Lateral Model Predictive Control for Autonomous Vehicle Prototypes

Nico Steinhardt, Raphael Wenzel, Malte Probst , Markus Amann

2023

Preprint:

This is an accepted article published in IFAC World Congress 2023. The final authenticated version is available online at: https://doi.org/[DOI not available]

Lateral Model Predictive Control for Autonomous Vehicle Prototypes

Raphael Wenzel * **Nico A. Steinhardt** * **Markus Amann** *
Malte Probst *

* *Honda Research Institute Europe GmbH, Carl-Legien-Straße 30, 63073 Offenbach am Main, Germany (e-mail: firstname.lastname@honda-ri.de*

Abstract: This paper shows a (lateral) Model Predictive Control (MPC) implementation on an Autonomous Driving (AD) prototype. Rapid prototyping and testing of AD functions in a realistic environment is a crucial step to understanding the advantages and shortcomings of algorithms in research and development of AD. Prototype vehicles show a specific set of requirements which differ from the control deployed in the final products. Such vehicles are potentially equipped with steering and pedal actuation, as well as high-precision localization systems. The control system is used for lateral trajectory control – it expects desired trajectory commands from a high-level motion planner. Building a precise prototype vehicle trajectory control has high challenges due to actuation lag and velocity-dependent vehicle dynamics, making MPC especially suited for such applications. The dynamic models used in this controller allow for simple identification and parametrization by conducting basic driving maneuvers and applying a series of commands to the actuators while recording the vehicle's reactions with a reference measurement system. As these dynamic effects are heavily velocity-dependent, the model linearizes its internal equations at the expected velocity, which is part of the trajectory command. This enables a wide velocity range of control, reaching from standstill to about 70 km/h. In this paper, we will present both the model and architecture of the lateral control as well as the identification steps necessary to deploy it.

Keywords: Automotive Control; Intelligent Autonomous Vehicles

1. INTRODUCTION

As the research on Autonomous Driving progresses, more and more situations are within the scope of active automation. Advanced Driver Assistance Systems (ADAS), which have used controllers to influence and stabilize the vehicle, have a well defined scope in which both ADAS functionality and control have to work. In the past, this allowed to deploy individual controllers to each ADAS and allowed the application and parametrization for each vehicle type individually. With the rise of intelligent AD systems, which have a much broader scope, advanced control techniques which work reliably in changing conditions have to be found.

This is also true for prototype vehicle control, where the demands and requirements are different from controllers deployed in the final production vehicle. Due to safety reasons, controllers for end-product ADAS are tuned for robustness in the specific use case rather than usability for multiple use cases. Making quick changes for testing is simplified with the system shown here, while all under the assumption of having an environment with proven test safety, e.g. multi-staged safety switches and a professional driver. Therefore, under these conditions, provable stability of the controller in an end-product fashion is of less importance than the overall control quality and explainability of the approach. Prototype vehicles are usually equipped with more reliable and high quality sensors, such as high-end LIDARs or RTK-GPS systems.

Furthermore, prototype vehicles can be used for multiple systems under test, making a versatile controller a platform component on which several AD and ADAS functions can rely. At our Honda Research Institute Europe (HRI-EU), such a research platform vehicle comprising multiple sensors as well as pedal and steering actuation is used for generic ADAS research, as shown in Weisswange et al. (2019) in the context of a highway assistance function.

Higher-level ADAS and AD functionalities based on trajectory planning require control accuracy within centimeter range, often over velocities from standstill up to highway speed, such as described in Probst et al. (2021). Especially for testing such functions on a prototypical base, the abstraction and compensation of real-world effects like actuator and vehicle dynamics simplify the vehicle-independent function development significantly.

This work provides a linear, time-variant Model Predictive Control (MPC) algorithm based on the work of Gutjahr et al. (2017) which is designed for such use cases, providing a precise and general trajectory interface upstream and still being sufficiently simple to be fully parametrized by measuring the vehicle's reaction to some basic driving maneuvers.

