MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Inverse Learning for Human-Adaptive Motion Planning

Menner, Marcel; Berntorp, Karl; Di Cairano, Stefano

TR2019-146 December 11, 2019

Abstract

This paper presents a method for inverse learning of a control objective defined in terms of requirements and their probability distribution. The probability distribution characterizes tolerated deviations from the deterministic requirements, is modeled as Gaussian, and learned from data using likelihood maximization. Further, this paper introduces both parametrized requirements for motion planning in autonomous driving applications and methods for their estimation from demonstrations. Human-in-the-loop simulations with four drivers suggest that human motion planning can be modeled with the considered probabilistic control objective and the inverse learning methods in this paper enable more natural and personalized automated driving.

IEEE Conference on Decision and Control (CDC)

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Inverse Learning for Human-Adaptive Motion Planning

Marcel Menner, Karl Berntorp, Melanie N. Zeilinger, and Stefano Di Cairano

***Abstract*— This paper presents a method for inverse learning of a control objective defined in terms of requirements and their probability distribution. The probability distribution characterizes tolerated deviations from the deterministic requirements, is modeled as Gaussian, and learned from data using likelihood maximization. Further, this paper introduces both parametrized requirements for motion planning in autonomous driving applications and methods for their estimation from demonstrations. Human-in-the-loop simulations with four drivers suggest that human motion planning can be modeled with the considered probabilistic control objective and the inverse learning methods in this paper enable more natural and personalized automated driving.**

I. Introduction

Humans can perform complex tasks that are difficult to achieve with autonomous systems. One reason is that it is difficult to analytically model human knowledge or incorporate human expertise into a control objective. Autonomous driving is one such task where humans' capabilities of real-time decision-making and trading-off various objectives are hard to achieve by pure model-based approaches [1], [2]. Calibrating a control objective to achieve human-like control of a complex system such as an autonomous vehicle can be a challenging, time-consuming and expensive task. Inverse learning methods such as imitation learning [3], inverse reinforcement learning [4], or inverse optimal control [5] approach the problem of calibrating a control policy or control objective in a more systematic way: Given a desired behavior observed in demonstrations, what is the underlying objective that caused it?

In this paper, we propose a method to learn a control objective, which consists of deterministic requirements and a probability distribution that represents tolerated deviations from the requirements accounting for uncertainties and noise, or that the requirements may not be all perfectly achieved. This control objective for decision-making is proposed in [6], where a particle filter extracts the motion plan for autonomous driving. This paper considers the inverse problem, where motion plans generated by a different "actor", e.g., a human, are given from demonstrations and we estimate the deterministic requirements and the probability distribution. First, we propose a parametrized requirement function to be used for motion planning, and methods to learn its parameters. Second, we present a likelihood maximization method to estimate the probability distribution. Human-in-the-loop simulations with four drivers suggest that both the probability distribution and the parameters of the requirement function are individual, thereby allowing for tailoring the motion planner to individual preferences.

The motivation for this work is to automate the training and calibration of autonomous driving to achieve a more natural and personalized driving style for human passengers, while retaining safety and behavioral guarantees of model-based motion-planning algorithms. While a pure end-to-end learning approach results in a black-box algorithm that is difficult to assess and verify, a pure model-based approach may be easier to assess and verify but is difficult to calibrate for achieving human-like and personalized driving. Here, we pursue a gray-box learning approach and calibrate the model-based motion-planning algorithm in [6] with the proposed learning method using demonstrations of human drivers. In this way, the overall driving behavior is guaranteed to have the general properties and guarantees of the motion planner [6], including collision avoidance and specific behaviors in safety-critical decisions. Yet, among the admissible motion plans, the ones that are closer to the driver's natural behavior and that enhance passenger comfort are chosen.

Related learning approaches are presented in [7]–[14]. Imitation learning methods [7] learn control policies directly from demonstrations and inverse reinforcement learning/inverse optimal control methods [8]–[14] assume that demonstrations are caused by an underlying objective function, whose parameters are learned. Typically, inverse optimal control methods such as [8]–[12] assume a deterministic control objective and the resulting control actions are deterministic. Under this assumption, the performance may deteriorate in the presence of imperfect information such as noisy data or sub-optimal demonstrations. Some notable exceptions are [13], where a risk-metric model is introduced to circumvent risk-neutral, deterministic objective functions, and [14], where policies are constructed for scenarios with multiple future outcomes. Differently from [8]–[11], we learn a probabilistic control objective and, therefore, demonstrations are also interpreted as nondeterministic. Similar to [12]–[14], we consider nondeterministic decision-making, however, we use a different model and learn a probability distribution which adds stochasticity to the control objective. This is especially relevant in autonomous driving, due to the presence of uncertainty in the environment, modeling errors, and sensing and localization errors.

Notation: $p(\boldsymbol{x}_{0:T}|\boldsymbol{y}) := p(\boldsymbol{x}_0, \boldsymbol{x}_1, ..., \boldsymbol{x}_T|\boldsymbol{y})$ denotes the conditional probability density function (PDF) of $\boldsymbol{x}_k \in \mathbb{R}^n$ at time $k = 0, ..., T$, conditioned on $\boldsymbol{y}$. Given mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ stand for the Gaussian distribution and PDF, respectively, $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ means $\boldsymbol{x}$ sampled from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, and $\propto$ reads

M. Menner, K. Berntorp, and S. Di Cairano are with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, 02139, USA {menner, berntorp, dicairano}@merl.com

M. N. Zeilinger is with the Institute for Dynamic Systems and Control, ETH Zurich, 8092 Zurich, Switzerland {mzeilinger}@ethz.ch

proportional to. The expected value of x is $\mathbb{E}[x]$ and for a matrix $\boldsymbol{Z}$, z_{ij} is the element in the ith row and jth column.

II. Preliminaries & Problem Statement

We consider discrete-time vehicle models of the form

$$\boldsymbol{x}_k = \boldsymbol{f}\left(\boldsymbol{x}_{k-1}\right) + \boldsymbol{g}\left(\boldsymbol{x}_{k-1}\right)\boldsymbol{u}_k, \tag{1}$$

where $\boldsymbol{f}$ and $\boldsymbol{g}$ are in general nonlinear functions, $\boldsymbol{x}_k \in \mathbb{R}^{n_x}$ is the state at time k, and $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$ denotes the input applied from discrete time-step $k-1$ to k. The behavior of (1) is modeled with respect to requirements $\boldsymbol{y}_k \in \mathbb{R}^{n_y}$ with

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) + \boldsymbol{v}_k, \tag{2}$$

where we call $\boldsymbol{h}$ the requirement function and $\boldsymbol{v}_k$ is the slack, with $\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ if all requirements are obeyed perfectly. The requirement function $\boldsymbol{h}$ is deterministic with potentially unknown parameters. On the other hand, the tolerated deviations from the requirements, represented by the slack $\boldsymbol{v}_k$, are probabilistic and, therefore inflict a probability distribution upon the requirements. The goal of the proposed method is to learn the parameters of the requirement function $\boldsymbol{h}$ and the probability distribution from demonstrations. Note that we deviate from standard literature by using $\boldsymbol{u}_k$ with index k instead of $k-1$ in (1) to ease notation.

