Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 59-26 (2025) 329–334

Parameter Tuning Under Uncertain Road Perception in Driver Assistance Systems

Leon Greiser [*] **Christian Rathgeber** [*] **Vladislav Nenchev** [**]
Sören Hohmann [***]

[*] *BMW Group, 85716 Unterschleissheim, Germany (e-mail: leon.greiser@bmw.de, christian.rathgeber@bmw.de).*
[**] *University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany (e-mail: vladislav.nenchev@unibw.de)*
[***] *Institute of Control Systems, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany (e-mail: soeren.hohmann@kit.edu)*

Abstract: Advanced driver assistance systems have improved comfort, safety, and efficiency of modern vehicles. However, sensor limitations lead to noisy lane estimates that pose a significant challenge in developing performant control architectures. Lateral trajectory planning often employs an optimal control formulation to maintain lane position and minimize steering effort. The parameters are often tuned manually, which is a time-intensive procedure. This paper presents an automatic parameter tuning method for lateral planning in lane-keeping scenarios based on recorded data, while taking into account noisy road estimates. By simulating the lateral vehicle behavior along a reference curve, our approach efficiently optimizes planner parameters for automated driving and demonstrates improved performance on previously unseen test data.

Keywords: Autonomous Vehicles, Trajectory and Path Planning, Parametric Optimization, Control problems under uncertainties, Predictive control

1. INTRODUCTION

Advanced Driver Assistance Systems (ADAS) promise to improve driving comfort, safety, and energy efficiency (Prado et al., 2024). With the increasing number of ADAS features, a unified underlying architecture is often used to reduce complexity and development costs. This architecture typically includes an environment model, which estimates information relevant for the motion of the vehicle, such as the lane center. Based on this, a Trajectory Planner (TP) computes a motion trajectory for the vehicle. The planning problem is typically formulated in Model Predictive Control (MPC) fashion, where the deviation of the states from a reference, as well as the control input, is minimized over a defined time horizon. It is then solved online in a receding horizon manner to adapt to changes in the environment.

Tuning the parameters of an MPC-based TP is often done manually by an expert to achieve the desired driving behavior. The parameters, i.e. the cost function weights, depend not only on the specific vehicle but also on the scenario. With ADAS covering progressively more scenarios, this approach becomes too costly or even infeasible. In addition, MPC has been reported to present challenges in tuning, mainly because one unified parameter set might not get the desired performance across different scenario clusters (Zarrouki et al., 2024).

To save computational costs and simplify the design, the planning and following of the trajectory is often divided into a longitudinal and a lateral part (Rajamani, 2012). This work focuses only on the lateral movement. For lateral planning, the tracked reference is usually the lane center. It is detected using a camera, sometimes in combination with map data. As map data is not always available, accurate lane detection remains a challenging task. Poor light and visibility conditions or degraded markings lead to inaccurate estimates of the lane center (Ding et al., 2020). These inaccuracies, however, result in a suboptimal trajectory with respect to the true lane center. To improve driving behavior, the noisy estimates have to be taken into account when tuning the parameters of the TP.

This paper studies the tuning of the parameters of the existing TP, while retaining its structure. We propose a method to find a set of parameters for lane-keeping scenarios that, given an inaccurately estimated reference, leads as closely as possible to the desired driving behavior. To this end, we make the following contributions.

(1) We derive a TP-agnostic formulation of the parameter optimization problem.
(2) We optimize by re-simulating without noise distribution assumptions and utilize reference trajectories from recorded data instead.
(3) We demonstrate the generalization of the optimized parameters by showing an improved cost on test data.

The remainder of the paper is organized as follows: In Section 1.1 we discuss related work on tracking noisy references. In Section 2 we present the TP formulation used, as well as the general problem statement. In Section 3 we derive our parameter optimization approach. The effectiveness of this approach is shown in Section 4 using real-world data. In Section 5 we present our conclusions.

10.1016/j.ifacol.2025.12.056

1.1 Related Work

The availability of large datasets has driven the interest in data-driven approaches across many domains, including ADAS. A variety of TP approaches have emerged that often rely on neural networks or other black-box architectures (Ganesan et al., 2024; Reda et al., 2024). These approaches result in satisfactory performance in scenarios similar to those included in the training sets but lack explainability (Tampuu et al., 2022) and generalization (Qureshi et al., 2021) to unseen scenarios.

A common approach to deal with noise are robust control approaches such as tube-based (Lee and Jeong, 2024) or min-max (Raimondo et al., 2009) MPC. Although these approaches have strong stability guarantees under bounded noise, they can be conservative and lead to insufficient performance (Liu et al., 2023). Unlike robust MPC, we implicitly leverage the noise properties in the parameter optimization process through the training data. If the probability distribution of the noise is known, stochastic MPC can often be a better performing alternative (Heirung et al., 2018). Probabilistic information on the road course can be integrated into the TP using a target funnel to enhance steering behavior (Bogenberger et al., 2025). In this work, we assume no explicit knowledge of the noise distribution, a requirement for stochastic MPC.

