



Curve Trajectory Model for Human Preferred Path Planning of Automated Vehicles

Gergo Ferenc Ignecci¹ · Erno Horvath¹ · Roland Toth² · Krisztian Nyilas³

Received: 8 November 2022 / Accepted: 31 August 2023 / Published online: 9 January 2024
© The Author(s) 2024

Abstract

Automated driving systems are often used for lane keeping tasks. By these systems, a local path is planned ahead of the vehicle. However, these paths are often found unnatural by human drivers. In response to this, this paper proposes a linear driver model, which can calculate node points reflective of human driver preferences and based on these node points a human driver preferred motion path can be designed for autonomous driving. The model input is the road curvature, effectively harnessed through a self-developed Euler-curve-based curve fitting algorithm. A comprehensive case study is undertaken to empirically validate the efficacy of the proposed model, demonstrating its capacity to emulate the average behavioral patterns observed in human curve path selection. Statistical analyses further underscore the model's robustness, affirming the authenticity of the established relationships. This paradigm shift in trajectory planning holds promising implications for the seamless integration of autonomous driving systems with human driving preferences.

Keywords Naturalistic driving · Identification · Driver models · Path planning

Abbreviations

ACC	Adaptive cruise control
ADAS	Advanced driver assistance systems
LDM	Linear driver model
LKA	Lane keep assist

1 Introduction

Development of Advanced Driver Assistance Systems (ADAS) has accelerated in the last decade. Next to premium brands, more and more vehicle manufacturers decide to develop and sell automated driving functions for safety and comfort. The European Union has made automated emergency braking and emergency lane keep assist functions mandatory starting from 2022. The openly available NCAP tests have been evaluating and rating active safety functions

for almost a decade. People travel more year-by-year, either due to working purposes or for free time activities. This clearly motivates the further development of automated driving functionalities.

A fundamental categorization of these functionalities distinguishes between longitudinal and lateral control functions. Longitudinal functions are realized by Adaptive Cruise Control (ACC). Many ACC systems are capable of driving in high traffic density in the speed range of 0 to 200 km/h, including stop-and-go functionality. On the other hand, lateral functions are typically denoted by Lane Keep Assist (LKA). This function can be activated automatically for emergency situations (only when vehicle crosses the lane edge), or can be continuously active, guiding the vehicle within the lane. The former is usually termed Emergency Lane Keep Assist, while the latter is referred as Active Lane Driving. This paper specifically delves into the realm of lateral control system, directing attention to the components integral to lateral control.

ADAS architecture has not changed significantly for decades. The main components of ADAS are detection, perception, behavior planning, motion planning and actuator control. This study specifically addresses the motion planning component, assuming the preliminary components, from detection to behavior planning, are available and work reliably. Previous research [1] has illustrated

✉ Gergo Ferenc Ignecci
gergo.igneczi@ga.sze.hu

¹ Vehicle Research Center, Szechenyi Istvan University, Egyetem ter 1, Gyor 9026, Hungary

² Institute for Computer Science and Control, Kende str. 13-17., Budapest 1111, Hungary

³ Robert Bosch Kft, Gyomroi str. 104-120, Budapest 1103, Hungary

diverse conceptualizations of the motion planning problem. The most used approach in automotive industrial solutions is the corridor-based approach, where the corridor forms a free space ahead of the vehicle. The corridor is defined by the lane edges as borders. The task is to plan a local path within the corridor based on safety and comfort considerations. The path must be kinematically feasible considering the vehicle control constraints.

One of the key tasks within the automated driving systems is to define a local trajectory, which describes the target path in front of the vehicle. Such a trajectory planning usually stands for designing a geometric target path and a respective kinematic profile (i.e., speed profile). Often the geometric target path is simplified to follow the midline of the lane. This is unnaturalistic, as human drivers, besides of driving on long straight roads, often do not follow such a strategy.

Literature has already presented approaches to combine various objectives (e.g., comfort, safety, efficiency) and human-likeness in trajectory planning. Several works used optimization techniques to provide trajectories that are optimal based on the above objectives [2–9], or do the same to generate optimal control trajectories [10, 11]. These approaches rely on assumption that human drivers also prefer to optimize such factors. However, optimizing on abstract kinematic quantities can be difficult in real time, and do not implicitly provide human-likeness. There are various solutions using machine learning algorithms [12–15]. Nevertheless, these are quite complex algorithms, while effective, tend to be computationally intensive.

