Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Real-time optimal motion planning for automated road vehicles ⋆,⋆⋆

Ferenc Hegedüs * **Tamás Bécsi** ** **Szilárd Aradi** **
Zsolt Szalay *** **Péter Gáspár** **

* *Robert Bosch Hungary, Budapest, Hungary (e-mail: ferenc.hegedus@hu.bosch.com)*
** *Department of Control for Transportation and Vehicle Systems, Budapest University of Technology and Economics, Budapest, Hungary (e-mail: {becsi.tamas; aradi.szilard; gaspar.peter}@mail.bme.hu)*
*** *Department of Automotive Technologies, Budapest University of Technology and Economics, Budapest, Hungary (e-mail: zsolt.szalay@gjt.bme.hu)*

Abstract: This paper presents a real-time optimal motion planner algorithm for road vehicles. The method is based on a cubic spline trajectory planner which is able to plan a set of vehicle motions driving from a given initial state to a required final state. Maximal dynamical feasibility and passenger comfort are ensured by minimizing the lateral acceleration and tracking errors as the vehicle moves along the trajectory. Tracking of the planned motion is realized during planning and execution as well by separate longitudinal and lateral controllers. Efficient implementation and small number of optimization variables enables real-time usage. The trajectory planner is first tested in a quasi real-time simulation environment and then under real working conditions at the dynamic platform of proving ground ZalaZone with a completely drive-by-wire Smart Fortwo. Measurement results are presented and analyzed in detail, and possible future research directions are mentioned.

Keywords: real-time motion planning, trajectory planning, dynamical feasibility

1. INTRODUCTION

Automation of road transportation is a very important and actual topic for today's vehicle industry and academic institutions in this field as well. Due to the many possible advantages, manufacturers of vehicle systems are investing large amounts into the research and development of automated vehicle features. Autonomous road vehicles are expected to be more energy-efficient and environment-friendly by optimizing driving strategies (Watzenig and Horn (2016)). They are also predicted to improve road traffic parameters, such as average travel time and traffic flow capacity significantly (Tettamanti et al. (2016)). The realization of self-driving cars includes numerous technical challenges. One of these is trajectory planning, the design of the vehicle's motion. Over the last two decades, various techniques were developed to address the motion planning problem of road vehicles. The vehicle's path can be build from geometric curves such as circular arcs, clothoids, or splines (Vorobieva et al. (2013)). In case of graph search methods, the vehicle's spatiotemporal surrounding is discretized into a graph where vertices are representing vehicle states and edges are representing motion primitives leading from one state to another. The vehicle's trajectory is then found by a search for the minimum-cost path in this graph (Tianyu Gu et al. (2015)). A common drawbacks of these approaches is that the nonholonomic dynamics of the vehicle is hard to incorporate in the planning problem.

A proven approach for dynamically feasible trajectory planning is nonlinear optimization. One of the important papers that lay down the basics of optimal motion planning for wheeled vehicles is the work of Howard and Kelly (2007), where the authors are driving a planetary rover on rough terrain. Since than, optimization based algorithms are often used to solve various motion planning setups. In Zhang et al. (2019) optimal model predictive techniques are used to plan collision-free trajectories for multiple vehicles simultaneously. Mixed-integer quadratic programming is used to in Qian et al. (2016) to incorporate obstacle avoidance and traffic rule constraints in the optimization problem. In Mote et al. (2016) the authors are dealing with a collision-tolerant formulation to be able to handle scenarios where a collision is inevitable with minimal damage. A major shortcoming of optimization-based methods is that depending on the complexity of the underlying system, they require significant computational effort in most cases. Because of this, real-time application is often impossible, restricting the usability of these methods to simulation purposes. There are examples in literature such as the

⋆ EFOP-3.6.3-VEKOP-16-2017-00001: Talent management in autonomous vehicle control technologies - The Project is supported by the Hungarian Government and co-financed by the European Social Fund.
⋆⋆The research reported in this paper was supported by the Higher Education Excellence Program in the frame of Artificial Intelligence research area of Budapest University of Technology and Economics (BME FIKP-MI/FM).

work in Ziegler et al. (2014) and Majd et al. (2018) for real-time solutions, but this comes at the price of using very simple kinematic vehicle models. This paper aims to present an algorithm, where the efficient implementation and the reduction of control parameters for the optimization allow real-time performance despite that a fairly complex vehicle model is used to ensure dynamical feasibility. Our main goal was to prove the applicability of the proposed method not only by simulation but with real vehicle measurements as well.