Another technique for dealing with reference signal noise is filtering with simple low-pass filters or more application-specific filters, such as a curvature corrected moving average (Steinecker and Wuensche, 2023). Spline fitting and interpolation are especially common in lane detection systems (Nuthong and Charoenpong, 2010) as these approaches are well suited to remove high-frequency noise components. In our experiments, we use reference data that has already been smoothed using spline fitting.

Due to computational restrictions and transparency, tuning is often preferred over advanced algorithms in practice (Maciejowski, 2009). Therefore, we focus on offline tuning of the cost parameters, leaving the controller structure itself unaltered. Wu et al. (2024) performed parameter optimization of the lateral TP using different local optimization techniques and showed improved performance on the training data. In contrast, we use a global optimization approach and show the generalization of the result by validation with test data. Further, we provide a formulation as a single bilevel optimization problem, making the optimization applicable to different TPs.

2. PROBLEM STATEMENT

We introduce the state representation used for the TP, followed by the TP formulation. Based on this, we present our problem statement.

2.1 Vehicle Kinematics

The state of the vehicle is usually expressed in Cartesian coordinates as the position $(x(t), y(t))$, the velocity $v(t)$, the orientation $\theta(t)$, and the driven curvature $\kappa(t)$. Because embedded hardware in vehicles provides limited computing power, the kinematics are commonly linearized along a reference curve (Gutjahr et al., 2016) as shown in Figure 1. This allows the TP to be divided into longitudinal and lateral parts. Typically, the reference curve used is the lane center, expressed in Euclidean space as $\boldsymbol{r}(s)$.

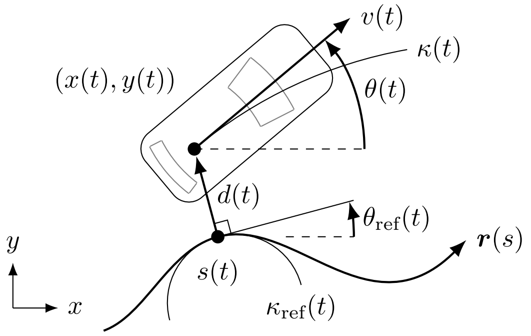

Fig. 1. States with reference $\boldsymbol{r}(s)$ (Gutjahr et al., 2016).

The arc length of the reference curve to the point of the vehicle is $s(t)$. The vector from that point to the vehicle is orthogonal to the reference curve. The length of this vector is equal to the absolute value of the lateral offset $d(t)$. The orientation and curvature of the curve at that point are $\theta_{\mathrm{ref}}(t)$ and $\kappa_{\mathrm{ref}}(t)$. Using these coordinates, the kinematics of the lateral movements can be linearized and discretized (Gutjahr et al., 2016) with sample time T_s. The input is the second derivative of the curvature $u_k = \ddot{\kappa}_k$ to obtain a sufficiently smooth trajectory. The orientation of the reference curve is treated as the disturbance $z_k = \theta_{\mathrm{ref},k}$ to the system. With the state vector $\boldsymbol{x}_k = [d_k, \theta_k, \kappa_k, \dot{\kappa}_k]^\top$ at time step k, the kinematics are

$$\begin{aligned}\boldsymbol{x}_{k+1} = {} & \underbrace{\begin{bmatrix} 1 & v_k T_s & \frac{1}{2} v_k^2 T_s^2 & \frac{1}{6} v_k^2 T_s^3 \\ 0 & 1 & v_k T_s & \frac{1}{2} v_k T_s^2 \\ 0 & 0 & 1 & T_s \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{A}_k} \boldsymbol{x}_k \\ & + \underbrace{\begin{bmatrix} \frac{1}{24} v_k^2 T_s^4 \\ \frac{1}{6} v_k T_s^3 \\ \frac{1}{2} T_s^2 \\ T_s \end{bmatrix}}_{\boldsymbol{B}_k} u_k + \underbrace{\begin{bmatrix} -v_k T_s \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\boldsymbol{D}_k} z_k . \end{aligned} \tag{1}$$

2.2 Lateral Planning

The goal of the TP is to find a feasible and comfortable trajectory. Therefore, the kinematics (1) are used to optimize the inputs over a limited time horizon in an MPC fashion with the goal of minimizing the control input as well as the deviation of the states from a reference. Boundary conditions for control input, safety, or comfort can be formulated as additional constraints. The resulting optimization problem is a Quadratic Programming (QP) problem that can be solved efficiently online. In this work, we discard most of the boundary conditions for simplicity. In this case, the trajectory planning problem at time step k can be formulated using the cost function