The primary focus of this study is centered around the development of a resilient driver model capable of generating path geometries that emulate human driving behavior. For the current investigation, considerations of kinematic effects are excluded, which the authors plan to incorporate into the modeling framework in future iterations.

The rest of this paper is structured as follows: Sect. 2 gives an overview of existing solutions for the path planning problem. Section 3 introduces the observations based on real-world driving data, proposes a driver model structure, and discusses the implementation and calibration of the driver model. Section 4 illustrates validation tests through statistical analysis of the data and a case study. Finally, Sect. 5 concludes this study.

2 Literature Overview

In the field of driver models for path planning, there are only limited number of works in the literature. There are many driver models for motion controllers [16–20], concentrating on how drivers act on the actuators to realize motion targets. However, they do not cover how the target path is generated.

While driver trajectory models have been proposed with the goal of planning trajectories preferred by human drivers, they frequently overlook the human-like nature of the path and often optimize abstract kinematic quantities of the trajectory.

Various machine learning techniques are employed for generating human-like paths. The Inverse Reinforcement Learning (IRL) method, for instance, is used to train models on human-like path set [15]. The proposed reward function in IRL has four terms for both longitudinal and lateral planning: velocity keeping, lane keeping, lane boundary keeping, and collision avoidance. While this approach yields favorable results in complex traffic scenarios, it still misses the essential path selection preferences of drivers. The lane-keeping term calculates an error between the vehicle and the lane center, which may not align with the preferred path of human drivers.

There is a trajectory planner using multiple Euler-curves optimized on lateral jerk and acceleration to design a curve trajectory [7]. By minimizing the lateral acceleration and jerk, this trajectory aims to provide high level comfort for passengers. However, this solution lacks the connection to actual human-chosen paths and rely solely on the fact that minimization of acceleration and jerk is sufficient to meet passengers' requirements of smooth driving. Similar optimization criteria are used for Ref. [5]. A path model is proposed that has a smoothing spline to reduce lateral jerk and acceleration in curve transitions [8]. The article states that smoothing is a sufficient criterion for human-like path selection.

A method is introduced to generate a global trajectory which is optimized for racing, aiming to reduce lap time by increasing the curve speed of the vehicle [9]. The article states that path curvature minimization is the ultimate way to reduce the centripetal force on the vehicle, thus maximizing curve speed. This is an important statement as it indicates curvature is one of the key factors influencing the trajectory shape.

There are only a few works providing planning solutions for human-like path geometry. In Ref. [6], it has been shown how human like trajectories can be generated for lane change maneuvers. The planner provides a combined solution, where optimization on different cost terms, e.g., comfort, stability, and dynamics happens. However, an additional term is added to the cost function, which is associated with the distance error between the drivers' path and the planned path. The cost weights are learned through optimization on a reference dataset, resulting in human-like path shapes achieved through optimization and learning.

In Ref. [21], a driver path model is introduced, which is trained on real-world driving data. The article has shown that there is a linear relation between the lane offset

selection and the lane offsets from the previous 2 s. The proposed linear model is structured according to Eq. (1).

$$D_i = a_{i-1}D_{i-1} + a_{i-2}D_{i-2} + a_{i-3} + e_i \tag{1}$$

where i is the calculation cycle, a_{i-1} , a_{i-2} , and a_{i-3} are the regression coefficients. D_i , D_{i-1} , and D_{i-2} are the lane offsets. e_i is a random error. The coefficients are calculated based on the data of 30 drivers in one given curve. This study has shown that drivers tend to select lane offsets based on two consecutive points in a recursive manner. Based on these results, it can be assumed that the lane offset change in three consecutive points (D_{i-2} , D_{i-1} , and D_i) can reflect the orientation change, which is directly connected to the curvature. Therefore, curvature may play an important role in the definition of the lane offset chosen by drivers.

