

Vision And Odometry Based Autonomous Vehicle Lane Changing

Gábor Péter, Bálint Kiss

*Department of Control Engineering and Information Technology
Budapest University of Technology and Economics
Budapest, Hungary
petergabor@iit.bme.hu, bkiss@iit.bme.hu*

Viktor Tihanyi

*Department of Automotive Technologies
Budapest University of Technology and Economics
Budapest, Hungary
viktor.tihanyi@gjt.bme.hu*

Abstract—Autonomous driving is an old desire of mankind. The main purpose of this article is to present an autonomous lane changing method, that enables fully automated lane changing if given safety criteria are met. Environment detection is executed by multiple sensors such as radar, LIDAR and a smart-camera. Intrinsic sensors of the vehicle are utilized as well to ensure a smooth transition between lanes. The test results obtained on a closed test track prove the usefulness of the presented methodology. The resulting algorithm and experiences collected during implementation could be useful for integration into a larger system.

Index Terms—autonomous, lane change, vision, odometry

I. INTRODUCTION

Autonomous lane change is a complex task as the vehicle has to follow a predefined curve, calculated online. There might be objects on the road, like other vehicles or pedestrians - their presence should be taken into account by the planning phase of the maneuver. Safety is top priority, therefore the maneuver should be blocked in any situations where the execution would risk any of those in the surroundings of the vehicle.

The idea of autonomous driving is rather old, the first implementations were found in extraterrestrial vehicles where direct control was impossible due to several limitations. The first implementations of terrestrial vehicle applications date back to the eighties. A notable article is published by Dickmanns and Zapp where they were able to use a visual feedback system for autonomous driving back in 1987 [1]. A more advanced vision based lane detection system, implemented on a transputer system is presented by Suzuki et al. [2]. Transputers enable parallel processing, an idea that was reborn in modern GPUs. Vision based lane detection systems have evolved into potent contenders in the automotive industry [3].

It was soon discovered that trajectories with noncontinuous curvature are impossible to be traveled by vehicles, thus different methods for smoothing the transition between segments with different radii were examined such as in [4]. One step further is the estimation of lanes on roads without any markings [5]. In order to detect the ambient space, the vehicle has to be fitted with different sensors, possibly in a redundant way. After having all these safety measures applied, the implementation has to ensure the actual lane changing on

a lower level, while maintaining safety on a higher level by monitoring the extrinsic sensor signals. A lane marking and object detection system using multiple sensor technologies is detailed in [6]. A sensor fusion based autonomous driving system with lane detection and drivable region detection is presented in [7].

The first published method for lane changing was released by Kato in 1997 [8]. Highway driving is easier than getting around in cities as normally no pedestrians and thus unexpected situations arise. Autonomous driving within cities was studied well by Franke et al. [9]. A robust lane change assistant is developed by Liu keeping track of the vehicles behind the ego vehicle based on an image feed [10]. After coping with normal situations, collision avoidance become a major research area as detailed by Shiller and Sundar in 1998 [11]. As driverless vehicles are still banned in most countries the new way became driver assistant systems. These do not directly take control of the vehicle but can display safety warnings in dangerous situations - for instance upcoming vehicles in blind spots [12].

Lane changing requires a trajectory and a suitable controller that drives the vehicle along this trajectory. Therefore path planning is only half of the story, controller design is of same importance as depicted by Hatipoglu et al. [13]. A vision based lane keeping based on nested PID algorithm is presented in [14], but also fuzzy control methods are available for lane changing such as [15]. A model based lane change controller was developed by Du [16], while a lane keeper/changer control algorithm with low ripple for highway driving is presented by Kang [17]. There are methods increasing driver comfort such as the torque based lane change assist presented in [18]. Cooperative autonomous driving using V2X communication standard is a fresh branch of research. Some methods rely on vehicle-vehicle communication [19] while other are using the infrastructure system as well [20]. It should be noted however, that connected vehicles are vulnerable to hacking [21].

Section II lists the features of the test vehicle, especially the sensors and actuators used for lane changing. Section III details the vision processing pipeline, followed by the odometrical relations in the next section. The implementation is detailed in section V while the results are evaluated in the last section.