$$\hat{J}_{\hat{\boldsymbol{w}}} = \textstyle\sum_{\tau=0}^{N} ||\hat{\boldsymbol{x}}_{k+\tau} - \hat{\boldsymbol{x}}_{\mathrm{des},k+\tau}^{(k)}||_{\hat{\boldsymbol{Q}}_\tau}^2 + \sum_{\tau=0}^{N-1} ||\hat{u}_{k+\tau}||_{\hat{R}_\tau}^2 \tag{2}$$

as the optimization problem

$$
\begin{aligned}
&\hat{u}^{*(k)}|_k^{k+N-1} = \\
&\operatorname*{arg\,min}_{\hat{u}|_k^{k+N-1}} \hat{J}_{\hat{\boldsymbol{w}}}(\hat{\boldsymbol{x}}|_k^{k+N}, \hat{\boldsymbol{x}}_{\text{des}}^{(k)}|_k^{k+N}, \hat{u}|_k^{k+N-1}) \\
&\qquad \text{s.t. } \hat{\boldsymbol{x}}_{k+\tau+1} = \hat{\boldsymbol{A}}_{k+\tau}^{(k)} \hat{\boldsymbol{x}}_{k+\tau} + \hat{\boldsymbol{B}}_{k+\tau}^{(k)} \hat{u}_{k+\tau} + \hat{\boldsymbol{D}}_{k+\tau}^{(k)} \hat{z}_{k+\tau}^{(k)} \\
&\qquad\quad \hat{\boldsymbol{x}}_k = \boldsymbol{x}_k^{(k)} \\
&\qquad\quad u_{\min} \le \hat{u}_{k+\tau} \le u_{\max}
\end{aligned}
\tag{3}
$$

over the planning horizon of N time steps. This formulation assumes that the longitudinal planning problem is already solved at time step k and that the planned velocity, as well as the planned position along the reference curve, is known for each time step in the planning horizon. The system matrix $\hat{\boldsymbol{A}}_{k+\tau}^{(k)}$, input matrix $\hat{\boldsymbol{B}}_{k+\tau}^{(k)}$, and disturbance matrix $\hat{\boldsymbol{D}}_{k+\tau}^{(k)}$ depend on the planned velocity. The time step of planning is expressed in the superscript. The expected disturbance $\hat{z}_{k+\tau}^{(k)}$ depends on the estimated reference curve and the planned positions along that curve. Here, the superscript also expresses the time step of estimation of the reference curve. The desired state $\hat{\boldsymbol{x}}_{\text{des},k+\tau}^{(k)}$ is derived from the reference curve as

$$
\hat{\boldsymbol{x}}_{\text{des},k+\tau}^{(k)} = \begin{bmatrix} 0 & \theta_{\text{ref},k+\tau}^{(k)} & \kappa_{\text{ref},k+\tau}^{(k)} & \dot{\kappa}_{\text{ref},k+\tau}^{(k)} \end{bmatrix}^\top . \tag{4}
$$

The state cost matrix $\hat{\boldsymbol{Q}}_\tau \in \mathbb{R}_{\ge 0}^{4\times 4}$ and the control cost weight $\hat{R}_\tau \in \mathbb{R}_{>0}$ are defined as

$$
\begin{aligned}
\hat{\boldsymbol{Q}}_\tau &= diag([\hat{w}_{d,\tau}\ \hat{w}_{\theta,\tau}\ \hat{w}_{\kappa 0,\tau}\ \hat{w}_{\kappa 1,\tau}]^\top) \\
\hat{R}_\tau &= \hat{w}_{\kappa 2,\tau}.
\end{aligned}
\tag{5}
$$

For simplicity, we limit $\hat{\boldsymbol{Q}}_\tau$ to a diagonal matrix, although off-diagonal entries could be used. Both can vary over the planning horizon and are parametrized by

$$
\begin{aligned}
\hat{\boldsymbol{w}}_\tau &= [\hat{w}_{d,\tau}\ \hat{w}_{\theta,\tau}\ \hat{w}_{\kappa 0,\tau}\ \hat{w}_{\kappa 1,\tau}\ \hat{w}_{\kappa 2,\tau}]^\top ,\ \hat{\boldsymbol{w}}_\tau \in \mathbb{R}_{>0}^5 \\
\hat{\boldsymbol{w}} &= \begin{bmatrix} \hat{\boldsymbol{w}}_0^\top & \hat{\boldsymbol{w}}_1^\top & \dots & \hat{\boldsymbol{w}}_N^\top \end{bmatrix}^\top \text{ with } \hat{\boldsymbol{w}}_\tau = \beta^\tau \hat{\boldsymbol{w}}_0,
\end{aligned}
\tag{6}
$$

which we refer to as the Cost Function Parameters (CFP) in the following. In this paper, we constrain the weights $\hat{\boldsymbol{w}}_\tau$ over the TP horizon using the decay $\beta \in [0, 1]$ to reduce the number of parameters. The state $\boldsymbol{x}_k^{(i)}$ at time step k is relative to the reference curve estimated at time step i. Therefore, the initial planning state $\hat{\boldsymbol{x}}_k = \boldsymbol{x}_k^{(k)}$ is the state at time step k and relative to the reference curve estimated at time step k.