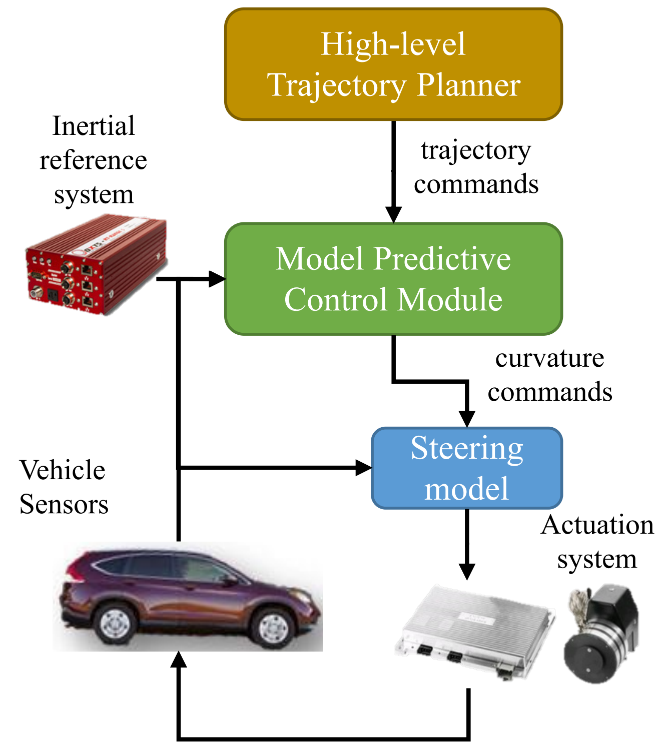

Fig. 1. Block diagram of trajectory control

We set up a rapid prototyping system which is portable in different vehicles, suitable for upcoming next steps handling even more complex, multi-vehicle scenarios like shared traffic spaces as presented in Wenzel et al. (2021). The basic selection criteria for the algorithm and detailed information about the modelling process can be found in Wenzel (2018).

2. SYSTEM SETUP

2.1 Scope of MPC

The algorithm shown in this paper is designed to apply a trajectory control on a car, modelling the vehicle's and actuator's dynamic behavior and issuing steering commands. It expects sensor centimeter-accuracy measurements of the vehicle's state of motion and location. The resulting output command is the desired curvature, with the assumption that the actuation system is able to follow these commands accurately.

2.2 Typical prototype setup

A typical system setup of a rapid-prototyping vehicle, which fulfills the above criteria, is shown in Fig. 1. The MPC module receives its desired trajectories from a high-level planner like described in Probst et al. (2021), obtains its current state measurements from a Differential GNSS inertial reference system, and issues its resulting commands downstream. By this, the MPC forms an abstraction layer between trajectories and currently desired driving parameters. As a minimum requirement, a trajectory command is comprised of spatial locations over time as desired values, as well as of (expected / desired) velocities over time as main linearization parameter. Typically, one

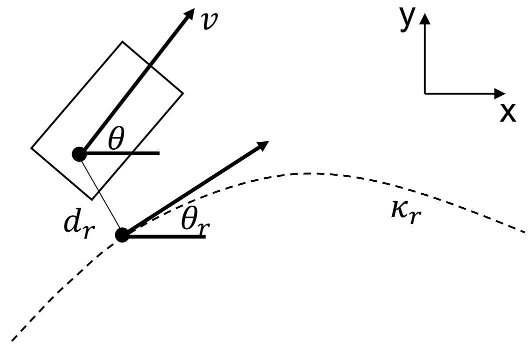

Fig. 2. Position and orientation of the vehicle relative to reference trajectory

more abstraction layer is required downstream to convert these driving parameters into actuation commands, considering the current state of driving and the steering characteristics. In Fig. 1, this is shown as steering model. As this paper focuses on the MPC module, this layer is not described in detail here.