Fig. 1. Test vehicle

II. TEST VEHICLE

The test vehicle is a first generation Smart. After years of research and development, the current feature set of the car has little resemblance with its original version. The model was chosen for its simplicity and lower turning radius than almost any other competitors at the time the research started. A specialty of this micro car is its wheel location: The wheels are placed towards the four corners of the chassis making the car rather agile. The turning radius of the newest generation is less, than 7 meters, which is an outstanding value for a production car.

This agility on the other hand requires sophisticated drive algorithms as the car itself is rather unstable thanks to the low axle distance. Track width varies between front and rear axle, both around 1500 millimeters, while wheelbase is around 1900 millimeters. Average cars have slightly wider track width, with a wheelbase above 2500 millimeters. The axle distance is one key point to the agility of the car, the other key point is its rear drive system. Without the need of a front drive shaft, the maximum steering angle for the front wheels is larger than on average front wheel driven cars.

A. Sensors

The test vehicle is equipped with a complete set of sensors, providing all the measurements required for autonomous driving. The following list sums the ones, required for lane changing and the next paragraphs detail each one of them. Some of them are intrinsic, measuring the vehicles internal state, while others collect information from the ambient objects. Some of them are active, while others are emitting a signal in order to provide a measurement.

- 1) Front radar
- 2) Rear radar
- 3) LIDAR
- 4) Smart camera
- 5) ABS wheel speed sensors
- 6) Differential GPS
- 7) Steering wheel sensor

1) *Front radar*: The front radar is an active extrinsic sensor, gathering information about the upcoming obstacles in front of the vehicle. An obstacle detected by the front radar should disable the lane change function at certain situations.

2) *Rear radar*: The rear radar is identical to the front radar, thus it is also an active extrinsic sensor, but in this case gathering information about the object behind the vehicle. In case of another vehicle approaching at higher speed than the vehicles own, the lane change feature should be blocked.

3) *LIDAR*: The LIDAR (Light Detection and Ranging) is an active extrinsic sensor, emitting a laser beam and obtaining a close range polar map of the distance versus bearing. The installed sensors have an effective range of 16 meters and they are both fitted on the front corners of the car, sharing the frontal view, while having a separate side view and rear view region.

4) *Smart camera*: The smart camera is a passive extrinsic device, detecting other cars and lane markings. As its frame rate is bounded and processing takes some time, it also uses the cars speed signals to compensate these limitations with predictions. It can detect lane edge markings for the ego lane and for both the left and right lane as well, so altogether four parametric curves are its relevant output.

5) *ABS wheel speed sensors*: ABS wheel sensors are passive intrinsic sensors, measuring the speed of each of the wheels independently. As these are passive sensors, they do not work below 2.7 km/h. As long as the wheel speed is above this threshold value, they supply a reliable value, in case the effective tire diameter is estimated correctly! The supplied value is always reliable as it is in RPM, but using incorrect effective wheel diameter would corrupt the measurements. As the tires are made of rubber, their effective diameter differs under different loads. The exact value was measured with a scale from the ground to the axle center point while the vehicle was standing still. It should be noted that the changes in tire pressure should be taken into account as well!

6) *Differential GPS*: GPS is basically a passive extrinsic sensor, delivering the absolute coordinates in the world frame. Differential GPS is a bit more complicated as it communicates with nearby stations to cancel out unwanted atmospheric artifacts in order to have a more precise position estimate. A standard GPS can have refresh rate of 10 Hz and a position accuracy in the meters magnitude, while a differential GPS can go as low as some centimeters, which is required for precise location within a lane. Some more advanced GPS receivers also have an IMU (Inertial Measurement Unit) and combining the two sensors and some sensor fusion, they can provide position estimates at higher rates such as 100 Hz.

7) *Steering wheel sensor*: The steering wheel sensor is an active intrinsic sensor to determine the exact orientation of the steering wheel. It is necessary for any operation that involves lateral motion of the vehicle, such as lane changing. The sensor value is not used directly but it is an essential part of the steering control that was developed during the project.