2.3 Problem formulation

We consider a discrete-time system

$$
\boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k u_k + \boldsymbol{D}_k z_k \tag{7}
$$

with the state vector $\boldsymbol{x}_k \in \mathbb{R}^n$ and the input $u_k \in \mathbb{R}$. We aim to find a value for $\hat{\boldsymbol{w}}$ of cost function $\hat{J}_{\hat{\boldsymbol{w}}}$ such that a defined cost $J_{\boldsymbol{w}}(\boldsymbol{x}|_k^{k+M}, \boldsymbol{x}_{\text{des}}|_k^{k+M}, u|_k^{k+M-1})$ over a horizon M is minimal. The state trajectory $\boldsymbol{x}|_k^{k+M}$ is generated from the inputs $u|_k^{k+M-1}$ using (7). The inputs are generated by minimizing the cost function $\hat{J}_{\hat{\boldsymbol{w}}}(\hat{\boldsymbol{x}}|_k^{k+N}, \hat{\boldsymbol{x}}_{\text{des}}^{(k)}|_k^{k+N}, \hat{u}|_k^{k+N-1})$ with (7) over $N < M$ in a receding horizon manner. The noise $\boldsymbol{\eta}_k^{(i)}$, sampled at step i, of the desired state $\hat{\boldsymbol{x}}_{\text{des},k}^{(i)} = \boldsymbol{x}_{\text{des},k} + \boldsymbol{\eta}_k^{(i)}$ is of unknown distribution. The goal is for $\hat{\boldsymbol{w}}$ to compensate for noise that is state-specific and varies over the prediction horizon.

In the context of the lateral TP, we aim to find a value for $\hat{\boldsymbol{w}}$ of the TP cost function $\hat{J}_{\hat{\boldsymbol{w}}}$ that minimizes the distance between the driven trajectory $\boldsymbol{x}|_k^{k+M}$ and the true lane center $\boldsymbol{x}_{\text{des}}|_k^{k+M}$ according to $J_{\boldsymbol{w}}$. To this end, we use recorded data of the inaccurately estimated lane center $\hat{\boldsymbol{x}}_{\text{des},k}^{(i)}$ and the true lane center.

3. OPTIMIZATION-BASED PARAMETER TUNING

In this section, we derive a compact formulation for the discussed parameter optimization problem using a simulation of the kinematics with respect to a reference curve. To this end, we present a method to solve the planning problem with a reference curve that is different from that used for the simulation.

Since we focus on the lateral motion, we also use (1) for simulation. We assume that the tracking error of the planned trajectory is low in comparison to the inaccuracies of the estimated reference curve, as the vehicle dynamics are mostly compensated by a lower-level controller. This way, we can focus on the TP and close the loop using the linearized kinematics (1). This model depends on the velocity. Since only lateral motion is simulated, we assume that all longitudinal states are known *a priori*. For these, as well as for the reference curves, we used recorded data in our experiments in Section 4.

3.1 Switching the Reference Curve of a State

When simulating along a reference curve, states with different reference frames are used. Besides the estimated reference curves for each time step k, i.e. the lane center, that are used to solve the MPC problem, another reference curve is used for simulation. This requires translating the simulation state into a new reference frame for the initial state of the MPC. The only part of the state vector $\boldsymbol{x}_k$ that depends on the reference curve is the lateral offset d_k. To transform the simulation state $\boldsymbol{x}_k^{(sim)}$ to the initial state of the MPC $\boldsymbol{x}_k^{(k)}$, only the lateral offset has to be transformed. We define a state offset to switch the reference curve as

$$
\begin{aligned}
\boldsymbol{x}_k^{(k)} &= \boldsymbol{x}_k^{(sim)} + \Delta \boldsymbol{x}_k^{(sim \to k)} \\
\Delta \boldsymbol{x}_k^{(sim \to k)} &= \begin{bmatrix} \Delta d_k^{(sim \to k)} & 0 & 0 & 0 \end{bmatrix}^\top .
\end{aligned}
\tag{8}
$$

Under the assumption that both curves are sufficiently smooth and parallel, we estimate the change in lateral offset using the small-angle approximation as

$$
\Delta d_k^{(sim \to k)} \approx d_{\text{pt}}^{(k)}(\boldsymbol{r}^{(sim)}(s_k^{(sim)})). \tag{9}
$$

Here, $s_k^{(sim)}$ is the arc length to the vehicle along the simulation reference curve $\boldsymbol{r}^{(sim)}(\cdot)$ at time step k. For a point $\boldsymbol{p}$, $d_{\text{pt}}^{(k)}(\boldsymbol{p})$ is the lateral offset of that point from the reference curve estimated at time step k. Hence, $d_{\text{pt}}^{(k)}(\boldsymbol{r}^{(sim)}(s_k^{(sim)}))$ is the lateral offset of the current point on the simulation reference curve relative to the reference curve estimated at time step k. With the reference curve used for simulation known *a priori* and the longitudinal data, as well as the estimated reference curves obtained from real-world recordings, $\Delta d_k^{(sim\to k)}$ can be calculated in advance of the optimization.