A. Motion Planner & Modeling Assumptions

Using this probability distribution and the requirement function, the considered motion planner as in [6] constructs the state trajectory PDF given the requirements, i.e., $p(\boldsymbol{x}_{0:T}|\boldsymbol{y}_{0:T})$ with $\boldsymbol{y}_k$ in (2) from time $k=0$ through $k=T$, and extracts the state trajectory from the PDF. The extracted state trajectory is then used as motion plan. In this context, the motion-planning problem is formulated as an optimal estimation problem, in which the requirements $\boldsymbol{y}_k$ in (2) are treated as sensor measurements and $\boldsymbol{u}_k$ in (1) is the input (process) disturbance. We model the input disturbance in (1) as Gaussian distributed $\boldsymbol{u}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$ and the slack in (2) as $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$.

Control Inputs: In order to find the control inputs that result from this model, we use independence of the random variables to write the joint probability recursively:

$$p(\boldsymbol{x}_{0:T}|\boldsymbol{y}_{0:T}) \propto \textstyle\prod_{k=1}^{T} p(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{x}_{k-1})$$

and, at the first order, $\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{D}_k\boldsymbol{u}_k$ with $\boldsymbol{H}_k = \partial\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}_k)/\partial\boldsymbol{x}|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k}$, $\boldsymbol{D}_k = \partial\boldsymbol{h}(\hat{\boldsymbol{x}}_k, \boldsymbol{u})/\partial\boldsymbol{u}|_{\boldsymbol{u}=\hat{\boldsymbol{u}}_k}$,

$$p(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{x}_{k-1}) \approx \mathcal{N}(\hat{\boldsymbol{x}}_k, \boldsymbol{G}_k\boldsymbol{\Sigma}_k\boldsymbol{G}_k^{\mathrm{T}}), \tag{3a}$$

where $\boldsymbol{G}_k = \boldsymbol{g}(\boldsymbol{x}_{k-1})$, and

$$\hat{\boldsymbol{x}}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{G}_k\hat{\boldsymbol{u}}_k \tag{3b}$$

$$\hat{\boldsymbol{u}}_k = \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{f}\left(\boldsymbol{x}_{k-1}\right), \boldsymbol{0})\right) \tag{3c}$$

$$\boldsymbol{K}_k = \boldsymbol{Q}\left(\boldsymbol{H}_k\boldsymbol{G}_k + \boldsymbol{D}_k\right)^{\mathrm{T}}\boldsymbol{\Gamma}_k^{-1} \tag{3d}$$

$$\boldsymbol{\Gamma}_k = (\boldsymbol{H}_k\boldsymbol{G}_k + \boldsymbol{D}_k)\boldsymbol{Q}\left(\boldsymbol{H}_k\boldsymbol{G}_k + \boldsymbol{D}_k\right)^{\mathrm{T}} + \boldsymbol{R} \tag{3e}$$

$$\boldsymbol{\Sigma}_k = \left(\boldsymbol{I} - \boldsymbol{K}_k(\boldsymbol{H}_k\boldsymbol{G}_k + \boldsymbol{D}_k)\right)\boldsymbol{Q}. \tag{3f}$$

Eq. (3) can be derived using the conditional Gaussian distribution of $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$ and is similar to a measurement update of an extended Kalman filter, where $\hat{\boldsymbol{x}}_k$ is the optimal state estimate, $\boldsymbol{K}_k$ is the optimal Kalman gain, $\boldsymbol{\Gamma}_k$ is the innovation covariance, and $\boldsymbol{\Sigma}_k$ is the estimate covariance.

The control inputs of the resulting motion planner are obtained from (3):

$$\boldsymbol{u}_k = \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{f}\left(\boldsymbol{x}_{k-1}\right), \boldsymbol{0})\right) + \boldsymbol{\sigma}_k \tag{4}$$

with the optimal Kalman gain $\boldsymbol{K}_k$ in (3d) and $\boldsymbol{\sigma}_k \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}_k\right)$ with $\boldsymbol{\Sigma}_k$ in (3f), and is therefore entirely specified by the covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ along with $\boldsymbol{y}_k$ and $\boldsymbol{h}$.

B. Outline

This paper presents a likelihood maximization method to estimate $\boldsymbol{Q}$ and $\boldsymbol{R}$ from state and input measurements (4) for general requirement functions in Section III. Further, it proposes a parametrized requirement function $\boldsymbol{h}$ for human-conscious motion planning in autonomous driving applications in Section IV and presents algorithms to learn its parameters in Section V.

III. Estimation of Covariance Matrices

We estimate the covariance matrices for a given requirement function from the posterior distribution

$$p(\boldsymbol{Q}, \boldsymbol{R}|\boldsymbol{x}_{0:T}, \boldsymbol{y}_{0:T}) \propto p(\boldsymbol{x}_{0:T}, \boldsymbol{y}_{0:T}|\boldsymbol{Q}, \boldsymbol{R})p(\boldsymbol{Q}, \boldsymbol{R}), \tag{5}$$

where $\boldsymbol{x}_{0:T}$ and $\boldsymbol{y}_{0:T}$ are the observed state and known requirement trajectories, respectively, from time $k=0$ through T, $p(\boldsymbol{Q}, \boldsymbol{R})$ is a prior probability, which is used to favor structurally beneficial $\boldsymbol{Q}, \boldsymbol{R}$, and $p(\boldsymbol{x}_{0:T}, \boldsymbol{y}_{0:T}|\boldsymbol{Q}, \boldsymbol{R})$ is the likelihood of the observations. The two probabilities are further specified in the following. In (5) and what follows, $\boldsymbol{x}_{0:T}$ refers to observations of closed-loop driving and differs from the open-loop motion planner in Section II-A by observing the inputs in (4) rather than $\boldsymbol{u}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q})$.

1) Likelihood $p(\boldsymbol{x}_{0:T}, \boldsymbol{y}_{0:T}|\boldsymbol{Q}, \boldsymbol{R})$: Consider the system dynamics (1) and the measurement (requirement) equation (2), along with the control law (4). If $\boldsymbol{h}$ in (2) is linear, i.e., $\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{D}_k\boldsymbol{u}_k$, and $p(\boldsymbol{x}_0, \boldsymbol{y}_0) = 1$, then

$$p(\boldsymbol{x}_{0:T}, \boldsymbol{y}_{0:T}|\boldsymbol{Q}, \boldsymbol{R}) = \prod_{k=1}^{T} p(\boldsymbol{u}_k|\boldsymbol{0}, \boldsymbol{Q})p(\boldsymbol{v}_k|\boldsymbol{0}, \boldsymbol{R})\frac{p(\boldsymbol{v}_k|\boldsymbol{0}, \boldsymbol{R})}{p(\boldsymbol{e}_k|\boldsymbol{0}, \boldsymbol{\Gamma}_k)} \tag{6}$$

with $\boldsymbol{e}_k = \boldsymbol{y}_k - \boldsymbol{h}(\boldsymbol{f}(\boldsymbol{x}_{k-1}), \boldsymbol{0})$, which is formally shown in Theorem 1. First, we provide two technical results.