2. OPTIMAL TRAJECTORY PLANNING

During trajectory planning, the primary goal is to drive the vehicle from a given initial state to a desired end state, through a set of allowed states. The motion of the vehicle can naturally be divided into a longitudinal and a lateral part. Considering the lateral motion, the vehicle's position x, y and yaw (heading) angle ψ have to be considered at least. Another important state variable that may worth dealing with is yaw rate $\dot{\psi}$ as stated in Hegedüs et al. (2017). The longitudinal part of the motion is characterized by the (longitudinal) velocity v of the vehicle. The

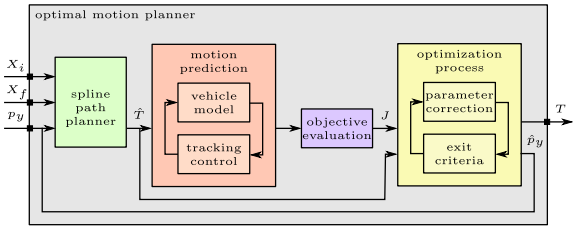

Fig. 1. Motion planner architecture

motion planning problem is then to find a trajectory that reaches the prescribed final state $X_f = [x_f \ y_f \ \psi_f \ v_f]^T$ starting from the given initial state $X_i = [x_i \ y_i \ \psi_i \ v_i]^T$, while moving through a safe and dynamically feasible region of the environment. Dynamical feasibility is achieved by the model-based prediction of the resulted motion. Due to the limited extent of the paper however, collision avoidance strategies are not part of the presented work. To reach passenger satisfaction, the planned motion must be comfortable and fast as well. The different passenger requirements can be considered through the optimization process. Although energy-efficiency and emission topics are also very important, they are not dealt with as the planned motion is too short-term to be able to address these aspects properly.

2.1 Cubic Spline Path Planning

The reference path of the vehicle is chosen as a cubic spline $y = S(x)$. The spline's end points are defined by the initial x_i, y_i and final x_f, y_f positions of the vehicle. The path spline is not a natural spline; the end point second derivatives are not stipulated to be zero (McKinley and Levine (1998)). Instead of that, the first derivative values are constrained at the end points to match the initial ψ_i and final ψ_f yaw angle of the vehicle:

$$S'(x_i) = \tan\psi_i \quad \text{and} \quad S'(x_f) = \tan\psi_f. \tag{1}$$

To get a path spline, $n \geq 1$ intermediate points have to be defined as well. Fig. 2 shows a set of cubic spline vehicle paths planned with one (top) and two (bottom) intermediate points.

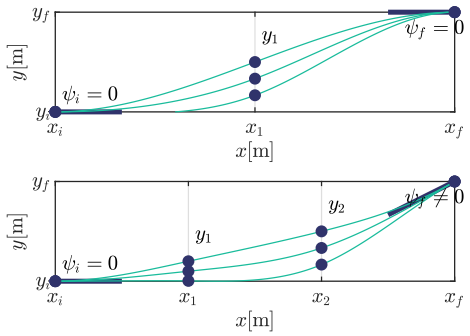

Fig. 2. Cubic path splines

As the end point values and derivatives are fully defined by the end states of the vehicle, the intermediate points of the path are remaining the free parameters of the planning. By varying the location of intermediate points, a whole set of paths can be generated that are leading from the initial end state to the final one. To further reduce the number of variables, intermediate points are distributed equidistantly along the longitudinal coordinate:

$$x_j = x_i + \frac{x_f - x_i}{n+1} j \quad j = 1 \dots n. \tag{2}$$

This leaves only the vector of lateral coordinates of the intermediate points $p_y = [y_1 \ y_2 \ \dots \ y_n]^T$ free to vary.