3.2 Optimization Problem Formulation

Equation (8) can be used to simulate the lateral vehicle movement along the true reference curve and solve the planning problem at each time step along the reference curve estimated at that time. This allows us to formulate the parameter optimization problem as

$$\begin{aligned} \min_{\hat{\boldsymbol{w}}_0,\beta} \; & J_{\boldsymbol{w}}(\boldsymbol{x}|_k^{k+M}, \boldsymbol{x}_{\text{des}}|_k^{k+M}, u|_k^{k+M-1}) \\ \text{s.t.} \; & \boldsymbol{x}_{k+1} = \boldsymbol{A}_k \boldsymbol{x}_k + \boldsymbol{B}_k \hat{u}_k^{*(k)} + \boldsymbol{D}_k z_k \\ & (3), (8) \text{ with } \boldsymbol{x}_k^{(sim)} = \boldsymbol{x}_k \end{aligned} \tag{10}$$

using the simulation cost function

$$J_{\boldsymbol{w}} = \textstyle\sum_{k=0}^{M} ||\boldsymbol{x}_k - \boldsymbol{x}_{\text{des},k}||_{\boldsymbol{Q}}^2 + \sum_{k=0}^{M-1} ||\hat{u}_k^{*(k)}||_R^2. \tag{11}$$

The resulting problem is a bilevel optimization problem and is visualized in Figure 2. The upper-level optimization task is to minimize the distance between the simulated states $\boldsymbol{x}_k$ and the desired states $\boldsymbol{x}_{\text{des},k}$. The desired states are chosen analogously to (4) from the simulation reference curve. Since this work focuses on the lane-keeping scenario, we choose the true lane center as the simulation reference curve. For the lower level, there are multiple optimization tasks. Equation (3) has to be solved for every time step k in the simulation horizon M. The optimization variables are the MPC CFP $\hat{\boldsymbol{w}}_0, \beta$ used in (6).

For the upper-level optimization problem, we use the diagonal state cost matrix $\boldsymbol{Q} \in \mathbb{R}_{\geq 0}^{4\times 4}$ and the control cost weight $R \in \mathbb{R}_{>0}$

$$\begin{aligned} \boldsymbol{Q} &= diag([w_d \; w_\theta \; w_{\kappa 0} \; w_{\kappa 1}]^\top) \\ R &= w_{\kappa 2}. \end{aligned} \tag{12}$$

These define the desired behavior with regard to the true reference curve and can be chosen arbitrarily in reasonable bounds. Given that they are defined for a finite receding horizon, we assume this horizon to be sufficiently large, such that the behavior approximately matches a trajectory optimized over the M steps, as done here. $\boldsymbol{Q}$ and R are parametrized by the vector

$$\boldsymbol{w} = [w_d \; w_\theta \; w_{\kappa 0} \; w_{\kappa 1} \; w_{\kappa 2}]^\top, \; \boldsymbol{w} \in \mathbb{R}_{>0}^5, \tag{13}$$

which will be referred to as the Desired Cost Function Parameters (DCFP) in the following.

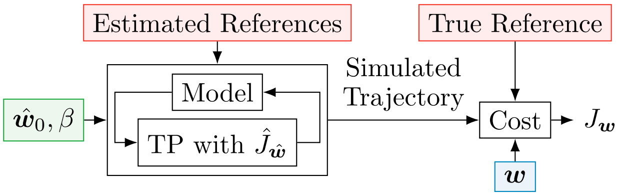

Fig. 2. Block diagram of our bilevel optimization problem formulation. The parameters (green) of the TP are tuned such that the simulated trajectory minimizes $J_{\boldsymbol{w}}$. Reference trajectories (red) and DCFP (blue) are inputs to the optimization.

The lower-level tasks of (10) are convex QPs and can be solved in polynomial time. The upper-level optimization task is non-convex as it contains non-convex constraints.

4. EXPERIMENTAL EVALUATION

We demonstrate the effectiveness of tuning the CFP in simulation, using the optimization problem formulation presented. We optimize multiple sets of CFP on training data, starting with arbitrarily chosen DCFP $\boldsymbol{w}$. We compare the simulation cost using a TP with the optimized CFP and a TP with the unaltered DCFP on a test dataset. In Section 4.1 we first provide details on the implementation and data used. In Section 4.2 we present our results, which we discuss in Section 4.3.