2.3 Trajectory control model

The MPC module is set up as a linearized state-space model. The state-space formulation is then discretized and converted into a Quadratic Programming (QP) formulation in real-time; the required optimization for MPC is done by the qpOASES algorithm (Ferreau et al. (2014)). The MPC model itself can be devided into three sub modules, decribing vehicle kinematics, vehicle and actuator dynamics:

`Vehicle kinematics` *describe the mechanical relationships between the vehicle's current state of motion and the planned trajectory, as shown in section 3.1. In this work, the term "vehicle kinematics" refers to all vehicle motion without time dependency.*

`The reference trajectory`*, which is shown in section 3.2, include the desired trajectory inside the state-space model using the disturbance term of state-space equations.*

`Vehicle dynamics` *have a strong, non-linear dependency on the vehicle's velocity. In order to deal with these non-linearities, the dynamics part as described in section 3.3 is linearized around the desired velocity of the commanded trajectory, assuming that the actual velocities will be reasonably close. In this work, the term "vehicle dynamics" refers to all time-dependent vehicle motion (e.g. lag and oscillations).*

`Actuator dynamics` *describe the mechanical characteristics of the steering actuation system over time. It contains the actual steering wheel angle δ as well as the input u, as can be seen in section 3.4. By optimizing the input vector to the model given the cost function and weights, we obtain the optimal input, yielding the actual commanded steering wheel angle for the actuator.*

3. MPC DESIGN

3.1 Vehicle Kinematics

Since the the scope of the MPC is confined to low dynamic manoeuvres, we assume that both slip angle β and slip angle rate $\dot{\beta}$ can be neglected compared to the yaw angle ψ and yaw rate $\dot{\psi}$ of the vehicle. Thus, in the following the course angle θ equals the yaw angle. Additionally, the steering model (see Fig. 1) uses a slip angle compensation mechanism to improve the overall control performance. The lateral deviation from the commanded trajectory is defined as the distance d_r between the vehicle reference point and the reference curve, as shown in Fig. 2. The distance can therefore be described by the longitudinal velocity of the vehicle and the difference between vehicle and reference curve angle:

$$\dot{d}_r = v_x(t)\sin(\theta - \theta_r) \tag{1}$$

The desired angle θ_r is dependent on the curvature κ_r (see Fig. 2) of the current point of the trajectory. Under the given assumptions, the equation for the lateral vehicle dynamics is given by:

$$a_y = v_x(t)(\dot{\psi} - \dot{\beta}) \approx v_x(t)\dot{\theta} \tag{2}$$

The state equation of the yaw angle then can be derived from Eq. 2 and is given by:

$$\dot{\theta} = \frac{a_y}{v_x(t)} \tag{3}$$

As these equations depend on the vehicle velocity $v_x(t)$ the model is linearized using the expected velocity at each time step of the receding horizon.

3.2 Reference Trajectory

The commanded trajectory is modelled as disturbance z. Eq. (1) simplifies the relation of reference angle and reference curvature:

$$\dot{\theta}_r = v_x(t)\frac{\cos(\theta - \theta_r)}{1 - \kappa_r d_r}\kappa_r \approx v_r \kappa_r \tag{4}$$

$$\dot{\kappa}_r = z \tag{5}$$

where $v_r(t)$ is the vehicle's velocity projected onto the commanded path. For small angle differences $\theta - \theta_r$ (which means a small control error) and reasonable curvatures, this can be approximated to be roughly $v_r(t) \approx v_x(t)$, as described in Gutjahr et al. (2017).

3.3 Vehicle Dynamics Model

The vehicle dynamics model is drastically simplified since the target of this MPC are low dynamics manoeuvres. A second-order lag element behaviour accounts for the not instantaneous change in lateral vehicle state as a result of of steering wheel actuation.

$$\dot{a}_y = r_y \tag{6}$$

$$\dot{r}_y = -\frac{1}{{T_C}^2}a_y - \frac{2D_C}{T_C}r_y + \frac{K_C(v_x(t))}{{T_C}^2}\delta \tag{7}$$

A downside of this model is that we have to be able to estimate the current lateral jerk r_y of the vehicle in order to accurately compute the current state for the MPC.