The literature review reveals two primary approaches to achieving human-like path planning:

Trajectory planning based on optimization of various kinematic quantities, which eventually produce a path that is close to driver selected path in its geometry.

Direct path planning approaches, where perceived information is used to produce a path that preserves the geometry preferences of human drivers. These solutions usually give a direct relation between human-likeness and the resulting path. Hence, they provide a more accurate replicate of human path planning. However, there are only a few works available for this selection approach.

To the best of the authors' knowledge, there is currently no existing solution that models the fundamental behavior of human path planning. The proposed approach aims to provide a model to describe the geometric planning strategy employed by human drivers.

3 Driver Model

This Section introduces the assumptions based on which the driver model structure is formulated. Then, it provides information of the data that is used for analysis. Observations are detailed, then the model structure is set up based on these observations.

3.1 Model System

This research primary focuses on developing a path planning model designed to reproduce human-like path geometries. The model system relies on the following assumptions:

- Assumption 1: Drivers follow a planning–control approach.

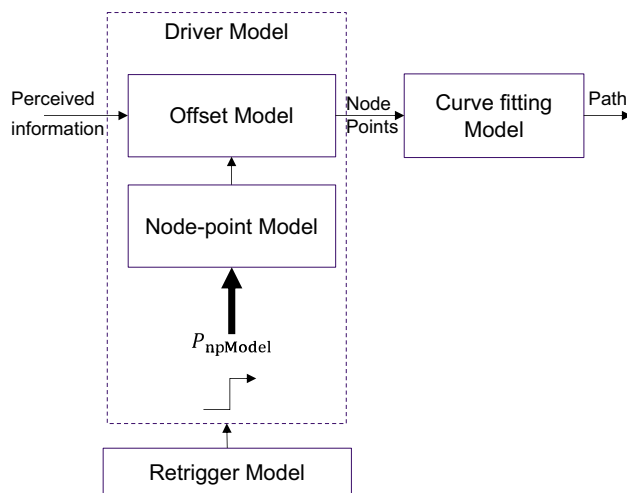


Fig. 1 Proposed model system for the curve path model

- Assumption 2: Drivers perceive environmental information, although they do not use all the information, focusing specifically on information from nominated road points (node point model).
- Assumption 3: Between nominated road points they follow a preconditioned behavior (i.e., one given curve type—curve fitting model).
- Assumption 4: They plan their behavior based on the perceived information (offset model).
- Assumption 5: They do replanning, if needed (retrigger model).

The proposed model system can be seen in Fig. 1, where the problem is systematically addressed in four main components. Based on the second assumption, recognizing that drivers perceive information within a preview distance, the total information is streamlined into discrete node points. The number of points may vary. The model that produces the node points is called the node point model. Based on the third assumption, drivers plan their behavior based on the condensed information and then follow the planned path according to instinctive preferences (comfort, speed, etc.). Therefore, a curve fitting model is introduced, which fits a human-preferred shaped curve onto the node points. According to the fourth assumption, drivers plan their behavior based on what they perceive from the preview distance. This can be simplified to planning the offsets compared to the mid-lane in the node points, which is called an offset model.

Using the fifth assumption, drivers replan their path occasionally. This is called a retrigger model.

The following sections provide details of the data and observations to identify the structure of the model elements.

3.2 Data Analysis

The aim of this work is to identify a driver model based on real-world driving data, which can replicate human-like path geometry. As seen in Fig. 1, the driver model system is divided into model elements. The two core elements of the driver model are the node point model and the offset model. To set up the structure of these models, the data of 15 drivers have been collected. Each driver drove approximately 40 km in real-world traffic. The test route lies on a two-lane road, Main Road 62, between cities of Dunaújváros and Székesfehérvár, in Hungary. The map is given in Fig. 2.

The test vehicle utilized in the study is a Skoda Octavia MK3 with an automatic gearbox. The measurements were conducted on workdays between 10 AM and 2 PM, ensuring consistent and comparable conditions. The weather conditions were consistently dry, and visibility was adequate throughout the duration of the measurements. Traffic conditions were characterized by low density, and no traffic jams were encountered.

