Bilevel Inverse Optimal Control for Lane-Change Costs in Automated Driving

Leon Greiser[1], Christian Rathgeber[1], Vladislav Nenchev[2], and Sören Hohmann[3]

Abstract— **Manual tuning of trajectory planners for advanced driver assistance systems is time-consuming and difficult to scale. Learning cost functions from demonstrations offers a viable data-driven alternative, but real-world lane-change data is affected by noisy measurements and missing reliable temporal segmentation methods. This paper studies the learning of quadratic cost functions for a linear parameter-varying kinematic model in lane-change scenarios from real-world driving data. We adapt an inverse optimal control-based segmentation approach to detect the lane-change start and use it to calibrate a heuristic segmentation rule. For cost function learning, we use bilevel inverse optimal control to accommodate incomplete state observations with statistical consistency under measurement noise and apply it to a large dataset using stochastic gradient descent. We also learn a parameter-varying cost function with the parameters represented by a neural network. The results show that both the curvature reference and the number of optimized entries in the cost matrices affect reconstruction, and that parameter-varying cost functions further improve reconstruction accuracy.**

I. INTRODUCTION

Advanced Driver Assistance Systems (ADAS) are common features in modern vehicles, supporting the driver in various scenarios [1]. The Trajectory Planner (TP) is a key component in many ADAS, with the goal of providing a safe and comfortable motion trajectory. It is often formulated as an online optimization problem [2]. In order to solve it efficiently, the longitudinal and lateral motion of the vehicle are commonly divided into separate planning problems [2]. This work focuses on the lateral movement in lane-change scenarios.

In optimization-based approaches, the cost function of the TP is the primary means of defining the driving style. In practice, the cost function parameters are manually tuned by experts. As parameters are often scenario-specific, this process is time-consuming, expensive, and does not scale. With ADAS covering more and more scenarios, the method of manual tuning becomes infeasible.

An automated approach to tuning the TP includes learning the cost function in an automated way. Either, to be used directly inside the TP, or indirectly for tuning [3]. Inverse Optimal Control (IOC) provides a method to learn a cost function directly from demonstration data.

[1]Leon Greiser and Christian Rathgeber are with BMW Group, 85716 Unterschleissheim, Germany {leon.greiser, christian.rathgeber}@bmw.de

[2]Vladislav Nenchev is with the University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany vladislav.nenchev@unibw.de

[3]Sören Hohmann is with the Institute of Control Systems, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany soeren.hohmann@kit.edu

We consider the problem of learning a quadratic stage cost function for a Linear Parameter-Varying (LPV) kinematic model from human-driven lane-change trajectories. The goal of this work is not to learn the best possible policy, but to study learning quadratic cost functions for a fixed LPV model. The main challenges are the noisy measurements due to the ill-conditioned state estimation problem and the lack of temporal segmentation of the driving data. We investigate whether bilevel IOC with initial state optimization can be used to learn a shared cost function from real-world data. Further, we study how well it reconstructs unseen trajectories on the position level and whether a parameter-varying cost function improves reconstruction accuracy.

To this end, we make the following contributions:

1) Propose a bilevel IOC framework for learning quadratic cost functions for reconstructing lane-change behavior from demonstrations.
2) Adapt IOC-based segmentation to lane-change start detection and use it to calibrate a scalable heuristic segmentation rule.
3) Empirically show that parameter-varying cost functions improve trajectory reconstruction accuracy.
4) Quantify the impact of cost-design choices such as the curvature reference and the number of optimized cost-matrix entries on trajectory reconstruction accuracy.

The remainder of this work is structured as follows. First, we position our paper relative to existing IOC methods in Section I-A. We describe the used LPV system and optimal control formulation in Section II. In Section III we present our methods for lane-change segmentation and cost function learning. We explain our experimental setup in Section IV and present the results in Section V. Finally, we draw our conclusion in Section VI.

A. Related Work

IOC [4] and Inverse Reinforcement Learning (IRL) [5] have previously been used to infer cost or reward functions from driving data. Within IRL, prior work focused on reward learning in the continuous domain using motion-planning-based sampling [6] and the discretized setting using a polynomial trajectory sampler [7]. IRL has further been used for the lane-change maneuver with style-grouped demonstrations [8]. In contrast to these works, we focus on learning structured quadratic cost functions for a kinematic model directly from human-driven lane-change trajectories.

IOC has been considered for longitudinal human driving behavior [9], prediction of autonomous race cars [10], and online cost inference of human-driven neighboring vehicles [11]. In contrast to these works, we focus on learning shared and parameter-varying quadratic cost functions for lane-change maneuvers. IOC has also been studied for steering behavior along a reference trajectory [12]. In contrast, we study whether bilevel IOC can learn cost functions for human-driven lane-change maneuvers from noisy, weakly segmented real-world data.