3.4 Actuator Dynamics

In order to smooth out the actuated behavior and to account for inherent lag in the actuation system (e.g. behaviour of the steering servo motor and its controller), the actuator dynamics are also included into the MPC model. By modelling the actuator dynamics as a second-order lag element, we can directly penalize high steering rates. This improves the overall behavior of the controller and inspires trust of passangers as the actuator movement becomes smooth and calm. Therefore, the steering rate ϕ is introduced. The non-oscillating actuator dynamics can be modeled through sufficiently high damping:

$$\dot{\delta} = \phi \tag{8}$$

$$\dot{\phi} = -\frac{1}{{T_A}^2}\delta - \frac{2D_A}{T_A}\phi + \frac{K_A}{{T_A}^2}u \tag{9}$$

Again, this model relies on measuring both the current steering angle as well as the current steering rate of the actuator, while the latter can be also be estimated if no direct measurements are available.

3.5 Overall Model

The continuous linear time-variant system model is given by the following equation:

$$\dot{\mathbf{x}}(t) = \mathbf{A}_C(t)\mathbf{x}(t) + \mathbf{B}_C(t)\mathbf{u}(t) + \mathbf{E}_C(t)\mathbf{z}(t) \tag{10}$$

with the state vector $\mathbf{x}(t) = [d_r, \theta, \theta_r, \kappa_r, a_y, r_y, \delta, \phi]$

$$\mathbf{A}_C(t) = \begin{bmatrix} 0 & v_x(t) & -v_x(t) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{v_x(t)} & 0 & 0 & 0 \\ 0 & 0 & 0 & v_x(t) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{{T_C}^2} & -\frac{2D_C}{T_C} & \frac{K_C(v_x(t))}{{T_C}^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{{T_A}^2} & -\frac{2D_A}{T_A} \end{bmatrix} \tag{11}$$

$$\mathbf{B}_C(t) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{K_A}{{T_A}^2} \end{bmatrix}, \mathbf{E}_C(t) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

3.6 Matrix Batching

Generally, MPC solves the control problem by finding a trajectory $\mathbf{x}(k) = [\mathbf{x}(1), \mathbf{x}(2) \dots \mathbf{x}(n)]$ which yields the optimal solution for the given situation. In order to do that, we have to define the optimization problem for the complete time horizon by concatenating the model matrices for each time step according to the batch approach as described in Borrelli et al. (2017). This allows the consistent optimization of all of the timesteps in our quadratic solver QPOases (see Ferreau et al. (2014)) With the vectors:

$$\mathbf{u} = [u_0, u_1, ..., u_{N-1}]^T, \mathbf{u} \in \mathbb{R}^N \tag{13}$$

$$\mathbf{z} = [z_0, z_1, ..., z_{N-1}]^T, \mathbf{z} \in \mathbb{R}^N \tag{14}$$

$$\mathbf{x} = \left[x_1^T, ..., x_N^T\right]^T \tag{15}$$

$$\mathbf{y} = \left[y_1^T, ..., y_N^T\right]^T \tag{16}$$

$$\mathbf{x} = \mathcal{A}\mathbf{x}_0 + \mathcal{B}\mathbf{u} + \mathcal{E}\mathbf{z} \tag{17}$$

$$\mathbf{y} = \mathcal{C}\mathbf{x} \tag{18}$$

the batched state transition matrix $\mathcal{A}$ is created:

$$\mathcal{A} = \left[(\mathbf{A}_0)^T \left(\prod_{q=0}^{1} \mathbf{A}_{1-q} \right)^T \cdots \left(\prod_{q=0}^{N-1} \mathbf{A}_{N-1-q} \right)^T \right]^T \tag{19}$$

as well as the input matrix $\mathcal{B}$:

$$\mathcal{B} = \begin{bmatrix} \mathbf{B}_0 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_1\mathbf{B}_0 & \mathbf{B}_1 & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \left(\prod_{q=1}^{N-1} \mathbf{A}_{N+0-q} \right) \mathbf{B}_0 & \cdots & \mathbf{A}_{N-1}\mathbf{B}_{N-2} & \mathbf{B}_{N-1} \end{bmatrix} \tag{20}$$

These vectors and matrices are batched and aligned in such a way that all timesteps of the receding horizon are contained and that each consecutive step references on the step before. As such, the whole system behavior over a certain time is modeled within one state space formulation, allowing the time region inside the receding horizon to be optimized at once.