The vehicle was equipped with a lane detection system, designed to measure road geometry. The shape of the road is defined by the lane edges, which, in the dataset, are represented by a third-order polynomial. The path at time point t can be given in Eq. (2).

$$\bar{y}_i = c_0^i + c_1^i \bar{x}_i + 2c_2^i \bar{x}_i + 6c_3^i \bar{x}_i \quad (2)$$

where c_0 is the lateral distance, c_1 is the orientation, c_2 is the curvature, and c_3 is the curvature change of the lane, \bar{x}_i is the lateral coordinate and \bar{y}_i is the longitudinal coordinate of the foreseen road mid-lane in the planning frame, at calculation cycle i . The camera system has a look ahead distance of 150 m.

The following sign conventions are used:

- Positive offset—left side of mid-lane

Fig. 2 The test route, Main Road 62, between cities of Dunaújváros and Székesfehérvár in Hungary

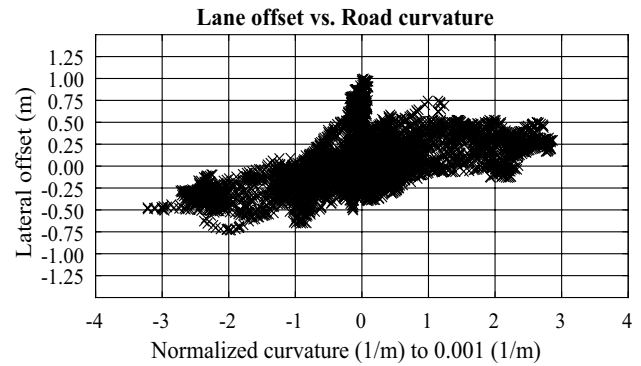


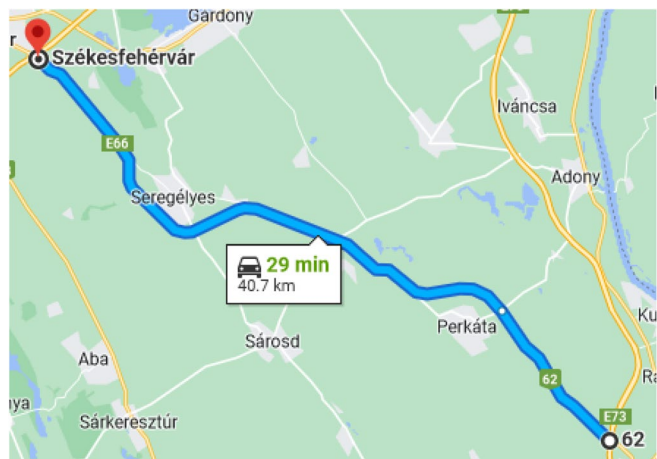
Fig. 3 Offset—curvature plot of a driver

- Negative offset—right side of mid-lane
- Positive curvature—left curve
- Negative curvature—right curve

3.3 Observations

It is assumed that drivers have different path selection behavior when driving in straight lines and curves. The key variable in the modelling problem is therefore assumed to be the road curvature. Similar statement has been made in other articles [6, 16, 21]. The road curvature is directly measured by the proposed lane detection system.

Figure 3 illustrates the relation between road curvature and selected driver offset. The entire route data of Driver 1 has been plotted. Curvature is positive in left curves, negative in right curves. Offset to the mid-lane is positive when the vehicle is on its left side, and negative on the right side. Among the test drivers, a notable consistency in behavior was observed, where several drivers exhibited similar patterns. The offset appears to be correlated with road curvature, and this correlation exhibits a close-to-linear trend: as the curvature increases, so does the selected offset from



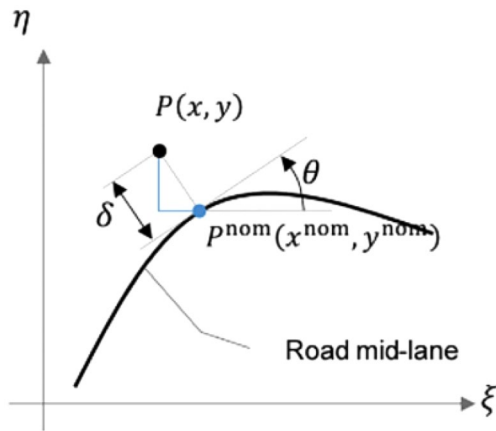


Fig. 4 Visualization of road offset value

the mid-lane. This suggests a tendency towards a behavior known as curve cutting.