Higher-order states of vehicle kinematics are hard to measure and result in an ill-conditioned state estimation problem. In this work, we use bilevel IOC [13], which works without complete state measurements and has been shown to be statistically consistent under measurement noise [14].

Performing IOC on lane-change trajectories requires determining the start point of the maneuver. Many works use heuristics based on the lateral offset, lateral velocity, and lateral acceleration [15]–[17]. However, the choice of parameters influences the segmentation. In this work, we adapt IOC-based segmentation such as [18] to the lane-change maneuver to infer the parameters for a simple heuristic.

II. Preliminaries

We introduce the system for the lateral vehicle kinematics, followed by the lateral planning problem formulation. We further present the IOC method utilized in our framework.

A. Lateral Vehicle Kinematics Model

As this work is motivated by the parameter tuning of a TP, we choose the kinematic model and cost function accordingly. The planning problem is commonly formulated as a Quadratic Programming (QP) problem with a LPV system model and a quadratic cost function [2]. For the vehicle kinematic model, we use the discrete-time, LPV system from [3]

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

linearized along the curve $\gamma(s)$ as shown in Fig. 1. The curve is usually a path along the lane center, parameterized by the arc length s. The kinematic state vector $\boldsymbol{x}_k = [d_k \; \theta_k \; \kappa_k \; \dot{\kappa}_k]^\top$ at time step k is composed of the lateral offset d_k, heading θ_k, curvature κ_k, and curvature rate $\dot{\kappa}_k$. The control input $u_k = \ddot{\kappa}_k$ is the second derivative of the curvature. The heading of the curve $\theta_{\gamma,k}$ at the current arc length s_k acts as a disturbance $z_k = \theta_{\gamma,k}$. The arc length s_k is assumed to be known. For the TP this means that the longitudinal planning problem is solved prior to the lateral one.

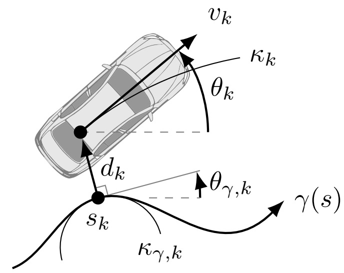

Fig. 1. Lateral vehicle kinematics model along the reference curve [2].

B. Lateral Planning Problem

Most trajectory planning problems include a set of safety constraints [2]. We disregard these constraints, which leaves us with an analytical solution to the planning problem. For the lateral planning, this results in the optimization problem

$$\begin{aligned}\min_{u_{0:N-1}} \; & \textstyle\sum_{k=0}^{N} ||\boldsymbol{x}_k - \boldsymbol{x}_{\text{ref},k}||^2_{\boldsymbol{Q}_k^{\boldsymbol{\vartheta}}} + \sum_{k=0}^{N-1} ||u_k||^2_{R_k^{\boldsymbol{\vartheta}}} \\ \text{s.t. } & (1), \forall k \in [0, N-1], \\ & \boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}}\end{aligned} \tag{2}$$

with notation $||\boldsymbol{v}||^2_{\boldsymbol{M}} = \boldsymbol{v}^\top \boldsymbol{M} \boldsymbol{v}$. The reference state $\boldsymbol{x}_{\text{ref},k}$ is defined as $\boldsymbol{x}_{\text{ref},k} = [d_{\text{ref},k} \; \theta_{\text{ref},k} \; \kappa_{\text{ref},k} \; \dot{\kappa}_{\text{ref},k}]^\top$ with the reference offset $d_{\text{ref},k} = 0$ and the reference heading $\theta_{\text{ref},k} = \theta_{\gamma,k}$. The reference curvature $\kappa_{\text{ref},k}$ can be set to zero or to the curvature of the curve $\kappa_{\gamma,k}$. This choice is further elaborated in Section IV-B. The reference curvature rate $\dot{\kappa}_{\text{ref},k}$ is also set to zero. The state cost matrix $\boldsymbol{Q}_k^{\boldsymbol{\vartheta}} \in \mathbb{R}^{4\times4}_{\geq 0}$ and control cost matrix $R_k^{\boldsymbol{\vartheta}} \in \mathbb{R}_{>0}$ are parametrized by the cost function parameter vector $\boldsymbol{\vartheta} \in \mathbb{R}^p$.