Lemma 1. *Let $\boldsymbol{u}_k$ as in (4) and $\hat{\boldsymbol{u}}_k = \boldsymbol{K}_k\boldsymbol{e}_k$. Then,*

$$\begin{aligned} &p(\boldsymbol{x}_k|\boldsymbol{y}_k, \boldsymbol{x}_{k-1}, \boldsymbol{Q}, \boldsymbol{R})p(\boldsymbol{y}_k|\boldsymbol{x}_{k-1}, \boldsymbol{Q}, \boldsymbol{R}) \\ &= p(\boldsymbol{u}_k|\hat{\boldsymbol{u}}_k, \boldsymbol{\Sigma}_k)p(\boldsymbol{v}_k|\boldsymbol{0}, \boldsymbol{R}). \end{aligned}$$

Lemma 2. *Let $\boldsymbol{u}_k$ as in (4) and $\hat{\boldsymbol{u}}_k = \boldsymbol{K}_k\boldsymbol{e}_k$. Then,*

$$p(\boldsymbol{u}_k|\hat{\boldsymbol{u}}_k, \boldsymbol{\Sigma}_k) = p(\boldsymbol{u}_k|\boldsymbol{0}, \boldsymbol{Q})\frac{p(\boldsymbol{v}_k|\boldsymbol{0}, \boldsymbol{R})}{p(\boldsymbol{e}_k|\boldsymbol{0}, \boldsymbol{\Gamma}_k)}.$$

Proof of Lemma 1 and Lemma 2 (Sketch). The results are obtained by showing that the respective PDFs are equivalent. Equivalence of the PDFs' determinants can be shown using Sylvester's determinant identity [15]. □

Theorem 1. *Consider* (1) *and let* $p(\boldsymbol{x}_0, \boldsymbol{y}_0) = 1$. *If* $\boldsymbol{v}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$ *with* $\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{D}_k \boldsymbol{u}_k$ *in* (2) *and* $\boldsymbol{u}_k$ *as in* (4) *with* $\boldsymbol{\sigma}_k \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_k)$, *then* (6) *holds.*

Proof. First, $p(\boldsymbol{x}_k, \boldsymbol{y}_k | \boldsymbol{x}_{0:k-1}, \boldsymbol{y}_{0:k-1}) = p(\boldsymbol{x}_k, \boldsymbol{y}_k | \boldsymbol{x}_{k-1})$ and conditioning yields

$$p(\boldsymbol{x}_{0:T}, \boldsymbol{y}_{0:T}) = \prod_{k=1}^{T} p(\boldsymbol{x}_k, \boldsymbol{y}_k | \boldsymbol{x}_{k-1}) p(\boldsymbol{x}_0, \boldsymbol{y}_0).$$

Using $p(\boldsymbol{x}_k, \boldsymbol{y}_k | \boldsymbol{x}_{k-1}) = p(\boldsymbol{x}_k | \boldsymbol{y}_k, \boldsymbol{x}_{k-1}) p(\boldsymbol{y}_k | \boldsymbol{x}_{k-1})$,

$$p(\boldsymbol{x}_{0:T}, \boldsymbol{y}_{0:T} | \boldsymbol{Q}, \boldsymbol{R}) = \prod_{k=1}^{T} p(\boldsymbol{x}_k | \boldsymbol{y}_k, \boldsymbol{x}_{k-1}, \boldsymbol{Q}, \boldsymbol{R}) p(\boldsymbol{y}_k | \boldsymbol{x}_{k-1}, \boldsymbol{Q}, \boldsymbol{R}). \quad (7)$$

Reformulating (7) using Lemma 1 and 2 shows (6). □

Corollary 1. *For a purely stochastic control law* $\boldsymbol{u}_k = \boldsymbol{\sigma}_k$, *the sample covariances* $\boldsymbol{Q} = 1/T \sum_{k=1}^{T} \boldsymbol{u}_k \boldsymbol{u}_k^{\mathrm{T}}$ *and* $\boldsymbol{R} = 1/T \sum_{k=1}^{T} \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{T}}$ *are the maximum likelihood solutions to* $\max_{\boldsymbol{Q},\boldsymbol{R}} p(\boldsymbol{x}_{0:T}, \boldsymbol{y}_{0:T} | \boldsymbol{Q}, \boldsymbol{R})$.

Proof. This follows directly from Lemma 1 with $\hat{\boldsymbol{u}}_k = \boldsymbol{0}$ and $\boldsymbol{\Sigma}_k = \boldsymbol{Q}$, for which the sample covariance is the maximum likelihood estimator [16]. □

2) Prior Belief $p(\boldsymbol{Q}, \boldsymbol{R})$: We model the prior probability as a Gaussian distribution with PDF $p(\boldsymbol{t}(\boldsymbol{Q}, \boldsymbol{R}) | \boldsymbol{0}, \rho \boldsymbol{I}_{n_t})$, where $\boldsymbol{t}(\boldsymbol{Q}, \boldsymbol{R}) \in \mathbb{R}^{n_t}$ imposes the prior belief on $\boldsymbol{Q}, \boldsymbol{R}$ and ρ trades off prior belief and maximum likelihood estimation, i.e., for $\rho \to 0$, the maximum a posteriori estimate is the maximum of the prior and for $\rho \to \infty$, the maximum a posteriori converges to the maximum likelihood estimate.

3) Maximum a Posteriori & Overall Algorithm: Overall, we want to find $\boldsymbol{Q}$, $\boldsymbol{R}$ that maximize the likelihood:

$$\max_{\boldsymbol{Q},\boldsymbol{R}} \quad \log p(\boldsymbol{x}_{0:T}, \boldsymbol{y}_{0:T} | \boldsymbol{Q}, \boldsymbol{R}) p(\boldsymbol{Q}, \boldsymbol{R}) \quad (8a)$$

$$\text{subject to} \quad \boldsymbol{Q} \in \mathcal{C}_Q, \boldsymbol{R} \in \mathcal{C}_R, \quad (8b)$$

where $\mathcal{C}_Q, \mathcal{C}_R$ can be used to enforce constraints on $\boldsymbol{Q}, \boldsymbol{R}$, e.g., $\boldsymbol{Q} = \boldsymbol{Q}^{\mathrm{T}} \succeq \boldsymbol{0}, \boldsymbol{R} = \boldsymbol{R}^{\mathrm{T}} \succeq \boldsymbol{0}$. We use the first-order approximation $\boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) \approx \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{D}_k \boldsymbol{u}_k$, for which $p(\boldsymbol{x}_{0:T}, \boldsymbol{y}_{0:T} | \boldsymbol{Q}, \boldsymbol{R})$ is approximated as in (6). We optimize (8) with a projected gradient method in Algorithm 1 and define $\boldsymbol{J}_k = \boldsymbol{H}_k \boldsymbol{G}_k + \boldsymbol{D}_k$. We use Corollary 1 to initialize $\boldsymbol{Q}$ and $\boldsymbol{R}$. The projection `project` enforces $\boldsymbol{Q} \in \mathcal{C}_Q, \boldsymbol{R} \in \mathcal{C}_R$ and is specified in Section VII for the considered application. The step-size l does not affect the optimal solution but the convergence rate, and $\| \cdot \|_F$ denotes the Frobenius norm.