However, it's evident that, for a given curvature, the driver selects multiple offset points. This can be due to the fact that the figure contains the data of the entire test route. Alternatively, it implies that a scalar function between the road curvature and the lane offset cannot adequately describe the path planning problem. Therefore, this paper proposes employing a multi-dimensional model that can produce such non-scalar function behavior. Consequently, the number of node points will be higher than one. The model takes road curvature as input and produces lane offset as its output. The modelling problem is formulated in Eq. (3).

$$y(\xi) = y^{nom}(\xi) + \delta(\kappa)\cos(\theta(\xi))$$

$$x(\xi) = x^{nom}(\xi) - \delta(\kappa)\sin(\theta(\xi))$$

$$\delta(\kappa) = f(\kappa(\xi)) \tag{3}$$

where $[\xi, \eta]$ is a freely chosen coordinate frame. $[x^{nom}, y^{nom}]$ defines the nominal node point on the middle of the road. $[x, y]$ defines the final node point corrected with the offset value, δ is the offset value, θ is the road orientation angle, κ is the road curvature, $f(\kappa(\xi))$ is the model function. The above quantities are visualized in Fig. 4.

3.4 Model Structure

Based on the aforementioned observation, the following model structure is proposed for the node point model and the offset model:

- Nominate three node points within the maximum preview distance (node point model).

- Use road curvature as the key variable in the modelling problem.
- Calculate the offset to the mid-lane in the node points (offset model).

When selecting the number of node points, it is important to minimize the complexity of the algorithm, which makes it attractive to choose low number of node points. On the other hand, a relatively high number offers more model flexibility. As seen in the previous chapter, a minimum of two node points is necessary. Various driver models in the motion control field use look-ahead point-based approaches with one or two look-ahead points [16–19]. Given the long preview distance (150 m) from the camera system, the number of node points is extended to three. It is believed that three node points within the preview distance provide sufficient flexibility while avoiding excessive computational complexity.

A trade-off analysis was conducted to evaluate the selection of different numbers of node points. A route section was simulated, configuring the system to fit a curve to node points positioned on the mid-lane of the road. This setup excludes errors arising from the variability of human offset selection. Two indicators were calculated:

- Mean error distance between points of the planned path and the mid-lane as reference.
- Mean time of the planning sequence.

One planning step happens at a retrigger, and includes node point distance calculation, offset calculation, and curve fitting, while the planning sequence encompasses all planning steps during the road. The number of node points have been varied between 1 and 10. Then, both indicators have been normalized by their maximum value through the node point range. The results, depicted in Fig. 5, reveal that the calculation time starts to increase significantly after 4 node points, while the mean error distance improves less and less with an increasing number of node points. Based on this observation, three node points is chosen in the simulation.

The three designated node points are termed the near, mid and far range points, respectively. The distances between these node points are specified as parameters of the model, as shown in Eq. (4).

$$P_{npModel} = [d_n \ d_m \ d_f] \tag{4}$$

According to the second assumption, drivers perceive environmental information, but they only consider the information about the nominated road points when making decisions. Considering three node points and the road curvature as the input to the model, it is suggested to take the average value of the road curvature between the node points. This

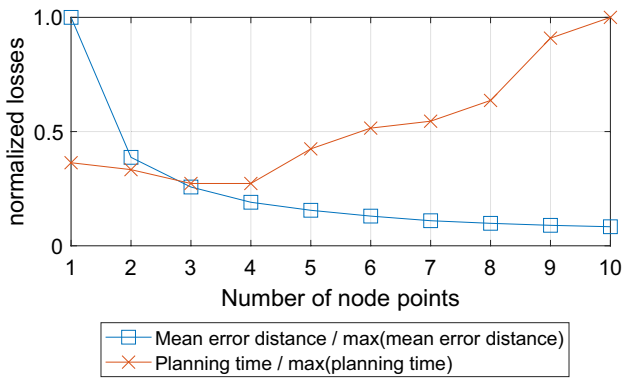


Fig. 5 Loss vs. calculation time of planning in the case of different number of node points

results in three average curvature values corresponding to the subsections defined by the three node points. For planning cycle i , the curvature input vector is as follows:

$$\bar{\kappa}_i = \begin{bmatrix} \kappa_{on_i} \\ \kappa_{nm_i} \\ \kappa_{mf_i} \end{bmatrix} \tag{5}$$

where on , nm , and mf stand for origin-near, near-mid, and mid-far subsection, respectively.