C. Bilevel Inverse Optimal Control

Consider system (1) with the optimal state and control trajectories $\tilde{\boldsymbol{x}}_{0:N}$ and $\tilde{u}_{0:N-1}$. We formulate the optimization problem of the bilevel IOC method following [13] as

$$\begin{aligned}\min_{\boldsymbol{\vartheta}, \boldsymbol{x}_{\text{init}}} \; & \textstyle\sum_{k=0}^{N} ||\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k||^2_{\tilde{\boldsymbol{Q}}} + \sum_{k=0}^{N-1} ||u_k - \tilde{u}_k||^2_{\tilde{R}} \\ \text{s.t. } & (2)\end{aligned} \tag{3}$$

with state reconstruction cost matrix $\tilde{\boldsymbol{Q}} \in \mathbb{R}^{4\times4}_{\geq 0}$, and control reconstruction cost matrix $\tilde{R} \in \mathbb{R}_{\geq 0}$ as hyperparameters. Note that the initial state $\boldsymbol{x}_{\text{init}}$ is optimized along the cost function parameters.

III. Method

In this section we derive a framework to learn quadratic cost functions from real-world noisy and weakly labeled demonstration data for the lane-change scenario. To this end we adapt IOC-based segmentation to lane changes and present a method to perform bilevel IOC on a large dataset.

A. Lane-Change Segmentation

Common datasets for autonomous driving such as [19] do not include exact temporal segmentation of the lane-change maneuvers. Start and end times of lane changes need to be inferred from the data. We adapt IOC-based segmentation to the lane-change maneuver and use it to tune the parameters of a simple heuristic.

We assume a trajectory containing a lane-change maneuver with sufficient temporal margin is composed of two optimal control problems. The first one is a lane-keeping problem, where the reference state is the lane center of the departure lane. The second one is a lane-change problem followed by a lane-keeping problem, where the reference state is the lane center of the target lane. To determine the time step $\underline{k}^*$ for the lane-change start, we aim to find the $\underline{k}$ that minimizes the combined IOC reconstruction cost of the two problems with common cost matrices. This can be formulated as

$$
\begin{aligned}
\min_{\boldsymbol{\vartheta},\boldsymbol{x}_{\text{init}},\underline{k}} & \textstyle\sum_{k=0}^{N} ||\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k||^2_{\tilde{\boldsymbol{Q}}} + \sum_{k=0}^{N-1} ||u_k - \tilde{u}_k||^2_{\tilde{R}} \\
\text{s.t.} & \min_{u_{0:\underline{k}-1}} \textstyle\sum_{k=0}^{\underline{k}-1}||\boldsymbol{x}_k - \boldsymbol{x}_{\text{ref},k}||^2_{\boldsymbol{Q}_k^{\boldsymbol{\vartheta}}} + \sum_{k=0}^{\underline{k}-1}||u_k||^2_{R_k^{\boldsymbol{\vartheta}}}, \\
& \quad \text{s.t. } (1), \forall k \in [0, \underline{k}-1], \\
& \quad\quad \boldsymbol{x}_0 = \boldsymbol{x}_{\text{init}} \\
& \min_{u_{\underline{k}:N-1}} \textstyle\sum_{k=\underline{k}}^{N}||\boldsymbol{x}_k - \boldsymbol{x}_{\text{ref},k}||^2_{\boldsymbol{Q}_k^{\boldsymbol{\vartheta}}} + \sum_{k=\underline{k}}^{N-1}||u_k||^2_{R_k^{\boldsymbol{\vartheta}}}, \\
& \quad \text{s.t.} (1), \forall k \in [\underline{k}, N-1].
\end{aligned} \tag{4}
$$

The system is linearized along the lane center of the target lane. Therefore, the disturbance z_k corresponds to the heading of the target lane. The reference state $\boldsymbol{x}_{\text{ref},k}$ is set with heading and curvature from the departure lane for $k \in [0, \underline{k}-1]$ and from the target lane for $k \in [\underline{k}, N]$. For $k \in [0, \underline{k}-1]$ we further set $d_{\text{ref},k}$ to the lateral distance between the target and departure lane.

For the cost matrices, we only optimize the diagonal entries. We parametrize them as

$$
\begin{aligned}
\boldsymbol{Q}_k^{\boldsymbol{\vartheta}} &= \operatorname{diag}(\boldsymbol{\psi}_{\boldsymbol{Q},k}) \operatorname{diag}(\boldsymbol{\psi}_{\boldsymbol{Q},k})^\top \\
R_k^{\boldsymbol{\vartheta}} &= \operatorname{diag}(\boldsymbol{\psi}_{R,k}) \operatorname{diag}(\boldsymbol{\psi}_{R,k})^\top
\end{aligned} \tag{5}
$$

with $\boldsymbol{\psi}_{\boldsymbol{Q},k} \in \mathbb{R}^4$, $\boldsymbol{\psi}_{R,k} \in \mathbb{R}$, and the constant parameters $[\boldsymbol{\psi}_{\boldsymbol{Q},k}^\top \ \boldsymbol{\psi}_{R,k}^\top]^\top = \boldsymbol{\vartheta}$ to guarantee positive semidefiniteness of the cost matrices. For numerical stability, we constrain $\operatorname{tr}(\boldsymbol{Q}_k^{\boldsymbol{\vartheta}}) + \operatorname{tr}(R_k^{\boldsymbol{\vartheta}}) = 1$.