For the sake of simplicity, a linear velocity profile is assigned to the cubic spline path, changing the vehicle velocity linearly in time from v_i to v_f.

For simulation and measurement usage, the resulted trajectory T is represented numerically in form of way points:

$$\begin{aligned} X_k &= [x_k \ y_k \ \psi_k \ v_k]^T, \quad k = i, 1, \cdots, n, f \ , \\ T &= [X_i \ X_1 \ \cdots \ X_n \ X_f] \, . \end{aligned} \tag{3}$$

The values of the path spline $y = S(x)$ can be calculated numerically with an equidistant sampling of the longitudinal coordinate x. The resolution here must be relatively dense ≈ 1 cm to provide a smooth result, as the path function can change rapidly. To be able to calculate velocity values for every coordinate pairs, and to provide a more compact representation, the coordinates are recalculated with equidistant sampling along the arc length, where a resolution of ≈ 10 cm is suitable.

2.2 Motion Prediction

During motion planning, the nonholonomic dynamics of road vehicles has to be considered to ensure the dynamical feasibility of the trajectory. Namely, the vehicle has to be able to track the planned motion. This can be reached by a model based simulation of the vehicle's motion along the planned trajectory.

Vehicle Dynamics For the prediction of the vehicle's motion, a precise nonlinear single track model is used which is basically the same as the one used in Hegedüs et al. (2019). The model describes the planar motion of the vehicle chassis as well as the rotational motion of the front and rear axles, reduced into two virtual wheels. It

includes a dynamic wheel slip model in order to provide a feasible solution also in case of critical maneuvers. To sufficiently model the steering actuator of the vehicle, an additional first order steering dynamics is introduced, which is described by:

$$\dot{\delta} = \frac{k_s}{T_s}\delta_{sw} - \frac{1}{T}\delta, \tag{4}$$

where δ_{sw} is the steering wheel angle, k_s is the steering ratio, and T_s is the time constant of the steering mechanism.

Tracking Control Path tracking is realized by a Stanley controller. Stanley controller is a nonlinear feedback control that ensures asymptotic tracking of the reference path. The wheel level steering angle of the front axle is calculated as

$$\delta_s = e_\psi + \arctan(K_s \frac{e_{lat}}{v}), \tag{5}$$

where e_ψ is the yaw tracking error, and e_{lat} is the lateral tracking error of the vehicle, and K_s is a gain parameter. The tracking errors are calculated compared to a reference point on the path. This reference point is the closest point of the path curve to the center of the front axle. As the path curve is given in discrete samples, the exact

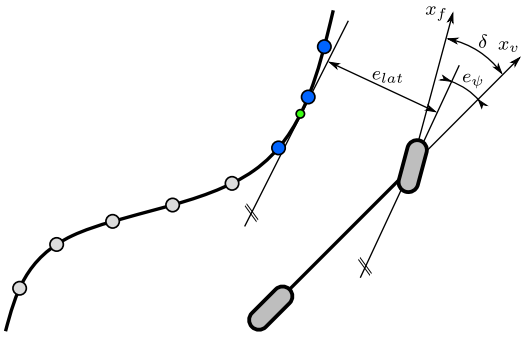

Fig. 3. Reference point interpolation

intersection way point $X_r = [x_r \ y_r \ \psi_r \ v_r]^T$ has to be interpolated. It is important to provide smooth interpolation results, as the yaw and lateral tracking errors are directly influencing the value of the steering angle. Fig. 3 shows the tracking error calculation. The true reference point (green) is calculated by a second order interpolation using the closest and the two neighboring way points (blue). For efficiency, the closest way points are searched in two steps. Firstly, only an equidistant subset of way points is checked in a reasonably sized surrounding of the vehicle to find the region in which the closest way points lie. Then, all the way points inside this region are evaluated again.