Based on the fourth assumption, drivers plan their behavior using the perceived information. This is an offset model. Based on the Eq. (3), the offsets are calculated in the node points compared to the mid-lane. In planning cycle i , this yields the following model output vector:

$$\bar{\delta}_i = \begin{bmatrix} \delta_{ni} \\ \delta_{mi} \\ \delta_{fi} \end{bmatrix} \tag{6}$$

where n , m , and f stand for near, mid and far range points.

The illustration of a generic example is shown in Fig. 6. The plot shows a generic example of the node point, curvature vector and offset vector selection. This is a curve entry scenario, where the planning frame is positioned in the straight section before the curve. In the preview distance, the curvature starts to increase. The node point offsets are always calculated as a perpendicular distance to the road mid-lane in the given node points.

Section 3.2 observed that there is a linear relation between the road curvature and the selected lane offset. Therefore, a linear offset model is proposed, Linear Driver Model (LDM), which calculates node point offset values based on the subsegment average curvature data. Equation (3) model function is given for the i th calculation cycle in Eq. (7).

$$\bar{\delta}_i = f(\bar{\kappa}_i) = P\bar{\kappa}_i \tag{7}$$

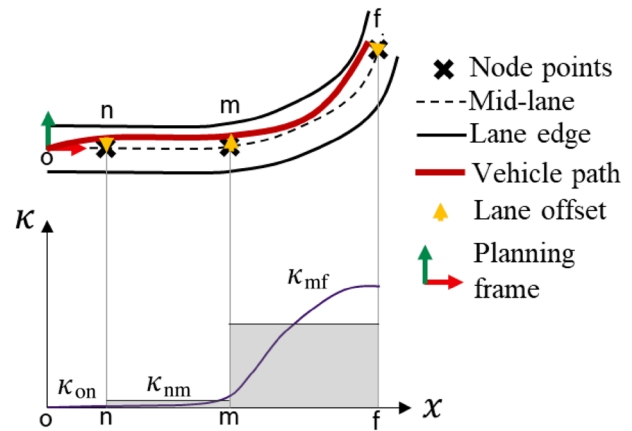


Fig. 6 Illustration of one planning cycle in a curve entry situation

where $P \in \mathbb{R}^{3 \times 3}$ is the parameter matrix that contains the weights of the linear combination of the curvature values. $\bar{\kappa}_i = [\kappa_{on} \kappa_{nm} \kappa_{mf}]^T$ is the input vector in the i th calculation cycle. κ_{on} , κ_{nm} and κ_{mf} are average curvature values in the subsections on , nm and mf respectively.

The combination of the node point model and the offset model formulates the complete driver model, as indicated in Fig. 1. The model is parametrized in terms of P and $P_{npModel}$. The assumption is that by selecting proper parameter values for both node point distances and the offset model, human-like paths can be generated.

3.5 Implementation

This section outlines a method for calibrating the driver model introduced in Sect. 3.3. While the previous discussions focused on structuring the model for planning a human-like path in a single cycle, it is crucial for real-time applications that planning occurs continuously during driving. Before delving into the calibration process, the implementation of the retrigger and curve fitting models is introduced.

Based on the fifth assumption, drivers occasionally replan, a process referred to as the retrigger model. For each replanning cycle, the actual perceived information is employed to design the node point offsets. It's suggested to use cyclic replanning, where node point calculation and curve fitting occur in even cycles, and the retrigger cycle can be varied.

Per the third assumption, drivers plan their behavior based on instinctive preferences between node points. Therefore, a curve is fit onto the node points. In prior work, Euler-curves have demonstrated the ability to describe the geometry of human paths with high accuracy [1]. This constitutes the curve fitting model.