Remark 1. *The complexity of the gradient computation of* (8) *with* (6) *using tensor calculus [17] is* $\mathcal{O}(T(n_y^4(n_y^2+n_u^2)))$, *which is feasible for the problem dimension we consider.*

Algorithm 1 Estimation of $\boldsymbol{Q}$, $\boldsymbol{R}$

1: $i = 0$, $\boldsymbol{Q}^0 = 1/T \sum_{k=1}^{T} \boldsymbol{u}_k \boldsymbol{u}_k^{\mathrm{T}}$, $\boldsymbol{R}^0 = 1/T \sum_{k=1}^{T} \boldsymbol{v}_k \boldsymbol{v}_k^{\mathrm{T}}$
2: **do** ▷ $\varepsilon = 10^{-6}$ in our case
3: $\mathrm{d}\boldsymbol{Q}, \mathrm{d}\boldsymbol{R} \leftarrow$ `getGrad` $(\boldsymbol{Q}^i, \boldsymbol{R}^i, \boldsymbol{Q}^0, \boldsymbol{R}^0, \{\boldsymbol{e}_k, \boldsymbol{J}_k\}_{k=0}^{T})$
4: $\boldsymbol{Q} = \boldsymbol{Q} + l \cdot \mathrm{d}\boldsymbol{Q}$; $\boldsymbol{R} = \boldsymbol{R} + l \cdot \mathrm{d}\boldsymbol{R}$
5: $\boldsymbol{Q}^{i+1}, \boldsymbol{R}^{i+1} \leftarrow$ `project`$(\boldsymbol{Q}, \boldsymbol{R})$
6: $i \leftarrow i + 1$
7: **while** $\| \boldsymbol{Q}^i - \boldsymbol{Q}^{i-1} \|_F + \| \boldsymbol{R}^i - \boldsymbol{R}^{i-1} \|_F \leq \varepsilon$

IV. Requirements for Autonomous Driving

The requirements for the motion-planning application in this paper are: i) Stay in the middle of a target lane, ii) maintain a target velocity, iii) drive in a natural way for the passenger, and iv) keep a safety distance to surrounding obstacles. The requirements $\boldsymbol{y}_k$ in (2) are formalized as

$$\boldsymbol{y}_k = \begin{bmatrix} 0 \\ v_{\mathrm{nom}} \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{u}_k) = \begin{bmatrix} h^l(p_{X,k}, p_{Y,k}) \\ v_{x,k} \\ h^c(a_{x,k}, a_{y,k}) \\ h^o(d_k, v_{x,k}) \end{bmatrix}$$

where $h^l(p_X, p_Y)$ denotes the squared distance from the centerline at vehicle position p_X, p_Y, v_{nom} and v_x are the nominal and current velocity, respectively, $h^c(a_x, a_y)$ is the passenger comfort requirement with longitudinal acceleration a_x and lateral acceleration a_y, and $h^o(d, v_x)$ is the obstacle avoidance requirement with separation distance d between ego vehicle (EV) and obstacle vehicles (OVs). Both the passenger comfort and obstacle avoidance requirements are expected to vary between drivers and are introduced next.

A. Passenger Comfort Requirement

We model the passenger comfort requirement as penalty for longitudinal and lateral accelerations, as well as their coupling. The accelerations and their coupling are well known to relate to the individual driving style [18]. The requirement is formalized as

$$h^c(a_x, a_y) = \bar{a} \cdot \frac{c(a_x, a_y) - c^0}{c^1 - c^0} \quad (9)$$

with $c^1 = (\sqrt{(\bar{a}^2 + \epsilon)}^{n_c} + \sqrt{\epsilon}^{n_c})^{\frac{1}{n_c}}$, $c^0 = (2\sqrt{\epsilon}^{n_c})^{\frac{1}{n_c}}$, and

$$c(a_x, a_y) = (\sqrt{(a_x^2 + \epsilon)}^{n_c} + \sqrt{((s \cdot a_y)^2 + \epsilon)}^{n_c})^{\frac{1}{n_c}}$$

and a small $\epsilon > 0$. The parameter $\bar{a}$ is a measure for maximum accelerations, s defines a unilateral scaling, i.e., for $s \neq 1$, $h^c(a_x, a_y) \neq h^c(a_y, a_x)$, and c^1, c^0 ensure $h^c(0, 0) = 0$ and $h^c(\bar{a}, 0) = h^c(0, \bar{a}/s) = \bar{a}$. The exponent n_c shapes the level sets of (9) so that, for higher n_c, the level sets are more circular, cf. the left plot in Figure 1.

Note that (9) differs from a (pseudo)norm for $[a_x \ a_y]^{\mathrm{T}}$ by virtue of $\epsilon \neq 0$ and $s \neq 1$. However, we introduce $\epsilon > 0$ (and c^1, c^0 as a consequence) for two reasons: First, $h^c(a_x, a_y)$ is differentiable with respect to a_x, a_y for all n_c. Second, the level sets have different shape so that the penalty in combined longitudinal and lateral accelerations is reduced for smaller values relative to their magnitude (more circular shape around the origin), cf. the right plot in Figure 1.

B. Obstacle Avoidance Requirement

The obstacle avoidance requirement is modeled as a piecewise linear function

$$h^o(d, v_x) = \begin{cases} d_{\min} + t_s v_x - d & \text{if } d_{\min} + t_s v_x \geq d \\ 0 & \text{else,} \end{cases} \quad (10)$$

where $d_{\min}$ is the minimum distance to be kept from the OVs and $t_s v_x$ is the traveled distance of the EV within

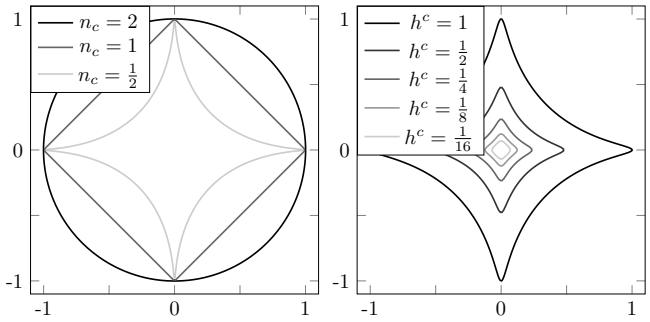

Fig. 1. Left: Level sets $h^c = 1$ with varying n_c, $s = 1$, and $\epsilon = 0$. Right: Varying level sets h^c with $n_c = \frac{1}{2}$ and $\epsilon = 0.01$.

the safety time t_s at v_x and takes into account that the safety distance from the OVs is velocity-dependent. Note that lateral obstacle constraints are considered through the motion planner as discussed in [6].

V. Estimation of Requirement Parameters

In this section, we present estimation methods to learn both the proposed passenger comfort requirement, i.e. n_c, $\bar{a}$, and s in (9), and the obstacle avoidance requirement, i.e. t_s and $d_{\min}$ in (10), from demonstrations of human driving.

A. Estimation of Passenger Comfort Requirement

Let $\mathcal{A}_x = \{a_{x,0}, ..., a_{x,T}\}$, $\mathcal{A}_y = \{a_{y,0}, ..., a_{y,T}\}$, and

$$\mathcal{A} = \left\{ \begin{bmatrix} a_{x,0} \\ a_{y,0} \end{bmatrix}, ..., \begin{bmatrix} a_{x,T} \\ a_{y,T} \end{bmatrix} \right\}$$
$$\mathcal{A}_{\mathrm{c}} = \{a_{x,0}, ..., a_{x,T}, a_{y,0}, ..., a_{y,T}\}.$$

Estimation of s: We compute s as the median of the M largest longitudinal accelerations divided by the median of the M largest lateral accelerations.

Estimation of $\bar{a}$: We compute $\bar{a}$ as the median of the M largest elements in absolute value of the set defined by $\mathcal{A}_x \cup (s \cdot \mathcal{A}_y)$, where $s \cdot \mathcal{A}_y = \{s \cdot a_{y_0}, ..., s \cdot a_{y_T}\}$.