The Stanley controller was chosen over infinite horizon LQR control because it provides similar performance but its realization is much more simple, as it does not require the calculation of the vehicle's lateral velocity as well as the tracking error derivatives, which would only be possible by an additional state observer (Snider et al. (2009)). Fig. 4 shows the tracking of a circular path with a longitudinal velocity of 20 m/s. The tracking errors of the Stanley and LQR controllers are in the same range.

Longitudinal velocity tracking is carried out by an infinite horizon LQR control which is adjusting the total driving

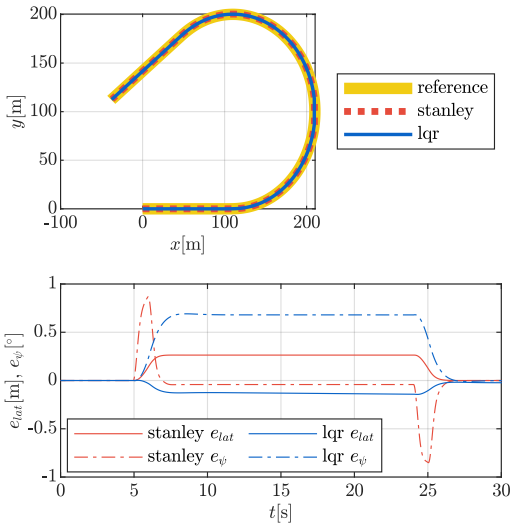

Fig. 4. Path tracking (above) and tracking errors (below) with Stanley and LQR controllers

or braking torque of the vehicle by a state feedback of the longitudinal velocity and the velocity tracking error.

2.3 Optimization Problem Formulation

The goodness of a trajectory regarding dynamical feasibility can be expressed as the motion that can be followed with the minimal tracking errors. To provide a faster journey, travel time can be minimized as well. Regarding comfort, the magnitude of lateral acceleration is also often kept at a minimum level (Shin et al. (2018), Zhang et al. (2013)). Considering the factors presented above, a cost functional can be expressed:

$$J(X(t,p_y)) = w_t t + \frac{1}{t}\int_0^t (w_{e_{lat}}|e_{lat}| + w_{e_\psi}|e_\psi| + w_{a_y}|a_y|)dt, \tag{6}$$

where $w_{e_{lat}}$, w_{e_ψ}, w_{a_y}, and w_t are weighting factors, a_y is the lateral acceleration of the vehicle, and t is the travel time along the trajectory. The resulting optimization problem can be mathematically formulated as:

$$\begin{aligned} &\underset{p_y}{\text{minimize}} \quad J(X(t,p_y)) \\ &\text{subject to} \quad \dot{X}(t,p_y) = f(X(t,p_y), X_i, X_f), \end{aligned} \tag{7}$$

where the function f represents the common dynamics of the vehicle and the tracking control. The meaning of Eq. 7 is that given the current initial state X_i and the desired final state X_f, we are searching the optimal lateral coordinate values p_y for the cubic spline path function, that will minimize the travel time t as well as the average absolute value of tracking errors e_{lat} e_ψ and lateral acceleration a_y, as the vehicle is driven through the trajectory.

2.4 Implementation

For real-time performance the vehicle and controller models were implemented in C++ using the linear algebra package Eigen. The differential equation system of the vehicle and controllers are solved by an own implementation of the 4$^{\text{th}}$ order Runge-Kutta method. The step size for the solution is chosen as 1 ms, so that the transient

wheel slip model provides plausible output also in case of dynamically demanding scenarios. The optimization problem is solved by an Interior-point method. The exit criterion for the step size of the optimization variable p_y is set to ≈ 1 mm. The algorithm is fast enough to be used for real-time measurements in case of one or two intermediate points applied in the cubic spline trajectory.