Solving (4) is computationally expensive. Instead of solving it for every trajectory in the dataset, we solve it for a subset and use the results to tune the parameters of the following heuristic. Given the current ego lane, we determine the time when a certain lateral offset is last crossed before the ego lane changes, i.e. the center between the lanes. Additionally, we add a temporal margin. The tunable parameters are the lateral offset threshold and the temporal margin.

Note that (4) cannot be applied to the end of the lane change analogously. The end of the lane change is not clearly defined, as the target lane stays the reference, even when it has been reached. We use the same parameters and the mirrored heuristic for the lane-change end.

B. Inverse Optimal Control with Measurement Noise

For system (1), higher-order states are difficult to measure. They can be estimated, however, the state estimation is ill-conditioned as the system acts as an integrator chain. For this reason, we use bilevel IOC as described in Section II-C, which works with incomplete state observations and has been shown to be statistically consistent under measurement noise [14]. Optimizing the initial state allows us to not rely on the higher-order state estimates. We focus on the trajectory reconstruction at the position as the most accurately estimated state and additionally use a large dataset.

Adapting (3) to multiple trajectories through cumulative cost and shared cost function parameters is straightforward. However, solving the resulting optimization problem for a large number of trajectories is computationally expensive. We therefore use Gradient Descent (GD). Solving (2) analytically with the discrete-time algebraic Riccati equation allows us to efficiently compute the gradient through the inner optimization problem.

For the cost matrices, only the diagonal elements or the full matrices can be optimized. For the case where only the diagonals are optimized, we parametrize $\boldsymbol{Q}_k^{\boldsymbol{\vartheta}}$ and $R_k^{\boldsymbol{\vartheta}}$ using (5). For the case where the full cost matrices are optimized, we parametrize

$$
\begin{aligned}
\boldsymbol{Q}_k^{\boldsymbol{\vartheta}} &= \operatorname{ltmat}(\boldsymbol{\psi}_{\boldsymbol{Q},k}) \operatorname{ltmat}(\boldsymbol{\psi}_{\boldsymbol{Q},k})^\top \\
R_k^{\boldsymbol{\vartheta}} &= \operatorname{ltmat}(\boldsymbol{\psi}_{R,k}) \operatorname{ltmat}(\boldsymbol{\psi}_{R,k})^\top
\end{aligned} \tag{6}
$$

with $\boldsymbol{\psi}_{\boldsymbol{Q},k} \in \mathbb{R}^{10}$, $\boldsymbol{\psi}_{R,k} \in \mathbb{R}$, and $\operatorname{ltmat}(\cdot)$ denoting the transformation of a vector into a lower triangular matrix.

To use trajectories of different lengths and prevent long trajectories from dominating the reconstruction costs, we average the stage costs over the trajectory length. We define the reconstruction loss as

$$
\mathcal{L}_{\text{rec}} = \tfrac{1}{N+1} \textstyle\sum_{k=0}^{N}||\boldsymbol{x}_k - \tilde{\boldsymbol{x}}_k||^2_{\tilde{\boldsymbol{Q}}} + \tfrac{1}{N} \sum_{k=0}^{N-1}||u_k - \tilde{u}_k||^2_{\tilde{R}}. \tag{7}
$$

To ensure numerical stability, we define the normalization loss as

$$
\mathcal{L}_{\text{norm}} = \left(\tfrac{1}{N+1} \textstyle\sum_{k=0}^{N}\operatorname{tr}(\boldsymbol{Q}_k^{\boldsymbol{\vartheta}}) + \tfrac{1}{N} \sum_{k=0}^{N-1}\operatorname{tr}(R_k^{\boldsymbol{\vartheta}}) - 1\right)^2. \tag{8}
$$

The per-trajectory loss follows as $\mathcal{L} = \mathcal{L}_{\text{rec}} + \alpha\mathcal{L}_{\text{norm}}$ with the normalization loss factor $\alpha \in \mathbb{R}_{\geq 0}$ as a tunable hyperparameter.

