorithm (CUT - **C**ontrol **U**sing **T**rajectory) in subsequent sections.

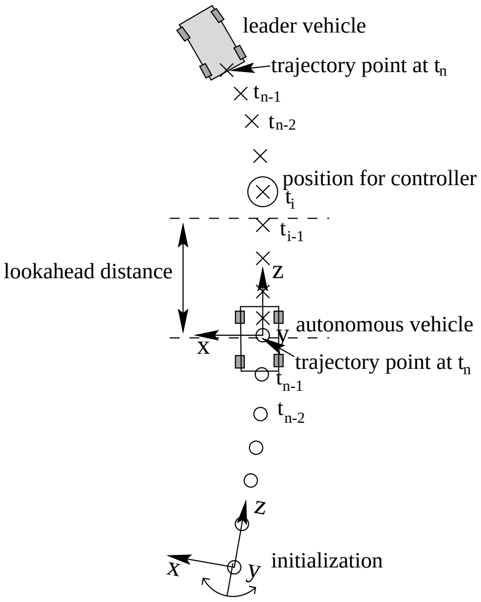

Figure 2: Mechanism of the CUT algorithm. Circles represent trajectory points of the ego-vehicle, crosses represent points of the leader vehicle.

3 Quantitative Analysis

3.1 Transformation to a Coordinate System at Rest

For the following analysis, the coordinate system of the previous figures is used. The y-axis pertrudes out of the paper plane. The (x, y, z) coordinates of the leader vehicle, the ego-velocity and steering angle are needed to reconstruct both trajectories. A flat road ($y = 0$) is assumed for all transformations. In a first step the coordinates of the ego-vehicle are transformed into a coordinate system at rest:

$$\begin{pmatrix} x'_{ego,n} \\ z'_{ego,n} \end{pmatrix} = \begin{pmatrix} x'_{ego,n-1} \\ z'_{ego,n-1} \end{pmatrix} + M_{rot} \begin{pmatrix} 0 \\ v_{ego,n} \end{pmatrix} \cdot \Delta t,$$

$$M_{rot} = \begin{pmatrix} cos(\gamma) & -sin(\gamma) \\ sin(\gamma) & cos(\gamma) \end{pmatrix}, \quad (1)$$

Here γ denotes the angle of the arch that the ego-vehicle has passed through since initialization:

$$\frac{v_{ego,n} \cdot \Delta t}{r_{ego}} = \gamma_n, \gamma = \Sigma\gamma_i. \quad (2)$$

n refers to the current time step, $n-1$ to the previous one. r_{ego} is the radius of curvature of the ego-vehicle. The primed coordinates refer to the coordinate system at rest.

In a second step, the coordinates for the leader vehicle are transformed into the same coordinate system:

$$\begin{pmatrix} x'_n \\ z'_n \end{pmatrix} = \begin{pmatrix} x'_{ego,n} \\ z'_{ego,n} \end{pmatrix} + M_{rot} \begin{pmatrix} x_n \\ z_n \end{pmatrix}. \tag{3}$$

The above transfomations are applied iteratively for every time step allowing the coordinates to be transformed to any point and orientation the autonomous vehicle has passed through.

3.2 The Vehicle Model

The radius of curvature r_{ego} is derived from the steering angle δ using the Ackermann model [14]:

$$x_c = sign(\delta) \cdot \frac{w}{2} + \frac{a+b}{\tan(\delta)}, \tag{4}$$

$$r_{ego} = \sqrt{x_c^2 + b^2}. \tag{5}$$

See Figure 3 for an explanation of the used quantities.