3. TEST EQUIPMENT

The main purpose of present work was to test the developed algorithm under real circumstances. To prepare for vehicle measurements, a quasi real-time simulation environment was set up first to provide absolutely safe testing capabilities. The real test drives were then carried out at the and of May 2019 at the newly established test track ZalaZone. ZalaZone is an innovative automotive proving ground that incorporates traditional features concentrating on driving dynamics with research and development infrastructure elements for the validation of highly automated road vehicles (Szalay et al. (2018)). The test maneuvers were driven at the dynamic platform which is circular surface with a diameter of 300 m, covered with special quality asphalt.

3.1 Vehicle Interface

The interface to the test vehicle is shown on Fig. 5. Just as in the internal motion prediction phase of the planner, the trajectory is executed by separate longitudinal (velocity tracking) and lateral (path tracking) controllers. For

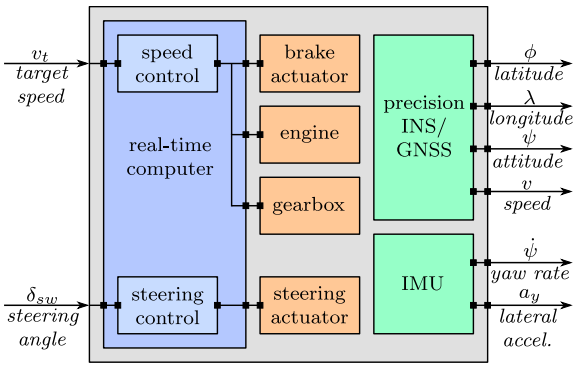

Fig. 5. Test vehicle interface

velocity tracking, only the target velocity v_t is specified as the test vehicle has its own longitudinal control implementation. Steering control is however realized with the trajectory planner's Stanley controller, and the resulted steering wheel angle δ_{sw} is provided to the vehicle.

The position and orientation of the vehicle are obtained in form of geodetic latitude ϕ, longitude λ and heading (attitude) angle ψ. For the evaluation of control law Eq. 5, the actual vehicle velocity v is also needed. Similarly, the lateral acceleration a_y is taken to calculate the value of cost functional Eq. 6 for evaluation purposes.

As the trajectory Eq. 3 is calculated in flat Earth coordinates, the vehicle position which is provided as geodetic latitude and longitude has to be transformed. For the conversion, the origin of the flat Earth coordinate system is chosen as $\phi = 46.895038$ °, $\lambda = 16.843526$ °, which is the approximate center of the dynamic platform.

On the physical level, all the signals are received from and transmitted to the test vehicle's main CAN (Controller Area Network) communication bus.

3.2 Simulation Environment

The real-time simulation environment aims to provide the same interface as the real test vehicle. The simulation utilizes the same vehicle model and longitudinal control as the trajectory planner's motion prediction layer. To model the CAN communication, MATLAB and Vehicle Network Toolbox is used. Timing is carried out using an internal timer object with a rate of 10 ms. To get geodetic latitude and longitude, a conversion from flat Earth vehicle position is needed with the same origin as in case of the real vehicle. During the simulation, all required vehicle signals can be plotted in real-time to help with the debugging of the planning algorithm.

3.3 Test Vehicle

Our test vehicle is a Smart Fortwo which is electronically controllable and is used for testing various automated functions (Tihanyi and Szalay (2017)). The car is shown on Fig. 6. On sensor side it is equipped with a precision INS/GNSS (Inertial Navigation System / Global Navigation Satellite System) navigation system iTrace, which is providing vehicle position, velocity and attitude data. Accelerations and turn rates of the vehicle chassis are also measured by a Bosch IMU (Intertial Measurement Unit) with 6 degrees of freedom. The test vehicle is mounted with a front view camera for lane detection, and a middle range front radar as well as two scanning lidars for object detection. The motion planning and tracking software is