4. PARAMETRIZATION

The design of the dynamic vehicle model facilitates rapid-prototyping applications in real vehicles. The parameters can be identified by a series of driving manoeuvres and actuator commands. Typically, the parameter identification is done with the following steps:

4.1 Steering to lateral acceleration gain $K_C(v_x(t))$

This variable describes the velocity-dependent gain factor of lateral acceleration per steering wheel angle and is typically measured in a series of driving in circles with constant velocity and constant steering wheel angle per measurement, but with varying velocities and steering angles over the series. The MPC's performance can be greatly improved if this variable is additionally linearized around the current steering wheel angle to account for the typical variable steering gear ratio, while this technique is not described in detail here for simplicity reasons.

4.2 Lateral time constant T_C and damping D_C

These variables comprise the time constant and damping of the second-order vehicle lag element as described in Eq. (7). Their identification can be done by manually conducting a steering wheel angle step response measurement with the vehicle at a constant and safe velocity and record lateral acceleration over time. With $K_C(v_x(t))$ already known, T_C and D_C can be adapted such that the computed / simulated response reasonably match the measurement.

4.3 Actuation gain K_A

This variable contains the gain factor of the MPC steering angle output u wrt. the actual steering wheel angle δ. Usually, this parameter is 1.0 by definition, but can be used for adapting an arbitrary output gain.

4.4 Actuator time constant T_A and damping D_A

These variables contain the time constant and damping of the second-order steering actuator lag element as described in Eq. (9). Their identification is possible by issuing a step command to the actuator and measuring the actual steering wheel angle over time. This measurement should be done while the vehicle is moving slowly, as steering forces are significantly higher in complete standstill and may lead to biased measurements. With K_A already known, T_A and D_A can be adapted such that the computed / simulated response reasonably match the measurement.

5. CONCLUSION

We have shown a Model Predictive Control algorithm for lateral control, with a very flexible and general interface (Trajectories to steering angle commands). While having high demands on motion and location measurements as well as vehicle state estimation, it is easily parametrizable for new vehicles in a rapid-prototyping fashion. Future work will show the performance of this approach on test vehicles under various driving conditions.

REFERENCES

Borrelli, F., Bemporad, A., and Morari, M. (2017). *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press. doi:10.1017/9781139061759.

Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., and Diehl, M. (2014). qpoases: a parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4), 327–363. doi: 10.1007/s12532-014-0071-1.

Gutjahr, B., Gröll, L., and Werling, M. (2017). Lateral vehicle trajectory optimization using constrained linear time-varying mpc. *IEEE Transactions on Intelligent Transportation Systems*, 18(6), 1586–1595. doi: 10.1109/TITS.2016.2614705.

Probst, M., Wenzel, R., Puphal, T., Komuro, M., Weisswange, T.H., Steinhardt, N., Bolder, B., Flade, B., Sakamoto, Y., Yasui, Y., and Eggert, J. (2021). Automated driving in complex real-world scenarios using a scalable risk-based behavior generation framework. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 629–636. IEEE. doi:10.1109/ITSC48978.2021.9564440.

Weisswange, T.H., Rebhan, S., Bolder, B., Steinhardt, N.A., Joublin, F., Schmuedderich, J., and Goerick, C. (2019). Intelligent traffic flow assist: Optimized highway driving using conditional behavior prediction. *IEEE Intelligent Transportation Systems Magazine*, 13(2), 20–38. doi:10.1109/MITS.2019.2898969.

Wenzel, R. (2018). *Predictive trajectory planning for lag compensation in a longitudinal and lateral vehicle control system*. Master's thesis, Technische Universität Darmstadt, Fachgebiet Fahrzeugtechnik.

Wenzel, R., Probst, M., Puphal, T., Weisswange, T.H., and Eggert, J. (2021). Asymmetry-based behavior planning for cooperation at shared traffic spaces. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, 1008–1015. IEEE. doi:10.1109/IV48863.2021.9575187.