B. Controller development

Sensing is only the first part, in order to make the vehicle act as commanded, actuators are required. By neglecting the lateral component, only the steering system has to be actuated, and it is done by a servo motor built into the steering column.

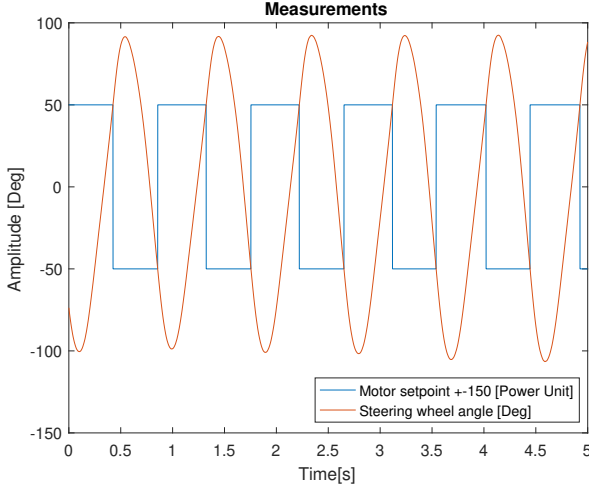


Fig. 2. Identification

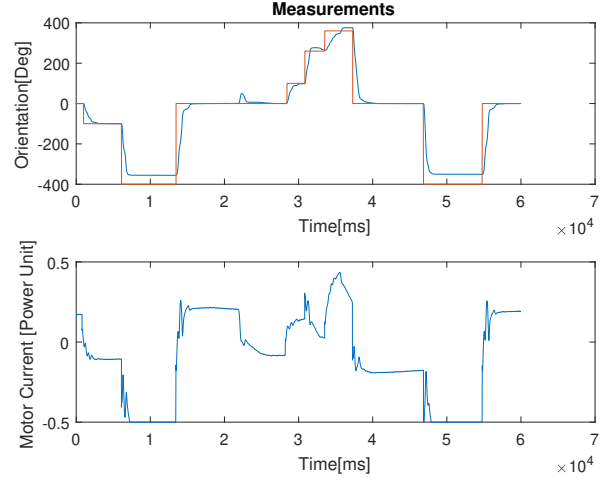


Fig. 3. Controller validation

The servo was retrofitted by Daimler and it is controlled via a current setpoint signal. The steering wheel angle is measured by a precise sensor, therefore a closed loop control is possible.

The chosen controller is a load estimator realized in state space. The controller development process consisted of the following tasks:

- 1) Identification
- 2) Model development
- 3) Controller design
- 4) Validation

1) *Identification*: Identification was realized by outputting a 150 Power Unit setpoint to the steering servo and inverting its sign as the feedback signal reached the boundary of a predefined safety region. This process eventually translates into a rapid cyclic left-right rotation of the steering wheel as long as the identification process is running. Identification is realized using MATLAB and the dSpace Autobox hardware at a cycle period of 1 millisecond.

2) *Model development*: Having the setpoint and the resulting angle of the physical system, the transfer function of the steering mechanism could be identified, thus a model could be built. As the controller was chosen as a state space one, the plant was transferred into a state space model in the following form:

$$\dot{x} = Ax + Bu \quad (1)$$

$$y = Cx + Du \quad (2)$$

Substituting the actual values results the following equation system:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix} x + \begin{bmatrix} 0 \\ 25000 \end{bmatrix} u \quad (3)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \end{bmatrix} u \quad (4)$$

3) *Controller design*: Controller design is realized based on load estimator paradigm in state space. For this method, the Ackermann formula is the mathematical way of establishing a stable state feedback. To meet the required performance, the bandwidth of the system was chosen as 10 radians per second while the damping factor was defined as 0.85. The sampling time is the same as the identification rate: 1 millisecond.

4) *Validation*: Validation is an essential part of controller development as performance characteristics need to be measured before deployment. After obtaining satisfying simulation results, the control algorithm was tested on the actual test vehicle. The results are depicted in Fig. 3.