Fig. 6. Test vehicle

running on a Windows PC which is connected to the vehicle's main CAN network via a Vector CANcase XL interface. On low level, the vehicle's actuators are controlled by a DSpace AutoBox real-time computer unit. The internal speed controller is responsible for maintaining the desired target speed v_t by calculating the required driving or braking torque, and the required gear. The vehicle's original throttle pedal travel sensor is producing an analog voltage output. Based on the amount of required driving torque, this signal is calculated and provided to the engine control unit. During braking, the original brake pedal is pushed by a linear actuator trough a mechanism. The force control of the linear actuator is solved by commercial servo controller units. These are served with analog reference

signals as well, computed based on the required amount of braking torque. The original vehicle has a robotized gearbox already, so the AutoBox is simply connected to a small electronics that mimics the behavior of the original gear stick. The required steering wheel angle δ_{sw} is set by the internal steering control. The steering actuator is a BLDC (Bushless Direct Current) servo motor equipped with a servo control unit. Its position is measured via an analog potentiometer.

4. RESULTS

During the real test drive, lane change and curved lane keeping like trajectories were planned and tracked at the dynamic platform of the test track. The longitudinal velocity v_t of the vehicle was kept at 30 km/h for safety reasons, as this was the first time that the planning algorithm was used in real working conditions. The test drives were carried at from standstill position. The start of trajectory planning and tracking was triggered when the test vehicle first reached the target velocity. Weighting factors for the optimization problem (6) were $w_{e_{lat}} = 1$, $w_{e_\psi} = 0.2\frac{180}{\pi}$, $w_{a_y} = 0.5$, and $w_t = 0$.

4.1 Lane change maneuver

Fig. 7 shows the planned, predicted, and actual path of the vehicle (top), as well as the applied steering wheel angle δ_{sw} (bottom) in case of a lane change maneuver. The results show that despite the test vehicle is tracking the reference trajectory well, the required steering wheel angle to do so is greater. The main reason for this phenomenon

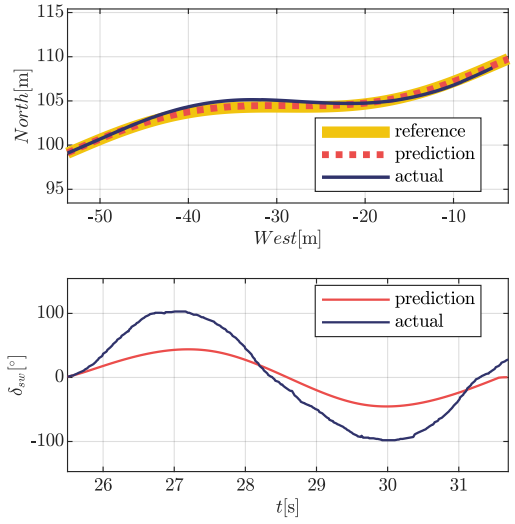

Fig. 7. Lane change path (top) and steering angle (bottom)

lies in the poor weather conditions during testing. The asphalt surface of the test track was covered with water during the whole period of measurements which reduced the value of friction coefficient below the value $\mu = 0.9$ that was used for planning. Another likely cause of the relatively big deviance is that there was no sufficient time to tune the parameters of the vehicle model, especially the steering ratio k_s, to optimally match the actual parameters of the test vehicle. The tracking errors during motion prediction and real test drive are shown on Fig. 8. The lateral acceleration a_y of the vehicle exceeds 3.4 m/s^2 along