4.1 Implementation

As described in the previous sections, we used data collected on a real-world vehicle. For the reference curve of the TP, we used the lane center of the online estimated road model, which includes a spline fitting. In practice, it is difficult to measure the true lane center. In order to get a good estimate, the vehicle was driven in the center of the lane and the driven trajectory was extracted from the recorded odometry. This data does not include the curvature derivative $\dot{\kappa}(t)$. It was estimated using a Kalman filter followed by a Rauch-Tung-Striebel filter. For the longitudinal data, e.g. the planned and simulated velocity, the recorded trajectory was used as well. All data was interpolated and resampled to $T_s = 0.1\,\text{s}$. A planning horizon of $N = 30$ steps was used.

The dataset we used has a total length of 927 s. It was recorded on various types of roads with varying curvatures and conditions. The velocity ranges from $40\,\text{km}\,\text{h}^{-1}$ to $100\,\text{km}\,\text{h}^{-1}$. For training we used 795 s and for testing 132 s of data. The data consists of multiple continuous sections that vary in length from 6 s to 60 s. Training and test data were split such that their state distributions match.

There are multiple methods to solve a bilevel optimization problem. We used a nested evolutionary algorithm, which is a popular approach. We solved the lower-level problems using the OSQP solver (Stellato et al., 2020). For the upper-level problem, we used differential evolution, a gradient-free heuristic (Storn and Price, 1997).

Because any set of cost function weights can be arbitrarily scaled without affecting the control law, we set $\hat{w}_{\kappa 2,0} = 1$. This leaves $\hat{w}_{d,0}$, $\hat{w}_{\theta,0}$, $\hat{w}_{\kappa 0,0}$, $\hat{w}_{\kappa 1,0}$, and β as optimization variables. We initialized $\beta = 1$ and the remaining optimization variables with the DCFP. We limited $\hat{\boldsymbol{w}}_0 \in [10^{-8}, 10^8]$ for numerical stability and $\beta \in [0.5, 1]$. To evaluate a set of CFP, we calculated the cost for each continuous section of the dataset and added them together.

4.2 Results

Table 1 shows the ten sets of DCFP *A*-*J* that we used. The DCFP were randomly generated. We used various reasonably chosen CFP sets to obtain an estimate of the variance of the error between the simulated and the desired states. We used the inverse of these variances as a set of neutral CFP. We then multiplied these neutral CFP with pseudo-random factors of $[0.25, 4]$.

Table 2 shows the optimized CFP. To improve comparability, we scaled each set to a magnitude similar to DCFP. It can be seen that for $\hat{w}_{d,0}$, $\hat{w}_{\kappa 1,0}$, and $\hat{w}_{\kappa 2,0}$, the weights have similar orders of magnitude. In contrast, outliers with significantly lower values can be observed for $\hat{w}_{\theta,0}$ and $\hat{w}_{\kappa 0,0}$. Most of $\hat{w}_{\kappa 1,0}$ are higher than $w_{\kappa 1}$. This is because the derivative of the curvature has significantly more noise when estimated using a Kalman filter as opposed to the estimate from the fitted spline. For β, all values are higher than 0.97. With this decay, the weights are reduced by 60 % at the end of the TP horizon. Lower values would significantly shorten the effective length of the horizon.

Table 3 shows the simulation cost on the test dataset using the DCFP and the optimized CFP for the TP. The right column further shows the relative change with a negative value showing an improvement. Except for *E*, *F*, and *G*, all optimized CFP show an improvement, with *C* showing the most significant. The average relative change across all ten sets is $-4.13\,\%$.

Figure 3 shows the simulated trajectories for a part of the test data. A TP using the optimized CFP and the DCFP of set *C* are compared. The lateral offset $d(t)$ using the optimized CFP is closer to the desired value most of the time. For the orientation $\theta(t)$, no large deviations can be observed. In the case of the curvature $\kappa(t)$ and curvature deviation $\dot{\kappa}(t)$, the trajectory based on the optimized CFP is smoother. Further, in both cases, the desired states are being followed more accurately. Lastly, in case of the control input $\ddot{\kappa}(t)$, a smaller amplitude can be observed, leading to the overall smoother trajectory.

The average absolute deviation of the states from their desired values over the test dataset are 0.119, 0.005 86, 0.591×10^{-3}, and 1.33×10^{-3}. The maximum absolute deviations are 0.646, 0.0472, 6.59×10^{-3}, and 11.9×10^{-3}. The input values are constrained by the MPC. Although we do not perform a formal stability analysis, these values indicate that with the optimized MPC parameters, the states remain close to their reference over the test dataset.

4.3 Discussion

The results in Section 4.2 show that for most of the chosen DCFP, the optimized CFP show improved performance over the DCFP when used in a TP with an inaccurate reference. While a cost reduction can be expected on the training dataset, we also showed a cost reduction on the test dataset. This indicates a generally better performance of the optimized CFP in scenarios covered by the test data. Further, we have shown that the optimization of the CFP works for a variety of desired cost functions if they are chosen in reasonable bounds. However, for a deeper understanding of the relationship between the desired cost function and performance gain, more DCFP sets would have to be evaluated. If the assumptions posed in this work hold, these improvements can be expected to transfer to the real-world. We matched T_s and N to the real system in this work, although variations may affect performance. While we provide a value for the average cost improvement, its impact on perceived driving dynamics is unclear. With an average optimization runtime of 2.5 h this approach is faster than manual tuning, although it is not yet scalable to large amounts of data.