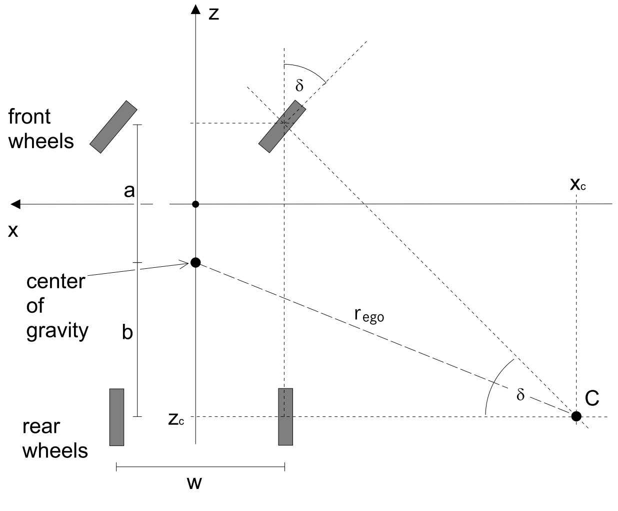

Figure 3: Ackermann Model (w: width of car, a: distance center to front axle, b: distance center to rear axle).

The Ackermann model is limited to small velocities and small lateral accelerations. Other models [14] that include side slip can be used to compute r_{ego} from δ to extend the validity of the transformation equation 1.

4 The Underlying Lateral Control

4.1 Interface of the Underlying Lateral Controller

In previously presented lateral control algorithms [5], the current x and z position of the leader vehicle has been the input to the lateral control. These quantities are now replaced by the x and z position of a trajectory point closer to the ego-vehicle determined by the lookahead distance. Using the appropriate control law, the desired steering angle is computed. The output to the actuator is the filtered desired steering angle.

4.2 The Underlying Lateral Controller

The choice of the underlying lateral control algorithm is uncritical with respect to the interpolation strategy since the distance to the target trajectory point is small. Hence no significant deviations occur. All lateral control algorithms regulate the x-offset to zero. The trajectory interpolation strategy determines the control law.

However, a controller and actuator with little delay and comparatively high dynamics is needed to follow arbitrary trajectories. Design issues of such controllers are not addressed in this paper. These issues including a stability analysis are treated in [8] and references therein.

For the simulations and for the research vehicle the lateral control algorithm interpolates an arch.

5 Results

5.1 Simulation Results

For the simulation a typical European street profile including a clothoid followed by an arch shape such as shown in Figure 4 is used. The velocity of both vehicles is $10m/s$ approximately constant over time and their distance is $25m$ constant over time.

The chosen street profile reflects a worst-case-scenario for car following in our application with limited opening angle of the sensor (30°) and limited velocity range in urban areas.

The simulated controller assumes an arch and regulates the new steering angle within $80ms$. The lookahead distance is $12m$ from the sensor ($11m$ from the front axle). Simulation results can be seen in Figure 5. Negative deviations are deviations towards the center of the curve. The dashed line denotes the old approach

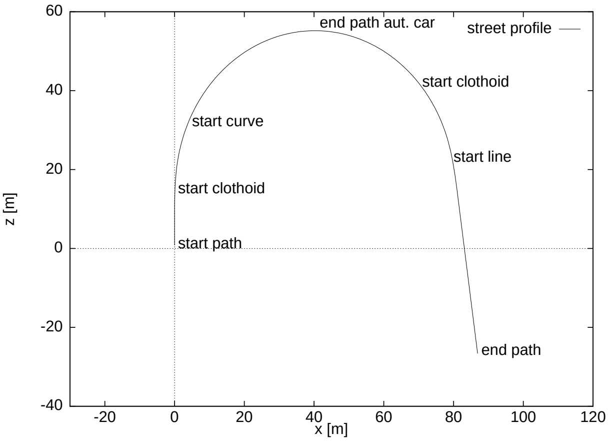

Figure 4: Route used for the simulation (coordinates in m).

and the solid line marks the deviation using the CUT algorithm. Even less deviation can be achieved by further reducing the lookahead but this also compromises the stability of a realistic underlying lateral controller.