The feedback value nicely follows the setpoint value, the only steady state error occurs as the setpoint is set to -400 degrees. As one can see, the output of the controller is saturated in this case, and the sole reason is a mechanical problem with the steering mechanics. Theoretically the steering wheel is operational between -370 degrees and +370 degrees, but the pinion and rack connection of the steering frame was most probably missed by one tooth, thus there is an offset of around 10 degrees in the whole system. This causes the system to saturate mechanically at around -360 degrees.

III. VISION

The most important sensing technique for a lane change algorithm is vision as the markings on road are undetectable by most other sensor types (LIDAR can detect lane markings based on their reflectance). As the need and possibility for autonomous driving increases, out of the box solutions are available at the market as well. The chosen camera is such a smart camera capable of delivering high level information about the ambient space in front of the car.

The problem is rather challenging as the camera is placed inside the car behind the windscreen, thus the light scatters and reflects on this extra surface. The greatest problem however is the direct sunlight which might totally blind the image sensor

as it is trying to compensate for the great amount of photons entering the armature.

To avoid most of the reflections a 3D printed support box was placed around the camera to cancel out unwanted reflections of objects placed on the dashboard during testing. The camera provides information in a 150 milliseconds interval, meaning that without a compensation the delay would be unbearable especially at higher speeds such as highway cruising. Therefore the camera requires the vehicles estimated speed in order to have a reliable output.

A. Lane polygons

The output stream includes multiple slots of information, however the relevant ones are the lane polygons. A lane polygon is a third order approximation that describes the curve of the lane markings in the frame attached to the camera itself. As the camera is capable of detecting the lane markings for both the ego lane and the two surrounding lanes, there are altogether 4 parameter sets. One parameter set includes the 4 polygon parameters and a last value that corresponds to the confidence level of the actual polygon detection. In case of a well detected lane marking curve it equals 100 percent while in case of non-existing lanes, it equals zero. At least in the ideal case, as practically it is something in between.

The first step before any further processing is to check whether a lane marking curve exists by checking the corresponding confidence variable. In case of a valid line, the polygon could be used for further processing, otherwise, the polygon has to be estimated. Lanes usually have markings at least on one side, or sometimes - in case of higher order roads - on both sides. In case of lesser roads without any lane markings, the camera is ineffective. But one valid polygon is enough for an estimation of the other missing polygons.

As the lane width is more or less constant for a given type of road, a missing lane curve could be estimated by offsetting the polygon from the other side of the same lane. The measurements showed that the test track is equipped with lanes of a width of 3.8 meters. This artificial lane polynom ensures a more or less correct representation of the missing lane markings. One drawback of this method is that it generates virtual lanes next to the actual road as well, thus a map based check is required to make sure the given lane exist to where the vehicle is ordered to switch lanes [22].

As the lane polygons represent the edges of the lanes, the center polygon of the lanes should be calculated as a first step. The polygons are given by their coefficients such as $C0$, $C1$, $C2$, $C3$. By having a coefficient set for both boundaries of a lane, the center curve could be parametrized as:

$$\begin{bmatrix} C0_{Center} \\ C1_{Center} \\ C2_{Center} \\ C3_{Center} \end{bmatrix} = \begin{bmatrix} C0_{Left} \\ C1_{Left} \\ C2_{Left} \\ C3_{Left} \end{bmatrix} / 2 + \begin{bmatrix} C0_{Right} \\ C1_{Right} \\ C2_{Right} \\ C3_{Right} \end{bmatrix} / 2 \quad (5)$$

The $C0$ values are the zero order coefficients, thus they represent the offset of the lane curve starting point and the

camera frames origin. The difference between $C0_{Left}$ and $C0_{Right}$ equals the lane width. Having the ego polynomial enables a control algorithm to stay at the center of the actual lane or if the curve is replaced by one, corresponding to a neighbor lane, a lane change could be executed.

But lane changing is not as easy as replacing the ego center polynomial with the center polynomial of the target lane. Theoretically it should work, but the smart camera is continuously refreshing lane information and at around the instant of the lane crossing, it enters an invalid state, thus outputting absolutely no lane information. On the other hand, the camera is optimized for the case when the car is driving along the center line of a lane, thus supplied coefficients degrade as the maneuver starts.