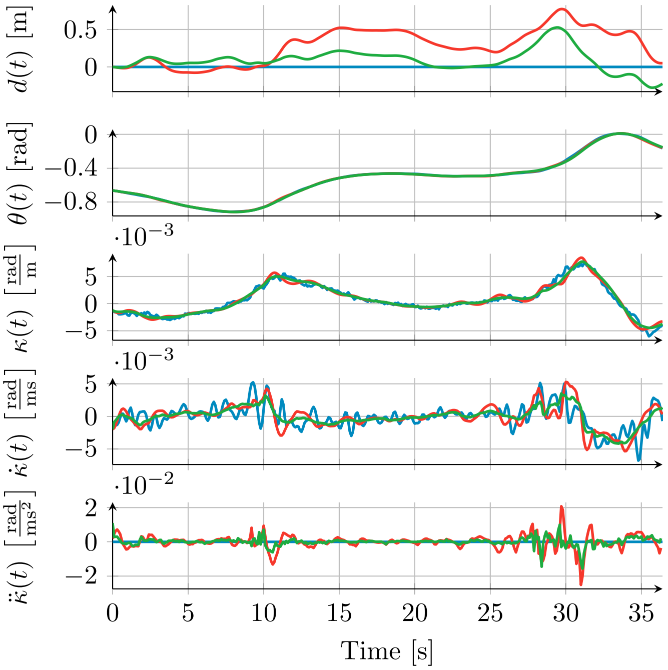

Fig. 3. Simulation on part of the test data using the TP with optimized CFP (green) *C* and with DCFP (red). The desired trajectory (blue), i.e. lane center, is included for reference.

5. CONCLUSIONS

In this work, we studied the optimization of the parameters of the cost function of an optimal controller to compensate for inaccuracies in its reference trajectory. Specifically, we investigated this problem for the lateral trajectory

Table 1. Desired Cost Function Parameters

Set	w_d $\times 10^1$	w_θ $\times 10^4$	$w_{\kappa 0}$ $\times 10^6$	$w_{\kappa 1}$ $\times 10^5$	$w_{\kappa 2}$ $\times 10^4$
A	1.49	3.38	1.61	2.34	0.846
B	0.811	0.284	2.33	2.36	3.91
C	0.557	3.56	2.13	0.803	0.908
D	0.875	0.563	0.906	1.48	1.23
E	2.87	0.356	0.475	1.23	1.94
F	2.84	0.388	0.253	6.19	7.98
G	0.738	0.956	0.233	5.55	1.12
H	7.74	2.08	2.86	5.33	2.88
I	1.55	0.514	2.10	1.20	1.20
J	2.37	0.358	1.95	0.548	8.46

Table 2. Optimized Cost Function Weights

Set	$\hat{\boldsymbol{w}}_{\boldsymbol{d,0}}$ $\times 10^1$	$\hat{\boldsymbol{w}}_{\boldsymbol{\theta,0}}$ $\times 10^4$	$\hat{\boldsymbol{w}}_{\boldsymbol{\kappa 0,0}}$ $\times 10^6$	$\hat{\boldsymbol{w}}_{\boldsymbol{\kappa 1,0}}$ $\times 10^5$	$\hat{\boldsymbol{w}}_{\boldsymbol{\kappa 2,0}}$ $\times 10^4$	$\boldsymbol{\beta}$
A	1.96	0.442	2.58×10^{-6}	6.33	0.143	1.00
B	0.821	0.150	0.278	6.16	1.47	1.00
C	1.29	0.813	7.61×10^{-6}	6.28	0.0637	0.992
D	1.78	0.196	0.202	6.02	0.742	0.997
E	3.53	2.87×10^{-5}	0.126	4.56	1.34	0.976
F	1.96	2.91×10^{-4}	0.228	4.28	3.07	0.974
G	1.34	0.129	0.248	6.76	0.327	0.998
H	2.92	0.156	1.84×10^{-10}	5.48	0.841	0.997
I	1.89	1.58×10^{-7}	0.434	6.22	0.230	1.00
J	1.58	0.0152	0.196	3.74	4.18	1.00

Table 3. Cost $J_{\boldsymbol{w}}$ on Test Dataset

Set	**TP with DCFP**	**TP with optimized CFP**	**Relative Change**
A	7135.3	6480.6	−9.18 %
B	5621.0	5386.5	−4.17 %
C	6923.8	5863.5	−15.32 %
D	3047.7	2968.7	−2.59 %
E	3040.3	3066.5	+0.86 %
F	6810.1	6870.7	+0.89 %
G	4622.1	4688.5	+1.44 %
H	11 622	11 589	−0.28 %
I	4530.8	4266.6	−5.83 %
J	6249.8	5806.9	−7.09 %

planning of a vehicle in a lane-keeping scenario, with the lane center as the reference. We derived a compact bilevel optimization problem formulation for the lateral movement of the vehicle to reduce the value of the desired cost function with respect to the true reference. Using collected real-world data, we performed multiple optimizations with different desired cost functions. With a reduced cost on the test dataset, we showed a general improvement toward the desired driving behavior in simulation. The main advantage of our approach is its practical relevance and simplicity. Instead of adding complexity by modifying the controller itself, it leverages offline optimization on collected data.