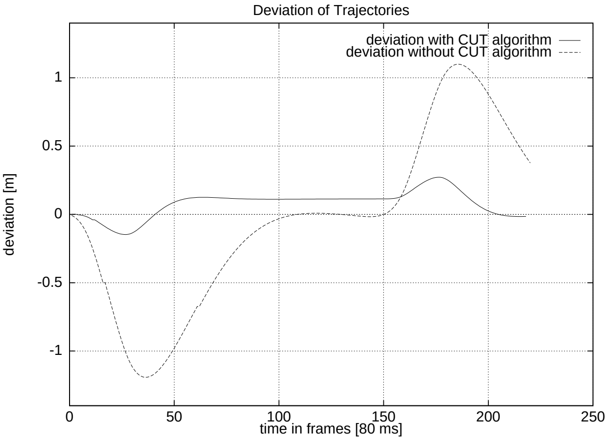

Figure 5: Deviation of the trajectories between the autonomous vehicle and the leader vehicle. The solid line denotes the deviation using the CUT algorithm, the dashed line indicates the standard approach.

The above simulation did not take any uncertainties into account. For a simple noise model, we take a noise signal of 3% of the distance evenly distributed in z and of $0.5m$ evenly distributed in x. That reflects the fact that position measurements with 3D sensors (range images, radar scanners, stereo cameras) have an error proportional to the distance. The noise x-direction was assumed to be independent of the distance since other effects incurring noise (occlusion, ...) dominate the error. There was no filtering of the sensor data. Figure 6 shows the results of the simulation including noise. No significant changes occur compared to the perfect sensor data.

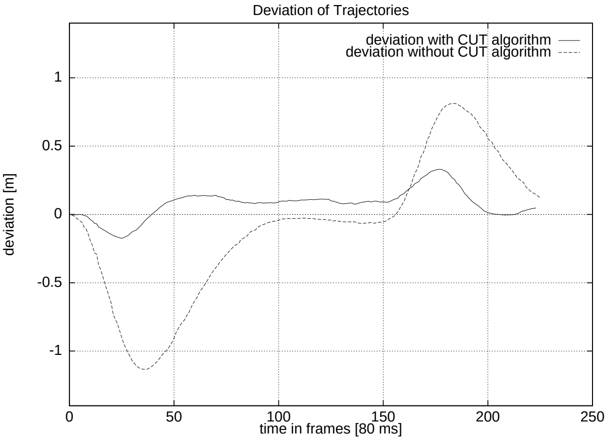

Figure 6: Deviation of the trajectories of the autonomous vehicle to the leader vehicle. The simulated noise is 3% of the distance evenly distributed in z and $0.5m$ evenly distributed in x.

5.2 Limitations of the Simulation

Above simulations illustrate the principal mechanism of the CUT algorithm. However, it does not reflect reality in several respects:

- Real 3D sensor data needs filtering. The Kalman filter [7] proves to be a good means of filtering for data that can be associated with a dynamic model. It is used for the trajectory measurements in our autonomous car.
- The lateral controller has slower control characteristics than the simulated controller and differentiating terms. Hence more delay occurs than in the simulation. This is needed to provide driving comfort and to suppress oscillations.
- The motion parameters used for the transformation have measurement errors. In addition, systematic errors occur when the Ackermann model is violated, e.g. when side slip occurs. These errors are bounded and do not accumulate over time. The autonomous vehicle's velocity and

steering angle are quite accurate since effects such as skewing are negligible in the urban velocity range. Although measurements of the steering angle are inaccurate for small velocities and hence yield inaccurate trajectory curvature, their contribution to the uncertainties is small because only the product $v_{ego} \cdot f(\delta)$[1] appears in the transformation equations. Also, the imprecise velocity measurements for velocities close to zero do not contribute much to errors since only the integral of the velocity contributes to the transformation. For higher velocities, the Ackermann model becomes invalid for comparatively small steering angles. More refined models have to be used.