IV. ODOMETRY

Solving the problem, where the camera loses lane information during the lane change period could be possible, if the relative displacement of the vehicle would be known from the point where it started the maneuver. The camera does not provide such information, but using odometrical correlations between measured wheel speed values and displacement could lead to a precise relative displacement.

Odometry gives the opportunity to predict a vehicles actual pose (position and orientation) relative to a known starting point in case the wheel speeds are measured. The more precise these measurements are, the better the results will be. The popular method for Ackermann steering vehicle odometry is to represent the vehicle as a bicycle having only two wheels. The rear wheel is place in the center of the rear axle, while the front wheel - representing the steered wheels - is placed on the center point between the two front wheels.

As the two front wheels have different steering angles at all time other than the neutral position, the steering angle of the virtual wheel could be obtained using a transformation that takes into account the axle distance of the car, and the track width of the front wheels. As one can see, this transformation requires precise measurement of these distances and angles having several point for errors.

A way more simplified and resource friendly method is to represent the car as a differential driven entity. As controlling the car is out of scope (only odometry is at hand), this is a fair assumption, as an Ackermann steering vehicle can execute a subset of the motions of a differential driven robot. Using this methodology, only the rear wheel distance is required and the wheel speeds from the rear sensors. The speed of the front wheels and the steering wheel orientation is totally irrelevant, which decreases the possible errors, as the vehicle had an unknown steering wheel sensor offset.

As long as there is no slip event along the rear axle, the odometry provides an accurate estimate of the vehicles relative displacement from a given starting point as summarized in Eq. 6 - 8, where x and y represent position, θ the orientation of the vehicle, while ΔL stands for displacement, δ_R and δ_L stands for wheel rotation and e equals the half axle length.

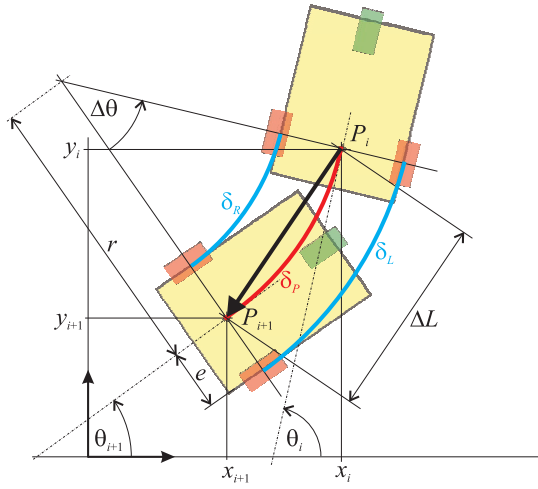


Fig. 4. Odometry

$$x_{i+1} = x_i + \Delta L \cos \left(\theta_i + \frac{\Delta \theta}{2} \right) \quad (6)$$

$$y_{i+1} = y_i + \Delta L \sin \left(\theta_i + \frac{\Delta \theta}{2} \right) \quad (7)$$

$$\theta_{i+1} = \theta_i + \Delta \theta = \theta_i + \frac{\delta_R - \delta_L}{2e} \quad (8)$$

A. Wheel sensors

Odometry relies on precise rotation values for the wheels, therefore the ABS speed sensors play an important role. As stated already these sensors are passive - using only induction, they require a minimum amount of wheel speed otherwise they produce no output signal. This minimum speed is around 2.7 kilometers per hour which is above the idling speed of any car even in first gear. So basically the problem of false sensor signals persist only in case of a stopped vehicle. This gives a limitation to the system: A lane change is not possible from a standstill. But as long as the vehicle is already in motion, the odometrical functions could be evaluated, giving precise results.

Luckily ABS requires the wheel speeds at high rate and with great precision, therefore these sensors provide excellent output that needs no further signal forming, other than the unit conversion as the output is given in RPM, rather than in meters per second. Providing the converted values into the odometrical function refreshes the vehicles pose.