Three Euler-curves are fit onto the node points. One curve is fitted onto each point pairs: on, nm, and mf. The Euler-curve can be fitted by four boundary conditions:

- Starting point position and orientation.
- End point position and orientation.

The position conditions are given by the node point positions, while the orientation condition is equal to the respective road orientation. The road orientation is provided by the lane detection system.

The implemented model and planner architecture can be seen in Fig. 7. The algorithm is implemented in MATLAB. Using pre-recorded data of the test drivers, simulation is done. The simulation has two different options:

- Model estimation simulations: the offset model is replaced by measurement data, therefore taking over the offset values selected by the driver.
- Model validation simulations: the offset model is implemented according to Eq. (7).

The driver model and the retrigger model have the following parameters:

$P_{npModel} = [d_n \ d_m \ d_f]$ —node point distance from the planner frame; $P \in \mathbb{R}^{3 \times 3}$ —the parameter matrix of the offset model;

$p_{retrigger}$ —the retrigger cycle of the planning.

Next, approaches are proposed to give the values of these parameters.

The node point distances can be selected based on various approaches. A relatively simple approach is to select node points within the preview distance which are equidistant to each other. However, such solution would not have any connection to the actual human curves. It has also been shown that a more accurate fit on drivers' paths can be achieved by selecting non equidistant node points. A previous work

has shown that the best fit on human path can be achieved if the node point distances are not equidistant but follow an increasing gap between each other [1]. As a result of an optimization executed on the data of three professional drivers, the following node point distances are calculated (values are shown in meters):

$$P_{npModel} = [d_n \ d_m \ d_f] = [10.0 \ 39.0 \ 137.0] \tag{8}$$

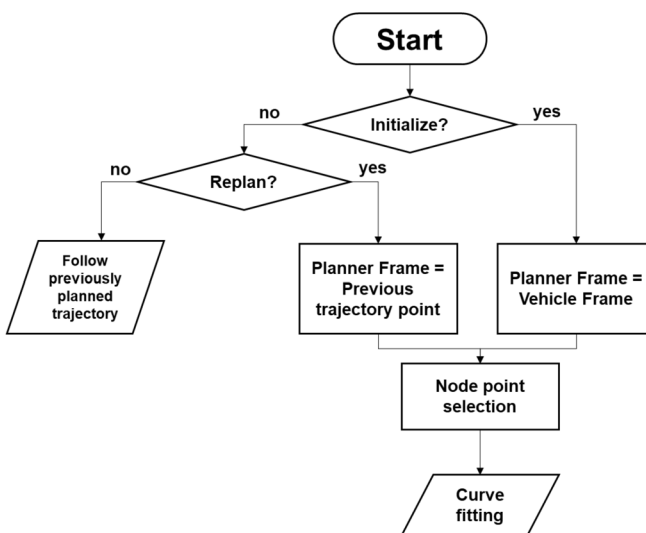
During the optimization process, we make the following steps:

- Three equidistant node points are positioned within a maximum preview distance of 250 m.
- A curve is fitted on the node points, constituting the local path.
- The curve points coincide with the exact points where the human paths are recorded.
- The mean distance between the fitted and the recorded path points are calculated; this represents the cost of the optimization.
- The node point distances are adjusted to minimize the cost.
- The same procedure is replicated for the entire test route, with a step size of 400 data samples, which is 8 s.
- For each optimization cycle, an optimal distance vector is calculated.
- Subsequently, all the distance vectors are averaged to obtain the overall distance vector, as given in Eq. (8).

The details of the optimization, the cost calculation, and the results can be found in Ref. [1].