- 3D imaging devices have limited opening angle. For extremely curved trajectories the leader vehicle might fall out of range of the sensor device. This is not taken into account in the simulation.
- The noise model used in the simulation is not realistic. Effects like occlusion are not taken into account and are expensive to simulate.

5.3 Preliminary Results from an Autonomous Vehicle

Our test vehicle is a Mercedes-Benz E420 equipped with actuators for throttle, brake and steering wheel. The sensor system consists of two CCD-cameras using stereopsis.

How can the center of the leader vehicle be measured adequately ? There are different approaches depending on the sensor. For truck platoons, the leader vehicle can be equipped with a fixed pattern to determine the center [5]. We choose to use a symmetry finder [13] for precise center measurement. The x and z coordinates of that point are sent to the lateral control.

In order to have the same initial conditions and the same behaviour of the leader vehicle we chose to use another simulation in the vehicle that creates a virtual leader vehicle. We pick the same profile as in the offline simulation for the leader vehicle (the first part of it). The distance to the ego-vehicle is around $25m$. The lookahead distance is set to $11m$.

The controller used here applies a control law that interpolates an arch like in the simulation. For passenger comfort and to compensate measurement errors the controller performs extensive low-pass filtering. It has a delay of around $400ms$ between desired steering angle as computed from the x and z coordinates and actual steering angle due to filtering and delay of the actuator. To filter the desired steering angle, a simple PD-controller with several limiters is used.

Figure 7 shows a plot of the deviations of the trajectories between the leader and the ego-vehicle. Negative deviations refer to deviations towards the inner side of the curve (cutting the corner). As expected the standard approach of car following results in cutting the corner when driving through a clothoid. Due to the controller delay and the relatively inert controller behaviour the trajectory of the ego-vehicle exhibits a slight overshoot. Moreover, slight oscillations indicate that the controller parameters are not yet perfectly adapted to the CUT algorithm.

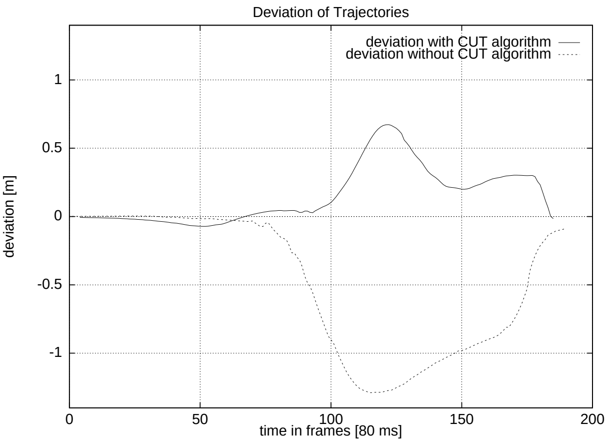

Figure 7: Deviation of the trajectories between the autonomous vehicle and the leader vehicle (solid: CUT algorithm, dashed: Controller interpolating an arch), measured in the autonomous test car.

In addition, the new CUT algorithm has been tested under real traffic conditions. No instabilities or oscillating phenomena were registered. A comfortable ride was provided at all times. Storing several traffic scenes with lateral and longitudinal controller active and reconstructing the trajectories never yielded a deviation above $0.4m$ for the CUT algorithm. Without the CUT algorithm, the observed deviations were much larger in very similar traffic situations. These numbers are somewhat imprecise since the measurement errors of the leader vehicle are not known. On the other hand the on-board leader vehicle simulation provides perfect sensor data for proper comparison.