Estimation of n_c: The exponent n_c is obtained by estimating the shape of level set $\bar{a}$ using the passenger comfort requirement as (pseudo)norm. First, we compute a set with strong coupling of longitudinal and lateral accelerations using a small $n_{c,0}$, denoted $\mathcal{F}_0$. Then, n_c is increased iteratively until the median comfort level in $\mathcal{F}_0$ is greater than or equal to $\bar{a}$.

B. Estimation of Obstacle Avoidance Requirement

We use a system identification-like approach similar to [19] to estimate the parameters $d_{\min}$ and t_s. The intuitive idea is that the observed data originates from either of two models: Driving with or without traffic. Considering the piecewise linear h^o in (10), we want the switch between the two models to coincide with $d_{\min} = d - t_s v_x$, where $d_{\min} < d - t_s v_x$ indicates traffic-free and $d_{\min} > d - t_s v_x$ is traffic-affected driving.

Estimation of $d_{\min}$: We estimate $d_{\min}$ as the median of the M smallest observed distances with $\mathcal{D} = \{d_0, ..., d_T\}$.

Estimation of t_s: Let

$$\begin{aligned}
\boldsymbol{h}^r(\boldsymbol{x}, \boldsymbol{u}) &= \begin{bmatrix} h^l(p_X, p_Y) & v_x & h^c(a_x, a_y) \end{bmatrix}^{\mathrm{T}} \\
\boldsymbol{H}_k^r &= \left. \frac{\partial \boldsymbol{h}^r(\boldsymbol{x}, \hat{\boldsymbol{u}}_k)}{\partial \boldsymbol{x}} \right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}_k}, \ \boldsymbol{D}_k^r = \left. \frac{\partial \boldsymbol{h}^r(\hat{\boldsymbol{x}}_k, \boldsymbol{u})}{\partial \boldsymbol{u}} \right|_{\boldsymbol{u}=\hat{\boldsymbol{u}}_k} \quad (11) \\
\boldsymbol{v}_k^r &= \begin{bmatrix} 0 & v_{\mathrm{nom}} & 0 \end{bmatrix}^{\mathrm{T}} - \boldsymbol{h}^r(\boldsymbol{x}_k, \boldsymbol{u}_k) \\
\boldsymbol{e}_k^r &= \begin{bmatrix} 0 & v_{\mathrm{nom}} & 0 \end{bmatrix}^{\mathrm{T}} - \boldsymbol{h}^r(\boldsymbol{f}(\boldsymbol{x}_{k-1}), \boldsymbol{0}),
\end{aligned}$$

where r denotes reduced (by the obstacle avoidance requirement). If the parameters of h^c are known (Section V-A), we can use Algorithm 1 to estimate $\boldsymbol{Q}^r$ and $\boldsymbol{R}^r$ using (11) for traffic-free driving. Further, in (4), let $\boldsymbol{K}_k = \boldsymbol{K}_k^{\mathrm{TF}} + \boldsymbol{K}_k^{\mathrm{TA}}$, where $\boldsymbol{K}_k^{\mathrm{TF}} = \boldsymbol{K}_k^r \boldsymbol{T}$ is the gain matrix in the absence of traffic (traffic-free) with $\boldsymbol{T} = [\ \boldsymbol{I}_3\ \boldsymbol{0}_{3\times 1}]$,

$$\boldsymbol{K}_k^r = \boldsymbol{Q}^r \boldsymbol{J}_k^{r\,\mathrm{T}} (\boldsymbol{J}_k^r \boldsymbol{Q}^r \boldsymbol{J}_k^{r\,\mathrm{T}} + \boldsymbol{R}^r)^{-1}, \ \boldsymbol{J}_k^r = \boldsymbol{H}_k^r \boldsymbol{G}_k + \boldsymbol{D}_k^r$$

and $\boldsymbol{K}_k^{\mathrm{TA}}$ is the traffic-affected gain. This decomposition of $\boldsymbol{K}_k$ yields $\boldsymbol{\sigma}_k = \boldsymbol{u}_k - \boldsymbol{K}_k^{\mathrm{TF}} \boldsymbol{e}_k \sim \mathcal{N}(\boldsymbol{K}_k^{\mathrm{TA}} \boldsymbol{e}_k, \boldsymbol{\Sigma}_k)$. Hence, in the absence of traffic ($\boldsymbol{K}_k^{\mathrm{TA}} \boldsymbol{e}_k = \boldsymbol{0}$), $\boldsymbol{\sigma}_k = \boldsymbol{u}_k - \boldsymbol{K}_k^{\mathrm{TF}} \boldsymbol{e}_k$ is sampled from a distribution with zero mean and, in the presence of traffic ($\boldsymbol{K}_k^{\mathrm{TA}} \boldsymbol{e}_k \neq \boldsymbol{0}$), $\boldsymbol{u}_k - \boldsymbol{K}_k^{\mathrm{TF}} \boldsymbol{e}_k$ is sampled from a distribution with mean $\boldsymbol{K}_k^{\mathrm{TA}} \boldsymbol{e}_k$.

We use this change in mean for the estimation of t_s:

$$t_s = \arg \min_{t_s, \mathcal{T}, \boldsymbol{a}_i} \sum_{k \in \mathcal{T}} I_k + \sum_{k \notin \mathcal{T}} J_k \quad (12a)$$

with $I_k = \|\boldsymbol{u}_k - \boldsymbol{K}_k^r \boldsymbol{e}_k^r\|_2^2$, $J_k = \|\boldsymbol{u}_k - \boldsymbol{K}_k^r \boldsymbol{e}_k^r - \boldsymbol{p}(d_k, v_{x,k})\|_2^2$ and

$$\mathcal{T} = \{i \mid d_{\min} + t_s v_{x,i} \leq d_i\}, \quad (12b)$$

where $\boldsymbol{p}(d, v_x) = \sum_{i=0}^{1} \boldsymbol{a}_i \left(d_{\min} + t_s v_x - d\right)^i$ with coefficients $\boldsymbol{a}_i \in \mathbb{R}^{n_u}$ is used to approximate the nonzero mean. Eq. (12) is a combinatorial problem, however, for a fixed t_s, it reduces to a convex least squares problem in $\boldsymbol{a}_i$. We solve (12) by enumerating t_s in Δt_{inc} increments with $t_s \in [0\mathrm{s}\ t_{s,\max}]$ and choose $t_{s,\max} = 10\mathrm{s}$, $\Delta t_{\mathrm{inc}} = 0.01\mathrm{s}$.

VI. Human-in-the-Loop Simulation Setup

We carried out human-in-the-loop simulations by controlling a vehicle in CarSim using the torque feedback Thrustmaster T300RS gaming steering wheel through a MATLAB interface. The track is a two-lanes oval circuit with two straights of 200m connected by two 180° turns with radius 53.6m at the centerline of the right lane, Both ego vehicle and obstacle vehicles drive anti-clockwise. Each test driver completed the following:

Task 0: The driver familiarized themself with the driving simulator and was prepared for Task 1 and Task 2. No data were recorded during this task.

Task 1 (15min recorded): The driver was instructed to stay in the right lane and that the target velocity is 50km/h. This task did not involve OVs.

Task 2 (15min recorded): The driver was allowed to use both lanes and the target velocity was 50km/h. We added 7 OVs, as specified in Table I, where the initial position is the distance along the track and the start, i.e., 0m, is the beginning of a straight segment and the EV's initial position.