C. Parameter-Varying Cost Function

The weights of the quadratic stage cost function can be learned not only as static parameters, but also as functions of features additional to the lateral kinematic states. This gives the cost function more flexibility. We model the parameter-varying stage cost function $[\boldsymbol{\psi}_{\boldsymbol{Q},k}^\top \ \boldsymbol{\psi}_{R,k}^\top]^\top = \boldsymbol{h}_{\boldsymbol{\vartheta}}(\boldsymbol{\phi}_k)$ with a small feedforward neural network $\boldsymbol{h}_{\boldsymbol{\vartheta}}(\cdot)$, detailed in Section IV-C, with feature vector $\boldsymbol{\phi}_k$ as input. The feature vector can contain different features that do not depend on the lateral kinematic states, such as longitudinal states or road features. The neural network is trained end-to-end using the same loss as described in Section III-B.

IV. Experimental Setup

We present the used dataset, the considered cost function training configurations and the evaluation metric. Further, we display our implementation details and hyperparameters.

A. Dataset

For our experiments, we used the nuPlan dataset [19], which contains urban driving data from four cities. We filtered and estimated the kinematic states using an extended Kalman filter with a Rauch-Tung-Striebel filter. For lane matching we used a hidden Markov model. For the lane-change segmentation, we kept the original sample rate of $T = 0.05\,\mathrm{s}$. For all other experiments, we subsampled to $T = 0.1\,\mathrm{s}$ to reduce the computational cost. We further filtered outliers based on duration, initial and maximum lateral offset. In total, we extracted $11\,327$ lane changes, which we split into a train set of 9515, a validation set of 836, and a test set of 976 samples. Fig. 2 shows the lateral offset d_k of four trajectories from the test set. Fig. 3 shows the estimated curvature κ_k of the same four trajectories. While the trajectories are similar on the position level, large differences can be seen for the curvature. This visualizes the ill-conditioned state estimation problem where very different higher-order states can produce similar behavior on the position level.

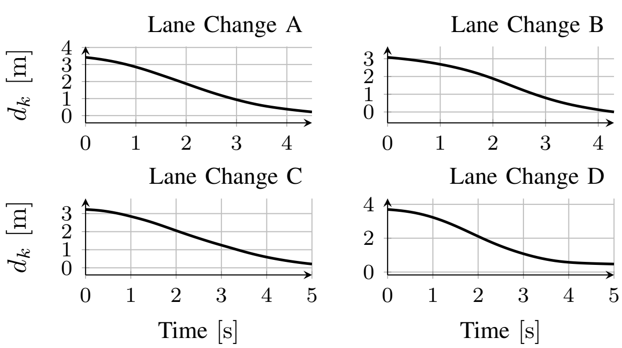

Fig. 2. Lateral offset of example trajectories from the test set.

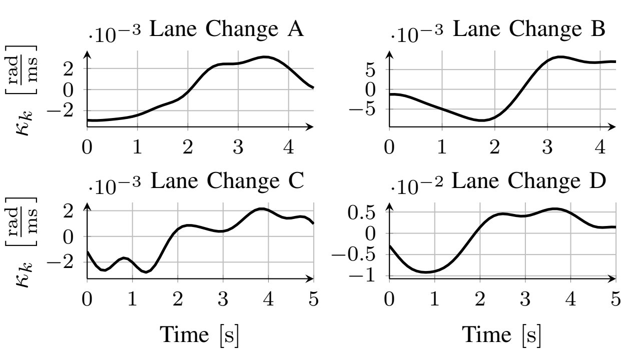

Fig. 3. Curvature of example trajectories from the test set.

B. Cost Function Structure

Regarding the structure of the cost function, there are two key design choices that we study in this work. The choice of state reference and the degree of freedom of the cost matrices. While the choice of reference is straightforward for the lateral offset and heading, for the reference of the curvature there are two options as described in Section II-B. The first one is to use the curvature of the reference curve in order to reduce the tracking error. The second one is to set the curvature reference to zero, penalizing large curvatures with large lateral accelerations and prioritizing comfort. As map data usually does not include higher-order derivatives, the reference for the curvature rate is set to zero. For the cost matrices, the diagonal elements are more intuitive and often the focus of manual tuning, while the off-diagonal elements increase the degree of freedom. We trained each of the four possible structure combinations with ten different random seeds.

C. Parameter-Varying Cost Function

To investigate whether a parameter-varying cost function can improve the reconstruction accuracy, we trained the cost function with all possible feature combinations. As features we used the longitudinal velocity, longitudinal acceleration, and the curvature of the reference curve. All features were normalized to mean zero and a standard deviation of one on the training set. For the neural network, we used a perceptron with two hidden layers of 64 neurons and Sigmoid Linear Unit (SiLU) activation functions. Again, we trained each of the eight possible feature combinations with ten different random seeds.