REFERENCES

Bogenberger, B., Bürger, J., and Nenchev, V. (2025). Trajectory Planning for Automated Driving using Target Funnels. In *2025 European Control Conference (ECC)*. IEEE, Thessaloniki, Greece.

Ding, L., Zhang, H., Xiao, J., Shu, C., and Lu, S. (2020). A Lane Detection Method Based on Semantic Segmentation. *Computer Modeling in Engineering & Sciences*, 122(3), 1039–1053.

Ganesan, M., Kandhasamy, S., Chokkalingam, B., and Mihet-Popa, L. (2024). A Comprehensive Review on Deep Learning-Based Motion Planning and End-to-End Learning for Self-Driving Vehicle. *IEEE Access*, 12, 66031–66067.

Gutjahr, B., Groll, L., and Werling, M. (2016). Lateral Vehicle Trajectory Optimization Using Constrained Linear Time-Varying MPC. *IEEE Transactions on Intelligent Transportation Systems*, 18(6), 1586–1595.

Heirung, T.A.N., Paulson, J.A., O'Leary, J., and Mesbah, A. (2018). Stochastic model predictive control — how does it work? *Computers & Chemical Engineering*, 114, 158–170.

Lee, T. and Jeong, Y. (2024). A Tube-Based Model Predictive Control for Path Tracking of Autonomous Articulated Vehicle. *Actuators*, 13(5), 164.

Liu, R., Shi, G., and Tokekar, P. (2023). Data-Driven Distributionally Robust Optimal Control with State-Dependent Noise. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 9986–9991. IEEE, Detroit, MI, USA.

Maciejowski, J. (2009). Discussion on: "Min-max Model Predictive Control of Nonlinear Systems: A Unifying Overview on Stability". *European Journal of Control*, 15(1), 22–25.

Nuthong, C. and Charoenpong, T. (2010). Lane detection using smoothing spline. In *2010 3rd International Congress on Image and Signal Processing*, 989–993. IEEE, Yantai, China.

Prado, A.A., Nenchev, V., and Rathgeber, C. (2024). Optimizing Energy-Efficient Braking Trajectories with Anticipatory Road Data for Automated Vehicles. In *2024 European Control Conference (ECC)*, 3280–3285. IEEE, Stockholm, Sweden.

Qureshi, A.H., Miao, Y., Simeonov, A., and Yip, M.C. (2021). Motion Planning Networks: Bridging the Gap Between Learning-Based and Classical Motion Planners. *IEEE Transactions on Robotics*, 37(1), 48–66.

Raimondo, D.M., Limon, D., Lazar, M., Magni, L., and Ndez Camacho, E.F. (2009). Min-max Model Predictive Control of Nonlinear Systems: A Unifying Overview on Stability. *European Journal of Control*, 15(1), 5–21.

Rajamani, R. (2012). *Vehicle Dynamics and Control*. Mechanical Engineering Series. Springer US, Boston, MA.

Reda, M., Onsy, A., Haikal, A.Y., and Ghanbari, A. (2024). Path planning algorithms in the autonomous driving system: A comprehensive review. *Robotics and Autonomous Systems*, 174, 104630.

Steinecker, T. and Wuensche, H.J. (2023). A Simple and Model-Free Path Filtering Algorithm for Smoothing and Accuracy. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, 1–7. IEEE, Anchorage, AK, USA.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S. (2020). OSQP: An operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4), 637–672.

Storn, R. and Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4), 341–359.

Tampuu, A., Matiisen, T., Semikin, M., Fishman, D., and Muhammad, N. (2022). A Survey of End-to-End Driving: Architectures and Training Methods. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4), 1364–1384.

Wu, H.J., Nenchev, V., and Rathgeber, C. (2024). Automatic Parameter Tuning of Self-Driving Vehicles. In *2024 IEEE Conference on Control Technology and Applications (CCTA)*, 555–560. IEEE, Newcastle upon Tyne, United Kingdom.

Zarrouki, B., Spanakakis, M., and Betz, J. (2024). A Safe Reinforcement Learning driven Weights-varying Model Predictive Control for Autonomous Vehicle Motion Control. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, 1401–1408. IEEE, Jeju Island, Korea, Republic of.