TABLE I
OVs' INITIAL POSITIONS & VELOCITIES

Vehicle ID	1	2	3	4	5	6	7
Initial Position	100m	300m	500m	600m	350m	550m	600m
Lane	right	right	right	right	left	left	left
Velocity	$5.5\frac{\text{m}}{\text{s}}$	$5.5\frac{\text{m}}{\text{s}}$	$5.5\frac{\text{m}}{\text{s}}$	$5.5\frac{\text{m}}{\text{s}}$	$4.5\frac{\text{m}}{\text{s}}$	$4.5\frac{\text{m}}{\text{s}}$	$4.5\frac{\text{m}}{\text{s}}$

VII. ESTIMATED PARAMETERS & PERSONALIZED MOTION PLANNING

This section presents results of the proposed methods with four human drivers. While three drivers are normal drivers, one (Driver 3) had professional test driving training, and aims at exercising the full performance envelope of the vehicle, i.e., performance driving style. We present the estimated parameters in Section VII-B and the resulting motion planners using such parameters in Section VII-C.

A. Design Choices

We consider the kinematic single track vehicle model [6]

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{p}_X \\ \dot{p}_Y \\ \dot{\psi} \\ \dot{v}_x \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} v_x \cos(\psi+\beta)/\cos(\beta) \\ v_x \sin(\psi+\beta)/\cos(\beta) \\ v_x \tan(\delta)/L \\ u_1 \\ u_2 \end{bmatrix}$$

represented in discrete time with sampling period $T_s = 0.5$s, where p_X, p_Y mark the position of the EV in the world frame, ψ is the heading (yaw) angle, v_x is the longitudinal velocity, δ is the steering angle of the front wheel, L is the wheel base, and $\beta = \arctan(l_r \tan(\delta)/L)$ is the kinematic body-slip angle. Accelerations are computed as $a_x = \dot{v}_x$, $a_y = v_x\dot{\psi}$. The inputs u_1 and u_2 are the longitudinal acceleration and steering rate, respectively.

Constraints $\mathcal{C}_Q, \mathcal{C}_R$: We require $\boldsymbol{Q}$ to be diagonal because, if $\boldsymbol{Q}$ had nonzero off-diagonal elements, the longitudinal acceleration and steering rate would be more likely to be coupled. For instance, if $q_{12} > 0$ ($\mathbb{E}[u_1 u_2] > 0$), accelerating ($\dot{v}_{x,k} = u_1 > 0$) would imply a preference on steering to the right ($\dot{\delta} = u_2 > 0$), which appears unnatural. Further, we impose $\boldsymbol{Q} = \boldsymbol{Q}^{\mathrm{T}} \succeq \varepsilon \cdot \boldsymbol{I}$ and $\boldsymbol{R} = \boldsymbol{R}^{\mathrm{T}} \succeq \varepsilon \cdot \boldsymbol{I}$ with $\varepsilon = 10^{-3}$ and model the centerline tracking as independent of the other requirements, i.e. $r_{12} = r_{13} = r_{14} = 0$, in order to avoid oscillations, e.g., of the velocity with the centerline tracking error ($\mathbb{E}[(v_{\text{nom}} - v_x)(0 - h^l)] = 0$).

Prior: The signal-to-noise ratio $\|\boldsymbol{J}\boldsymbol{Q}\boldsymbol{J}^{\mathrm{T}}\|/\|\boldsymbol{R}\|$ with $\boldsymbol{J} = \boldsymbol{H}\boldsymbol{G} + \boldsymbol{D}$ is to be close to one, which is beneficial for particle-filter algorithms [20]. We choose $\boldsymbol{G} = \boldsymbol{g}(\boldsymbol{x}^\star)$, $\boldsymbol{H} = \partial \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{0})/\partial \boldsymbol{x}|_{\boldsymbol{x}=\boldsymbol{x}^\star}$, $\boldsymbol{D} = \partial \boldsymbol{h}(\boldsymbol{x}^\star, \boldsymbol{u})/\partial \boldsymbol{u}|_{\boldsymbol{u}=\boldsymbol{0}}$ where $\boldsymbol{x}^\star$ denotes a nominal state where all requirements are fulfilled ($v_x = 50$km/h, $\delta = 0$) and design the prior as $\boldsymbol{t}(\boldsymbol{Q}, \boldsymbol{R}) = \text{vec}(\boldsymbol{J}\boldsymbol{Q}\boldsymbol{J}^{\mathrm{T}} - \boldsymbol{R})$ with $\rho = 1$, see Section III-2, with the vectorization operator $\text{vec} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n^2}$.

B. Estimation Results

Requirement Function: Figure 2 shows scatter plots of accelerations demonstrated by the four drivers along with the estimated level set $\bar{a}$ for the passenger comfort requirement. It shows that Driver 3's driving style yields high lateral accelerations with $\bar{a}/s = 7.28\text{m/s}^2$ relative to Driver 1, 2, and 4 with $\bar{a}/s = 2.70\text{m/s}^2$, 1.49m/s^2, and 2.84m/s^2, respectively. It indicates that Driver 1 and Driver 4 exhibit similar accelerations, whereas Driver 2 avoids larger accelerations. Table II specifies the estimated parameters s, $\bar{a}$, and n_c of the passenger comfort requirement as well as $d_{\min}$ and t_s of the obstacle avoidance requirement. Here, too, Driver 2 is the most conservative keeping a minimum distance to OVs of 13.6m, compared to $d_{\min} < 9$m for the other drivers, and reacting to OVs at $t_s = 7.00$s. Driver 3 is the least conservative with $d_{\min} = 5.89$m and $t_s = 3.84$s.

Covariance Matrices: Table II specifies the estimated covariance matrices for the four drivers. The matrices can be interpreted as follows: Low values represent a low tolerance of violating the respective requirement, e.g. $r_{22} = 0.995$ of Driver 3 versus $r_{22} > 10$ for all other drivers indicates that Driver 3 is more reluctant to deviate from the target velocity. Low off-diagonal values relative to their diagonal counterparts represent little coupling of the respective two requirements, e.g. $r_{23} = -0.260$ of Driver 3 indicates that Driver 3 is not as likely to reduce their velocity for the sake of reducing lateral accelerations on the vehicle. An important off-diagonal element is r_{24}, which represents the covariance of the velocity and obstacle avoidance requirement. For $r_{24} < 0$ ($\mathbb{E}[(v_{\text{nom}} - v_x)(0 - h^o)] < 0$), the driver is more likely to reduce the velocity when encountering traffic on the target lane. Similarly, for $r_{34} > 0$ ($\mathbb{E}[(0-h^c)(0-h^o)] > 0$), the driver is more likely to sacrifice comfort in traffic. The difference in scale of the obstacle avoidance requirement is caused by the driving style and the individual $d_{\min}$, t_s.