D. Metrics

Due to the ill-conditioned state estimation problem and the resulting lack of precise higher-order state estimates, we focused on reconstruction at the position level and set $\tilde{\boldsymbol{Q}} = \mathrm{diag}([1\ 0\ 0\ 0])$ and $\tilde{R} = 0$. We used the Root Mean Square Error (RMSE) of the lateral offset, corresponding to $\sqrt{\mathcal{L}_{\mathrm{rec}}}$, as the main metric in the following. Since we optimized the initial states during training, we did the same for testing. Therefore, we report the RMSE of the lateral offset after optimizing (3) with the learned cost function fixed. For consistency, we did this for both the train and test set.

E. Implementation Details

To get a smoother gradient and to reduce computation time, we used mini-batch GD with the AdamW optimizer in PyTorch. For the evaluation we solved (3) using CasADi with the IPOPT solver. After performing a hyperparameter search, we trained for 200 epochs with a batch size of 32 and a learning rate of 1.3×10^{-4}. We set the normalization loss factor to $\alpha = 1.5 \times 10^{-6}$.

V. Results

In this section we present the results of tuning the lane-change start detection heuristic using IOC-based segmentation. We further show the reconstruction accuracy of learned cost functions for the training configurations described in Section IV and discuss the findings.

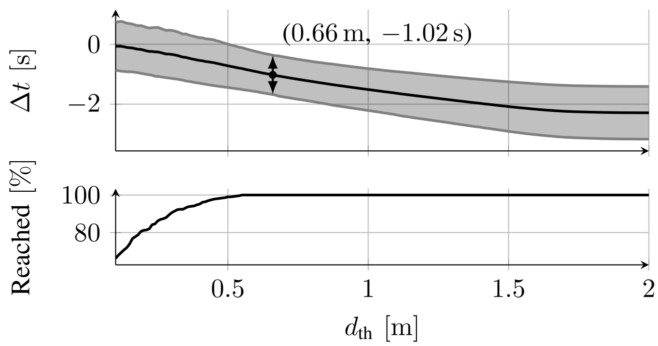

Fig. 4. Tuning of lane-change start detection heuristic parameters.

A. Lane-Change Segmentation

For the tuning of the lane-change segmentation heuristic, we optimized the lane-change start $\underline{k}$ using (4). We did this for a subset of 500 lane changes to keep the computation time reasonable. The upper axis of Fig. 4 shows the mean and standard deviation of the temporal delta $\Delta t = T(\underline{k}^* - k_{\text{th}})$ between the optimized start point $\underline{k}^*$ and the point k_{th} of crossing lateral offset threshold d_{th}. The lower axis shows the percentage of trajectories that reach the corresponding lateral offset threshold. To get the optimal parameters for the heuristic, we chose the threshold with the smallest Δt standard deviation as $d_{\text{th}} = 0.66\,\text{m}$ and the temporal margin as the average Δt at that threshold as $\Delta t^* = -1.02\,\text{s}$. We used these parameters for lane-change segmentation for all other experiments.

B. Cost Function Structure

TABLE I
IMPACT OF COST FUNCTION STRUCTURAL CHOICES ON THE RECONSTRUCTION RMSE

Optimized Elements	$\kappa_{\text{ref},k}$	RMSE [cm] Train	RMSE [cm] Test
diagonal	$\kappa_{\gamma,k}$	5.943 ± 0.009	6.030 ± 0.009
diagonal	0	6.246 ± 0.007	6.268 ± 0.007
full matrix	$\kappa_{\gamma,k}$	$\mathbf{5.462 \pm 0.005}$	$\mathbf{5.495 \pm 0.005}$
full matrix	0	5.768 ± 0.005	5.705 ± 0.005

Table I shows the RMSE of the reconstructed lateral offset for each of the four structure configurations. RMSE average and standard deviation are shown for train and test data. The first two columns contain the number of optimized elements and the reference for the curvature.

The RMSE on the test set varies only slightly from the one on the train set, indicating little overfitting. Setting the curvature reference to the curvature of the reference curve reduced the test RMSE by around $0.2\,\text{cm}$ compared to setting it to zero. Optimizing the full cost matrices instead of only the diagonal elements reduced the test RMSE by around $0.5\,\text{cm}$. With the largest standard deviation of less than $0.01\,\text{cm}$, these differences indicate an impact of both design choices on the reconstruction accuracy.