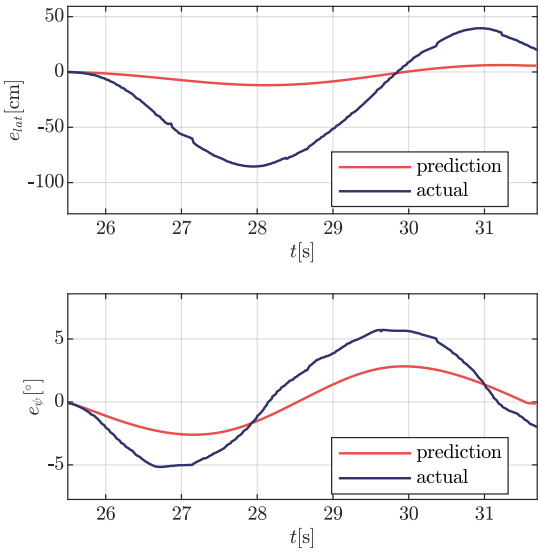

Fig. 8. Lateral (top) and heading (bottom) errors during lane change

this trajectory, which is considerable. The test vehicle is able to track the trajectory with a maximal lateral error of $\approx$ 80 cm and heading error of $\approx$ 5 °. The results are sufficient, but the errors are greater than in the simulation case. As the lateral controller was tuned with the simulation environment as a preparation before the actual measurement took place, the tracking was unfortunately not working with maximum performance.

4.2 Curved lane keeping maneuver

Fig. 9 shows the path and steering wheel angle δ_{sw} in case of a lane keeping maneuver. The required steering wheel angle to track the trajectory is again greater in the real world case. Trajectory tracking is carried out with a

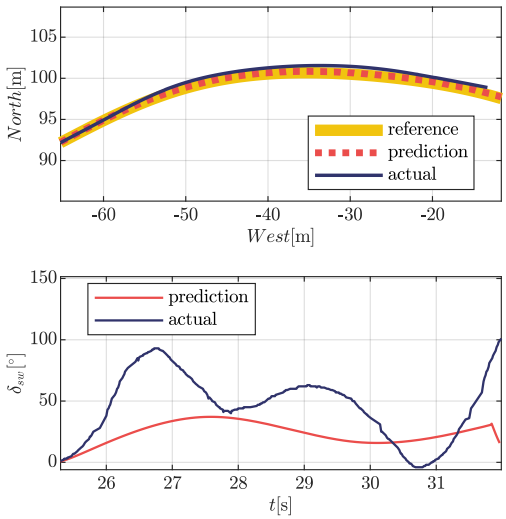

Fig. 9. Lane keeping path (top) and steering angle (bottom)

maximal lateral error of $\approx$ 95 cm and heading error of $\approx$ 2 ° as shown on Fig. 10. The result is again acceptable considering that the vehicle's lateral acceleration a_y exceeds 2.5 m/s^2.

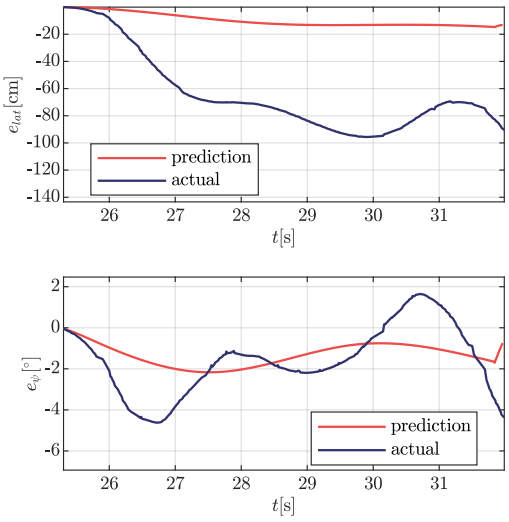

Fig. 10. Lateral (top) and heading (bottom) errors during curved lane keeping

5. CONCLUSIONS

In this paper a model-based real-time optimal trajectory planner was presented for road vehicles. The algorithm first plans a cubic spline trajectory from the given initial state to the required end state and then optimizes the shape of this trajectory to reach maximum dynamical feasibility and comfort by the prediction of the resulted motion. The efficient implementation in C++, and the small number of optimization variables allows real-time usage. Results show that the algorithm is capable to drive a real vehicle to the required target state at an acceptable tracking error level considering the dynamically demanding scenarios. A future development goal is to test the method not only when performing short-term individual maneuvers, but also in case of driving along a longer route, e.g. a race track. The parameters of the motion prediction layer shall be tuned to match the behavior of the actual vehicle as much as possible. For optimal tracking performance, the control parameters shall be also adjusted. The current work is not dealing with obstacle avoidance strategies, which are however essential and must be part of future research. Currently it is an open question if the optimization problem extended with the obstacle avoidance constraints could be solved fast enough to be used in real-time.