V. IMPLEMENTATION

The final system was implemented in MATLAB environment using Autobox from dSpace. The starting cycle time was set to 1 millisecond, but after adding more and more higher level functionality, the system became unstable. The solution was to decrease the processing cycle frequency from 1kHz to 100Hz, thus the cycle time increased ten times to 10 milliseconds. This change was necessary and the stability returned to its maximum state, without losing any of the functionality.

As the steering controller was designed properly with discrete matrices, nothing had to be replaced or redesigned after the time base change.

The Autobox hardware gives easy to use solutions for interfacing with a vehicle via CAN interfaces and analog/digital I/Os. The corresponding Simulink blocks enable a quick and solution oriented software development, without the hassle that usually comes with embedded system programming.

As for the lane changing state machine, the lane change could be initiated by the turn signal lever. As soon as a long enough pulse is detected, the high level safety core checks the ambient sensors for possible safety hazards. This includes an obstacle in the selected lane or an object approaching from behind with higher speed than the vehicle itself. These situations are treated as unsafe situations, where the lane change request is ignored and driver is notified about the blocking event.

In case the situation is evaluated being safe, the maneuver begins. The center curve of the target lane is fed to the lane following controller instead of the ego lane parameters and the odometry is reset to the origin. As the camera outputs information in its own frame, the origin in this case means $[0, 0, 0]$ for the X , Y and θ components of the vehicle. During the maneuver, the lane information are frozen, and the navigation runs only using odometry. With an appropriate tuning of the controller, a target steepness and behavior of the lane change could be set in order to make it as smooth as possible.

As the vehicle reaches the center line of the target lane, the camera starts to pick up the lane, and starts producing valid lane information. The critical point of the maneuver is the switching from the odometry driven feedback back to the camera driven state. To ensure a smooth transition, both the distance from the target curve and a correct orientation has to be within a given region in order to enable a smooth transition. In case the vehicle is released too early, the camera might have not picked up the new lane yet.

In case of a too late release, the curve might become corrupt at that distance as it is only a third order estimation of the actual lane center curve. In case of a release at a moment where the vehicle is rather misaligned results in an a sudden steering reaction that might end up in catastrophic oscillation. In case of a release at a moment where the vehicle is well oriented but the position is way off, the vehicle might not be able to reach the center line in time as the lane following controller uses low error amplifier in case of displacements.

VI. RESULTS

Combining the the smart camera and the built in ABS wheel speed sensors of the car, the resulting lane change became as smooth as a human driver would be able to manage. The system was tested on a semi closed test track with the resulting trajectories depicted in Fig. 5. The weak point of the system is the lane following algorithm itself, that is a simple pure pursuit algorithm with adjustable look ahead distance.



Fig. 5. Lane change - I



Fig. 6. Lane change - II

After thorough testing, this look ahead distance turned out to be a key element, as driving on a straight track is optimal at high look ahead distance resulting smooth steering wheel operations, while driving through the turn in the middle of the track requires a bit more aggressive behavior from the steering wheel, thus the look ahead distance needed to be reduced. As the camera outputs the lane information in polynomial form, all the parameters were at hand, so the change was rather easily implementable.

VII. CONCLUSION

The proposed fusion of vision and odometry results a stable and reliable method that meets the expectations of a lane changing feature of an autonomous vehicle. Future plans include the optimization of the lane follower algorithm to make it even smoother as current algorithms produce some ripple and unwanted motion of the steering wheel while driving along a straight track.

ACKNOWLEDGMENT

The project has been supported by the European Union, co-financed by the European Social Fund. EFOP-3.6.2-16-2017-00002

REFERENCES

[1] E. D. Dickmanns and A. Zapp, "Autonomous high speed road vehicle guidance by computer vision1," *IFAC Proceedings Volumes*, vol. 20, no. 5, pp. 221–226, 1987.

[2] A. Suzuki, N. Yasui, N. Nakano, and M. Kaneko, "Lane recognition system for guiding of autonomous vehicle," in *Intelligent Vehicles' 92 Symposium., Proceedings of the. IEEE*, 1992, pp. 196–201.

[3] A. A. Assidiq, O. O. Khalifa, M. R. Islam, and S. Khan, "Real time lane detection for autonomous vehicles," in *Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on. IEEE*, 2008, pp. 82–88.