C. Parameter-Varying Cost Function

TABLE II
IMPACT OF FEATURE CHOICE OF PARAMETER-VARYING COST FUNCTION ON THE RECONSTRUCTION RMSE

Velocity	Acceleration	Curvature	RMSE [cm] Train	RMSE [cm] Test
no	no	no	5.460 ± 0.006	5.493 ± 0.006
no	no	yes	4.552 ± 0.030	4.547 ± 0.038
no	yes	no	4.604 ± 0.038	4.719 ± 0.040
no	yes	yes	4.530 ± 0.057	$\mathbf{4.454 \pm 0.110}$
yes	no	no	4.602 ± 0.079	4.612 ± 0.070
yes	no	yes	4.588 ± 0.037	4.562 ± 0.059
yes	yes	no	$\mathbf{4.510 \pm 0.038}$	4.511 ± 0.062
yes	yes	yes	4.558 ± 0.075	4.488 ± 0.143

Table II shows the RMSE of the reconstructed lateral offset for each of the cost function feature combinations. The first three columns show which features were used in the cost function. For the cost function structure, we used the configuration with the lowest test RMSE from Section V-B.

Again, the difference in RMSE between the train and test set is small. However, most of the feature combinations did not show a noticeable difference in comparison, especially considering the standard deviation. The only visibly larger RMSE resulted from not using any features. This shows that adding features to the cost function can improve the reconstruction accuracy, but the specific choice of features has a smaller impact.

For the best run, i.e. using all features, Fig. 5 shows a reconstructed test trajectory. While the reconstruction matches the demonstration position closely, higher-order states show larger deviations. This can either be due to the state estimation error or suboptimality of the human demonstration data, the distinguishing of which requires accurate higher-order state estimates.

D. Discussion

The results in Section V-A show that it is possible to tune the parameters of a heuristic for lane-change segmentation using IOC-based optimization. The temporal margin indicates that the lane-change starts approximately one second before the threshold of $0.66\,\text{m}$ is crossed. Although these parameters minimize the temporal error of the used heuristic, a heuristic with more parameters could have the potential to further reduce this error.

The results in Section V-B show that the choice of curvature reference and number of optimized elements affect the reconstruction accuracy, with the latter having a slightly larger impact. This suggests that human drivers prefer matching the road curvature over minimizing lateral acceleration in lane-change maneuvers. Additionally, the cross-coupling terms in the cost matrices are non-negligible. This shows a disadvantage of manual tuning, which is often only focused on the diagonal cost matrix elements.

The results in Section V-C show that a parameter-varying cost function can outperform a static one in terms of reconstruction accuracy. However, the similar performance across

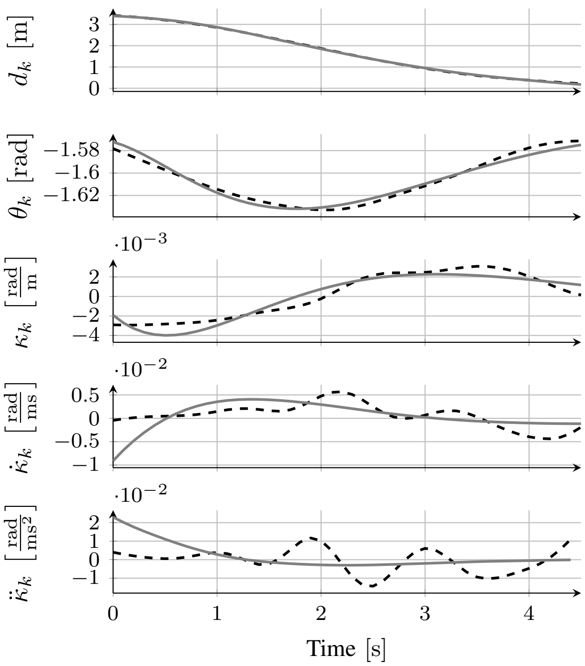

Fig. 5. Comparison between a trajectory (dashed) from the test set and its reconstruction (gray) using the learned cost function.

the different combinations suggests redundancy between the features. A plausible reason for this is that velocity, acceleration, and curvature act as proxies for the same driving context, e.g. a driver decelerating before a sharp turn.

The approach used in this work is limited to unconstrained LPV systems with quadratic costs. When considering constraints, the Riccati equation could be replaced with differentiable model predictive control. We further restrict the reconstruction metric to lateral position error. Optimizing each lane-change initialization instead of using the heuristic could improve the demonstration data quality, although it is computationally more expensive.

VI. Conclusion

In this work, we studied the use of bilevel IOC to learn quadratic cost functions for a LPV kinematic model from real-world human lane-change trajectories. We showed how IOC-based segmentation can be used to tune the parameters of a heuristic for lane-change start detection. We further performed an empirical evaluation of the impact of key structural choices of the cost function and of learning a parameter-varying cost function on the reconstruction accuracy. The results show that both the choice of curvature reference as well as the number of optimized elements in the cost matrices have an impact. Learning a parameter-varying cost function can further reduce the reconstruction error to a test RMSE of approximately $4.5\,\mathrm{cm}$. The specific choice of feature combination has a minor impact. Future work could investigate the use of more complex heuristics for lane-change segmentation or the use of soft-optimality approaches to account for suboptimality of demonstration data.