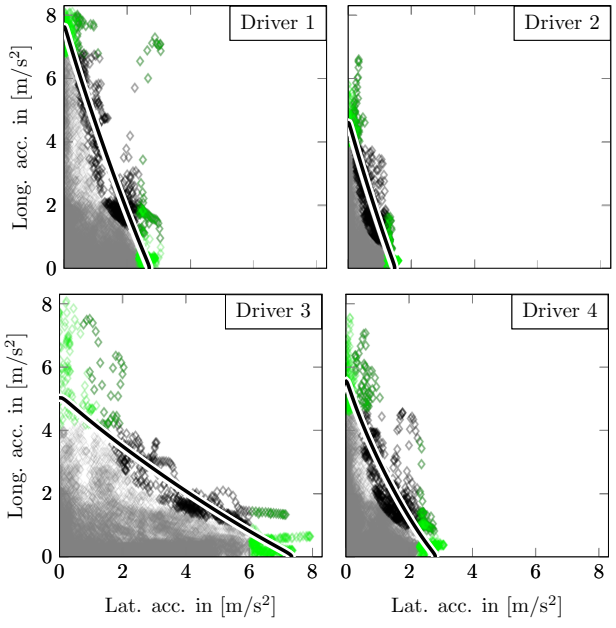

Fig. 2. Scatter plots of accelerations (absolute values) for estimating the passenger comfort requirement. All demonstrated accelerations of each driver are displayed in gray. The level set $h^c(a_x, a_y) = \bar{a}$ is displayed as black line, where s is estimated using the green data points (median of largest longitudinal divided by the largest lateral accelerations) and n_c is such that the median of the black data points is $\bar{a}$.

TABLE II

ESTIMATED PARAMETERS

Driver 1	
Passenger Comfort Requirement	$s = 2.86$, $\bar{a} = 7.73$, $n_c = 0.961$
Obstacle Avoidance Requirement	$d_{\min} = 7.32$m, $t_s = 4.10$s
Covariance Matrices	$Q = \begin{bmatrix} 10.8 & 0 \\ & 0.0017 \end{bmatrix}$, $R = \begin{bmatrix} 0.0848 & 0 & 0 & 0 \\ & 10.9 & -9.57 & -30.6 \\ & & 12.1 & 34.1 \\ & & & 182 \end{bmatrix}$
Driver 2	
Passenger Comfort Requirement	$s = 3.14$, $\bar{a} = 4.68$, $n_c = 0.951$
Obstacle Avoidance Requirement	$d_{\min} = 13.6$m, $t_s = 7.00$s
Covariance Matrices	$Q = \begin{bmatrix} 25.7 & 0 \\ & 0.0009 \end{bmatrix}$, $R = \begin{bmatrix} 0.0551 & 0 & 0 & 0 \\ & 25.7 & -10.2 & -150 \\ & & 4.97 & 43.1 \\ & & & 1260 \end{bmatrix}$
Driver 3	
Passenger Comfort Requirement	$s = 0.629$, $\bar{a} = 4.58$, $n_c = 0.941$
Obstacle Avoidance Requirement	$d_{\min} = 5.89$m, $t_s = 3.84$s
Covariance Matrices	$Q = \begin{bmatrix} 0.226 & 0 \\ & 0.0032 \end{bmatrix}$, $R = \begin{bmatrix} 0.257 & 0 & 0 & 0 \\ & 0.995 & -0.260 & -0.425 \\ & & 2.90 & 0.216 \\ & & & 4.62 \end{bmatrix}$
Driver 4	
Passenger Comfort Requirement	$s = 1.99$, $\bar{a} = 5.66$, $n_c = 0.861$
Obstacle Avoidance Requirement	$d_{\min} = 8.52$m, $t_s = 5.87$s
Covariance Matrices	$Q = \begin{bmatrix} 17.2 & 0 \\ & 0.0013 \end{bmatrix}$, $R = \begin{bmatrix} 0.0660 & 0 & 0 & 0 \\ & 17.2 & -6.77 & -89.8 \\ & & 4.46 & 23.3 \\ & & & 592 \end{bmatrix}$

C. Autonomous Motion Planning using the trained Models

We use the particle-filter algorithm in [6] with the proposed requirement function and estimated covariance matrices for validation, where we refer to the motion planner trained with the data obtained from Driver x as Planner x.

Figure 3 displays the trajectories of driving without OVs (Task 1) as mean and standard deviation over all laps of Driver 1–4 and Planner 1–4. It displays the velocity, the distance to the centerline ΔCL, and the lateral acceleration over the track position from 0 to 50% of the track. It can be seen that the velocity and lateral acceleration of the drivers and their respective planners match relatively closely with some notable exceptions. Driver 1, 3, and 4 exceeded the target velocity (indicated as constant, black line) of 13.89m/s on average on the straight segment. The path planner avoids almost always exceeding the target velocity by design as it is easier to fulfill the other requirements using lower velocity, which is desirable for autonomous driving applications. Furthermore, the planners track the centerline more closely than the drivers, which is due to the path planner optimization of the requirements.

Next, we consider a motion-planning scenario where both lanes are initially blocked by two slow OVs with velocities 30km/h and 25km/h on the right and left lane, respectively. Approaching the blockage, the planner must slow down and wait for the opportunity to overtake. Figure 4 illustrates the resulting trajectories of Planner 1–4. It displays the EV's velocity, the minimum distance to the other vehicles in the target lane, and a snapshot of the path, where the positions of the two OVs are frozen at the time of lane-change decision.

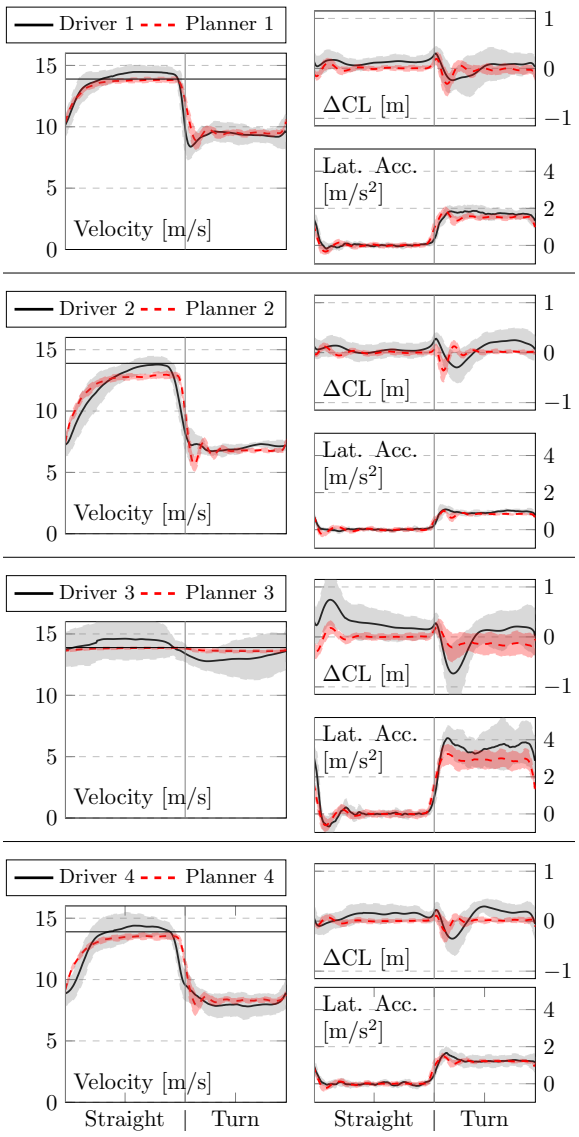

Fig. 3. Velocity, centerline tracking error, and lateral acceleration for the four drivers. The mean of all laps of the driving data is displayed in black with standard deviation in gray. The mean along with standard deviation of the path planner is displayed in dashed red.