[4] W. Nelson, "Continuous-curvature paths for autonomous vehicles," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on. IEEE*, 1989, pp. 1260–1264.

[5] M. A. Sotelo, F. J. Rodriguez, L. Magdalena, L. M. Bergasa, and L. Boquete, "A color vision-based lane tracking system for autonomous driving on unmarked roads," *Autonomous Robots*, vol. 16, no. 1, pp. 95–116, 2004.

[6] R. Schubert, K. Schulze, and G. Wanielik, "Situation assessment for automatic lane-change maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 607–616, 2010.

[7] Q. Li, L. Chen, M. Li, S.-L. Shaw, and A. Nüchter, "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 2, pp. 540–555, 2014.

[8] S. Kato, K. Tomita, and S. Tsugawa, "Lane-change manoeuvres for vision-based vehicle," in *Intelligent Transportation System, 1997. ITSC'97., IEEE Conference on. IEEE*, 1997, pp. 129–134.

[9] U. Franke, D. Gavrilu, S. Gorzig, F. Lindner, F. Puetzold, and C. Wohler, "Autonomous driving goes downtown," *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 6, pp. 40–48, 1998.

[10] W. Liu, X. Wen, B. Duan, H. Yuan, and N. Wang, "Rear vehicle detection and tracking for lane change assist," in *Intelligent Vehicles Symposium, 2007 IEEE. IEEE*, 2007, pp. 252–257.

[11] Z. Shiller and S. Sundar, "Emergency lane-change maneuvers of autonomous vehicles," *Journal of dynamic systems, measurement, and control*, vol. 120, no. 1, pp. 37–44, 1998.

[12] M. Ruder, W. Enkelmann, and R. Garnitz, "Highway lane change assistant," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 1. IEEE, 2002, pp. 240–244.

[13] C. Hatipoglu, U. Ozguner, and K. A. Redmill, "Automated lane change controller design," *IEEE transactions on intelligent transportation systems*, vol. 4, no. 1, pp. 13–22, 2003.

[14] R. Marino, S. Scalzi, and M. Netto, "Nested pid steering control for lane keeping in autonomous vehicles," *Control Engineering Practice*, vol. 19, no. 12, pp. 1459–1467, 2011.

[15] J. E. Naranjo, C. Gonzalez, R. Garcia, and T. De Pedro, "Lane-change fuzzy control in autonomous vehicles for the overtaking maneuver," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 3, p. 438, 2008.

[16] Y. Du, Y. Wang, and C.-Y. Chan, "Autonomous lane-change controller via mixed logical dynamical," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on. IEEE*, 2014, pp. 1154–1159.

[17] C. M. Kang, Y. S. Gu, S. J. Jeon, Y. S. Son, W. Kim, S.-H. Lee, and C. C. Chung, "Lateral control system for autonomous lane change system on highways," *SAE International Journal of Passenger Cars-Mechanical Systems*, vol. 9, no. 2016-01-1641, pp. 877–884, 2016.

[18] M. Bujarbaruah, Z. Ercan, V. Ivanovic, H. E. Tseng, and F. Borrelli, "Torque based lane change assistance with active front steering," in *Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on. IEEE*, 2017, pp. 1–6.

[19] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer, and A. Kovacs, "Enhancements of v2x communication in support of cooperative autonomous driving," *IEEE communications magazine*, vol. 53, no. 12, pp. 64–70, 2015.

[20] F. You, R. Zhang, G. Lie, H. Wang, H. Wen, and J. Xu, "Trajectory planning and tracking control for autonomous lane change maneuver based on the cooperative vehicle infrastructure system," *Expert Systems with Applications*, vol. 42, no. 14, pp. 5932–5946, 2015.

[21] A. Szijj, L. Buttyán, and Z. Szalay, "Hacking cars in the style of stuxnet," 2015.

[22] A. Barsi, V. Poto, A. Somogyi, T. Lovas, V. Tihanyi, and Z. Szalay, "Supporting autonomous vehicles by creating hd maps," *Production Engineering Archives*, vol. 16, 2017.