References

[1] A. A. Prado, V. Nenchev, and C. Rathgeber, “Optimizing Energy-Efficient Braking Trajectories with Anticipatory Road Data for Automated Vehicles,” in *2024 European Control Conference (ECC)*. Stockholm, Sweden: IEEE, June 2024, pp. 3280–3285.

[2] B. Gutjahr, L. Groll, and M. Werling, “Lateral Vehicle Trajectory Optimization Using Constrained Linear Time-Varying MPC,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1586–1595, 2017.

[3] L. Greiser, C. Rathgeber, V. Nenchev, and S. Hohmann, “Parameter Tuning Under Uncertain Road Perception in Driver Assistance Systems,” *IFAC-PapersOnLine*, vol. 59, no. 26, pp. 329–334, 2025.

[4] R. E. Kalman, “When Is a Linear Control System Optimal?” *Journal of Basic Engineering*, vol. 86, no. 1, pp. 51–60, Mar. 1964.

[5] A. Y. Ng and S. Russell, “Algorithms for inverse reinforcement learning.” in *Proceedings of the 17th International Conference on Machine Learning (ICML)*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 663–670.

[6] Z. Wu, L. Sun, W. Zhan, C. Yang, and M. Tomizuka, “Efficient Sampling-Based Maximum Entropy Inverse Reinforcement Learning With Application to Autonomous Driving,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5355–5362, Oct. 2020.

[7] Z. Huang, J. Wu, and C. Lv, “Driving Behavior Modeling Using Naturalistic Human Driving Data With Inverse Reinforcement Learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 239–10 251, Aug. 2022.

[8] J. Liu, L. N. Boyle, and A. G. Banerjee, “An Inverse Reinforcement Learning Approach for Customizing Automated Lane Change Systems,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9261–9271, Sept. 2022.

[9] L. Guo and Y. Jia, “Inverse Model Predictive Control (IMPC) Based Modeling and Prediction of Human-Driven Vehicles in Mixed Traffic,” *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 3, pp. 501–512, Sept. 2021.

[10] R. Reiter, F. Messerer, M. Schratter, D. Watzenig, and M. Diehl, “An Inverse Optimal Control Approach for Trajectory Prediction of Autonomous Race Cars,” in *2022 European Control Conference (ECC)*. London, United Kingdom: IEEE, July 2022, pp. 146–153.

[11] V. Bhattacharyya and A. Vahidi, “Automated Lane Change via Adaptive Interactive MPC: Human-in-the-Loop Experiments,” *IEEE Transactions on Control Systems Technology*, vol. 33, no. 4, pp. 1246–1257, July 2025.

[12] I. Kuchkarov, G. Mitiai, O. Petrosian, T. Lepikhin, J. Inga, and S. Hohmann, “Inverse optimal control with continuous updating for a steering behavior model with reference trajectory,” in *Mathematical Optimization Theory and Operations Research: Recent Trends*. Cham: Springer International Publishing, 2021, pp. 387–402.

[13] T. L. Molloy, D. Tsai, J. J. Ford, and T. Perez, “Discrete-time inverse optimal control with partial-state information: A soft-optimality approach with constrained state estimation,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*. Las Vegas, NV, USA: IEEE, Dec. 2016, pp. 1926–1932.

[14] A. Aswani, Z.-J. M. Shen, and A. Siddiq, “Inverse Optimization with Noisy Data,” *Operations Research*, vol. 66, no. 3, pp. 870–892, June 2018.

[15] T. Toledo and D. Zohar, “Modeling Duration of Lane Changes,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1999, no. 1, pp. 71–78, Jan. 2007.

[16] Q. Wang, Z. Li, and L. Li, “Investigation of Discretionary Lane-Change Characteristics Using Next-Generation Simulation Data Sets,” *Journal of Intelligent Transportation Systems*, vol. 18, no. 3, pp. 246–253, July 2014.

[17] S. Jokhio, P. Olleja, J. Bärgman, F. Yan, and M. Baumann, “Analysis of Time-to-Lane-Change-Initiation Using Realistic Driving Data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 5, pp. 4620–4633, May 2024.

[18] J. F.-S. Lin, V. Bonnet, A. M. Panchea, N. Ramdani, G. Venture, and D. Kulic, “Human motion segmentation using cost weights recovered from inverse optimal control,” in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. Cancun, Mexico: IEEE, Nov. 2016, pp. 1107–1113.

[19] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, “NuPlan: A closed-loop ML-based planning benchmark for autonomous vehicles,” in *CVPR ADP3 Workshop*, 2021.