REFERENCES

Hegedüs, F., Bécsi, T., Aradi, S., and Gáspár, P. (2017). Model based trajectory planning for highly automated road vehicles. *IFAC-PapersOnLine*, 50(1), 6958 – 6964. doi:https://doi.org/10.1016/j.ifacol.2017.08.1336. 20th IFAC World Congress.

Hegedüs, F., Bécsi, T., Aradi, S., and Gáspár, P. (2019). Motion planning for highly automated road vehicles with a hybrid approach using nonlinear optimization and artificial neural networks. *Strojniki vestnik - Journal of Mechanical Engineering*, 65(3), 148–160.

Howard, T.M. and Kelly, A. (2007). Optimal rough terrain trajectory generation for wheeled mobile robots. *The International Journal of Robotics Research*, 26(2), 141–166.

Majd, K., Razeghi-Jahromi, M., and Homaifar, A. (2018). Optimal kinematic-based trajectory planning and tracking control of autonomous ground vehicle using the variational approach. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 562–566. IEEE.

McKinley, S. and Levine, M. (1998). Cubic spline interpolation. *College of the Redwoods*, 45(1), 1049–1060.

Mote, M.L., Afman, J., and Feron, E. (2016). A framework for collision-tolerant optimal trajectory planning of autonomous vehicles. *CoRR*, abs/1611.07608.

Qian, X., Altché, F., Bender, P., Stiller, C., and de La Fortelle, A. (2016). Optimal trajectory planning for autonomous driving integrating logical constraints: An miqp perspective. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 205–210. IEEE.

Shin, H., Kim, D., and Yoon, S.E. (2018). Kinodynamic comfort trajectory planning for car-like robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 6532–6539. IEEE.

Snider, J.M. et al. (2009). Automatic steering methods for autonomous automobile path tracking. *Robotics Institute, Pittsburgh, PA, CMU-RITR*.

Szalay, Z., Hamar, Z., and Simon, P. (2018). A multi-layer autonomous vehicle and simulation validation ecosystem axis: Zalazone. In *International Conference on Intelligent Autonomous Systems*, 954–963. Springer.

Tettamanti, T., Varga, I., and Szalay, Z. (2016). Impacts of autonomous cars from a traffic engineering perspective. *Periodica Polytechnica Transportation Engineering*, 44(4), 244–250.

Tianyu Gu, Atwood, J., Chiyu Dong, Dolan, J.M., and Jin-Woo Lee (2015). Tunable and stable real-time trajectory planning for urban autonomous driving. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 250–256.

Tihanyi, V. and Szalay, Z. (2017). Autonomous vehicle platform for demonstration purposes. In *34th International Colloquium on Advanced Manufacturing and Repairing Technologies in Vehicle Industry, Visegrád, Hungary*, 145–148.

Vorobieva, H., Minoiu-Enache, N., Glaser, S., and Mammar, S. (2013). Geometric continuous-curvature path planning for automatic parallel parking. In *2013 10th IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 418–423.

Watzenig, D. and Horn, M. (2016). *Automated driving: safer and more efficient future driving*. Springer.

Zhang, K., Wang, J., Chen, N., and Yin, G. (2019). A non-cooperative vehicle-to-vehicle trajectory-planning algorithm with consideration of drivers characteristics. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 233(10), 2405–2420.

Zhang, S., Deng, W., Zhao, Q., Sun, H., and Litkouhi, B. (2013). Dynamic trajectory planning for vehicle autonomous driving. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 4161–4166. IEEE.

Ziegler, J., Bender, P., Dang, T., and Stiller, C. (2014). Trajectory planning for bertha a local, continuous method. In *2014 IEEE intelligent vehicles symposium proceedings*, 450–457. IEEE.