It shows that Planner 2 is the most conservative starting to decelerate early ($v_x < 10$m/s) and keeping the largest distance from the OVs ($d > 40$m and $d > 15$m on the right and left lane, respectively). Also, Planner 2's lane-change trajectory shows the smallest curvature, which is expected from its lower tolerance for lateral accelerations, and is consistent with Driver 2 being the most cautious of the test subjects. Planner 3 is the least conservative, decelerating later than the others (see distance and velocity for $t < -30$s) and its trajectory exhibits the highest curvature, which is consistent with the performance driving style of Driver 3.

Individuality of Planners: The motion planners exhibit some similarities, e.g., not exceeding the target velocity and a small centerline tracking error. In the traffic-free driving scenario in Figure 3, the individuality of the planners can be best identified in the velocity profile and its resulting lateral acceleration. Planner 1, 2, and 4 take the turn at 9.5m/s, 7m/s,

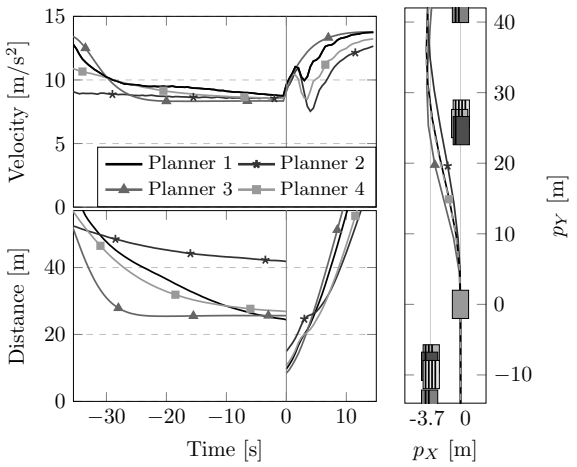

Fig. 4. Path planning in traffic. Top left: EV velocity. Bottom left: Distance from OVs in target lane. Right: Lane-change trajectory with OVs' positions frozen at time of decision and marked with x stripes for Planner x.

and 8m/s with lateral accelerations of 1.8m/s^2, 1.0m/s^2, and 1.5m/s^2, which matches their respective drivers' velocities. Planner 3 takes the turn at a slightly higher velocity than Driver 3, however, due to the optimization in the planning algorithm, the velocity is more constant during the turn and hence Planner 3 exhibits lower lateral accelerations than Driver 3, thereby fulfilling both requirements more closely.

Also, the considered traffic-affected scenario in Figure 4 shows individual components. Planner 3 accelerates quickly and monotonically after the opportunity to overtake presented itself. Planner 1, 2, and 4 exhibit a brief drop in velocity for time $>$ 0s. This drop appears at the peak curvature of the path when the EV turns right to align with the left lane and is caused by the tolerance for lateral accelerations.

Generalization to other City-Driving Scenarios: Due to combining data-based, i.e., learning, and model-based, i.e., particle-filter motion planning, approaches, the planners generalize well to road scenarios different from the training track. Table III shows the case of a circular track with varying radius, by reporting the mean of the velocities of Planner 1-4. All the planners decrease their velocities with decreasing turn radius, with Planner 2 and 4, which generally avoid higher lateral accelerations, slowing down the most.

TABLE III
CIRCULAR TRACK - AVERAGED VELOCITIES

Radius	500m	100m	50m	25m
Planner 1	13.1m/s	10.9m/s	9.31m/s	7.58m/s
Planner 2	11.4m/s	8.22m/s	6.64m/s	5.23m/s
Planner 3	13.8m/s	13.7m/s	13.5m/s	12.8m/s
Planner 4	12.7m/s	9.90m/s	8.10m/s	6.33m/s

Expected Limitation of Estimated Parameters: For significantly different driving scenarios, both the drivers and the planners are expected to behave differently. For instance in high-speed freeway driving, lane-change maneuvers may be slower (higher r_{11}) and/or velocities more constant (smaller r_{22}). This might prompt mode-dependent parameter sets for each planner, which will be addressed in future work.

VIII. CONCLUSION

This paper presented an inverse learning approach to calibrate a motion planner from demonstrations of human driving. It proposed a deterministic requirement function with a priori unknown parameters and an algorithm for their estimation. Further, it presented a likelihood maximization method to estimate the probability distribution defining tolerated deviations from the requirements. Human-in-the-loop simulations with four drivers showed that the estimates are different for each individual, thus resulting in the motion planner generating different motions that, while satisfying the intrinsic properties of the model-based planning algorithm, had a behavior similar to the corresponding drivers.

REFERENCES

[1] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33–55, 2016.

[2] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *J. Guid., Control, and Dyn.*, vol. 25, no. 1, pp. 116–129, 2002.

[3] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cogn. Sci.*, vol. 3, no. 6, pp. 233–242, 1999.

[4] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 1.

[5] P. Englert, N. A. Vien, and M. Toussaint, "Inverse KKT: Learning cost functions of manipulation tasks from demonstrations," *Int. J. Robot. Res.*, vol. 36, no. 13–14, pp. 1474–1488, 2017.

[6] K. Berntorp, T. Hoang, and S. Di Cairano, "Motion planning of autonomous road vehicles by particle filtering," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 2, pp. 197–210, 2019.

[7] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philos. Trans. Roy. Soc. London. Ser. B: Biol. Sci.*, vol. 358, no. 1431, pp. 537–547, 2003.

[8] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 475–482.

[9] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotionan inverse optimal control approach," *Auton. Robots*, vol. 28, no. 3, pp. 369–383, 2010.

[10] M. Menner and M. N. Zeilinger, "Convex formulations and algebraic solutions for linear quadratic inverse optimal control problems," in *Proc. Eur. Control Conf.*, 2018, pp. 2107–2112.

[11] M. Menner, P. Worsnop, and M. N. Zeilinger, "Constrained inverse optimal control with application to a human manipulation task," *arXiv:1812.11600*, 2018.

[12] P. M. Esfahani, S. Shafieezadeh-Abadeh, G. A. Hanasusanto, and D. Kuhn, "Data-driven inverse optimization with imperfect information," *Math. Program.*, vol. 167, no. 1, pp. 191–234, 2018.

[13] A. Majumdar, S. Singh, A. Mandlekar, and M. Pavone, "Risk-sensitive inverse reinforcement learning via coherent risk models," in *Proc. Robot.: Sci. and Syst.*, 2017.

[14] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," in *Proc. IEEE Int. Conf. Robot. and Autom.*, 2018, pp. 1–9.

[15] A. G. Akritas, E. K. Akritas, and G. I. Malaschonok, "Various proofs of Sylvester's (determinant) identity," *Math. and Comput. in Simul.*, vol. 42, no. 4–6, pp. 585–593, 1996.

[16] P. S. Dwyer, "Some applications of matrix derivatives in multivariate analysis," *J. Amer. Statist. Assoc.*, vol. 62, no. 318, pp. 607–625, 1967.

[17] K. B. Petersen and M. S. Pedersen, "The matrix cookbook (version: Nov. 15, 2012)," 2012.

[18] R. S. Rice, "Measuring car-driver interaction with the gg diagram," SAE Technical Paper, Tech. Rep., 1973.

[19] M. Lavielle, "Using penalized contrasts for the change-point problem," *Signal Process.*, vol. 85, no. 8, pp. 1501–1510, 2005.

[20] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 7, pp. 53–82, 2010.