The motion parameters were evaluated in a separate analysis. Velocity measurements showed errors of less than 1% above $2m/s$ and the absolute errors for

[1] Plugging in Equation 4 into Equation 5 and that result into Equation 2 yields Function f.

smaller velocities were also within $0.2m/s$, which does not contribute much to errors of the trajectory reconstruction. The steering angle inertia turned out to be a critical parameter. The steering angle offset of the sensor can be compensated by comparing the reconstructed trajectories going through a circle trajectory both clockwise and counter-clockwise. Errors on the steering angle sensor were found to be less than 2% for trajectory reconstruction under low lateral accelerations and dry weather conditions. Motion parameter errors can be compensated by measuring stationary objects in the traffic scene (see next section).

6 Conclusions and Future Work

The CUT algorithm achieves a significant improvement in precision for the lateral control in car following systems at very little computational expense. To take varying measurement quality of the position into account, it is planned to adjust the lookahead distance accordingly. Furthermore, one could adjust the lookahead proportional to the ego-velocity which yields a constant time difference between leader vehicle and ego-vehicle for all the trajectory points passed through. This accounts for the delay of the controller. A constant lookahead part should be added to account for the measurement errors of the sensors. In addition, it is possible to estimate the current deviation of the ego-vehicle from the leader vehicle's path. From that quantity one can derive the total uncertainty of the control loop and adjust the lookahead distance accordingly.

Similar to the lateral control, a longitudinal control algorithm can be built: The acceleration of the leader vehicle can be stored at every trajectory point and could be used to obtain the same acceleration at the same position for the ego-vehicle.

The transformation into a coordinate system at rest and consequently the measurement of the motion parameters can be improved significantly by measuring stationary objects in the scene. This is a common approach in the field of robotics (see e.g. [1]).

References

[1] M. Asada et al, "Representing Global World of a Mobile Robot with Relational Local Maps", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 20, No. 6, pp. 1456-1461, 1990.

[2] A. Bellon et al, "Real-time Collision Avoidance at Road Crossings on Board the PROMETHEUS-ProLab2 Vehicle",Proceedings of the Intelligent Vehicles 94 Symposium, 1994.

[3] P. Daviet et al., "Longitudinal and Lateral Servoing of Cars in a Platoon", Proceedings of the Intelligent Vehicles 96 Symposium, pp.41, 1996.

[4] E. D. Dickmanns et al,"The Seeing Passenger Car VaMoRs-P",Proceedings of the Intelligent Vehicles 94 Symposium,Oktober 1994.

[5] U. Franke et al., "Truck Platoon in Mixed Traffic", Proceedings of the Intelligent Vehicles 95 Symposium, pp.1, 1995.

[6] U. Franke et al., "Fast Stereo Object Detection for Stop and Go Traffic", Proceedings of the Intelligent Vehicles 96 Symposium, pp.339, 1996.

[7] R. E. Kalman,"A new approach to linear filtering and prediction problems", *Transactions ASME J. of Basic Engineering*, 1960.

[8] J. Kosecka, "A Comparative Study of Vision-Based Lateral Control Strategies for Autonomous Highway Driving", to appear in *IEEE Transactions of Robotics and Automation*, 1998.

[9] M. Parent et al., "Automatic Driving in Stop and Go Traffic", Proceedings of the Intelligent Vehicles 94 Symposium, pp.183, 1994.

[10] K. Sanyoshi, "Drive Assist System Using Stereo Image Recognition", Proceedings of the Intelligent Vehicles 96 Symposium, pp. 230, 1996.

[11] B. Ulmer et al., "VITA II Active Collision Avoidance in Real Traffic", Proceedings of the Intelligent Vehicles 94 Symposium, pp.1, 1994.

[12] J. Weber et al., "New Results in Stereo-Based Automatic Vehicle Guidance", Proceedings of the Intelligent Vehicles 95 Symposium, pp. 530, 1995.

[13] T. Zielke et al., "Intensity and Edge-Based Symmetry Detection with an Application to Car Following", *CVGIP: Image Understanding*, Vol. 58, No. 1, 1993.

[14] A. Zomotor, *Fahrwerktechnik: Fahrverhalten*, Vogel Buchverlag Würzburg, 1987.