The parameter matrix of the offset model can be calculated by linear regression on the human drivers' data. This provides path selection that is human-like. It's proposed to

Fig. 7 Implemented architecture of the model and the planning algorithm



accomplish linear regression for each driver separately, providing the best fit on personal path selection preferences. For this purpose, model in Eq. (7) can be reformulated as

$$\mathbf{D} = \mathbf{P}\mathbf{U} + \varepsilon \quad (9)$$

where ε is the regression error. $\mathbf{D} = [\overline{\delta_1} \overline{\delta_2} \dots \overline{\delta_N}]$, $\mathbf{D} \in \mathbb{R}^{3 \times N}$ is the node point offset matrix. $\mathbf{U} = [\overline{\kappa_1} \overline{\kappa_2} \dots \overline{\kappa_N}]$, $\mathbf{U} \in \mathbb{R}^{3 \times N}$ is the input matrix. N is the number of calculation cycles.

The value of \mathbf{P} is searched, so that ε would be minimal. This is equivalent to minimizing the 2 norm of $\varepsilon = \mathbf{D} - \mathbf{P}\mathbf{U}$. By the Moore–Penrose pseudoinverse of \mathbf{U} , the solution is:

$$\mathbf{P} = \mathbf{D}\mathbf{U}^T(\mathbf{U}\mathbf{U}^T)^{-1} \quad (10)$$

The even cycle of replanning is chosen to provide replanning frequently. Between two replanning, the previously planned path is followed. The replanning cycle is chosen based on two aspects:

- Following the previously planned path for an extended duration to leverage the planning characteristics.
- Initiating replanning as promptly as possible to allow flexible responses to upcoming perception information.

As the nominal vehicle speed in the test route is 25 m/s, the sampling time of the dataset is 0.05 s, and considering that the mid-range node point lies 39.0 m from the planning frame, the vehicle reaches the mid-range node point in 30 cycles. Therefore, the replanning cycle is set to 30 cycles, equivalent to 1.5s.

The following section conducts simulation runs to validate the model behavior, using the parameters obtained in this section.

4 Validation

4.1 Validation Concept

The validation of the driver model involves two distinct approaches. Initially, the model's performance is demonstrated through a case study, juxtaposing the human path of a specific driver with the model's output path in terms of lateral offsets and curvature. Subsequently, an extensive statistical analysis is conducted on the model's output, comparing it to human drivers.

4.2 Test Participants

Fifteen drivers were invited to participate in the driving study, selected through a random sampling process. Each driver traversed the identical reference route using the same vehicle, as outlined in Sect. 3.1.

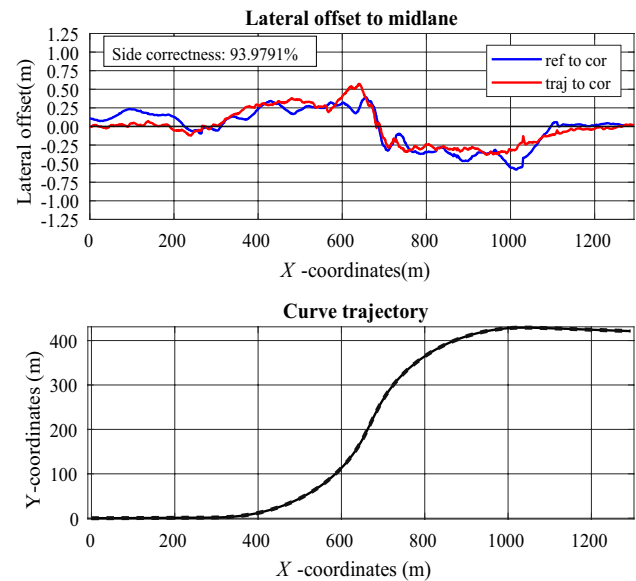


Fig. 8 Case study in an S-curve on country road(Comparing the lateral offset to the mid-lane of the planned ('traj') and the reference ('ref') path curvatures to the corridor ('cor'))

4.3 Case Study

Along the reference route, there are various curves with different radius. The highest curvature is measured in an S-curve combination. This curve is analyzed as a case study.

Figure 8 illustrates the reference offsets and the planned offsets to the mid-lane. During the curve approach phase, where the curvature is nearly zero, the planned path closely aligns with the mid-lane. Within the curve itself, the planned offset remarkably mirrors the reference, showcasing high accuracy. The planned path's characteristics closely resemble those of the reference path.

In Fig. 9, curvatures of the planned and the reference paths are compared. In the bottom subplot, the curvature difference to the corridor is depicted. The planned path exhibits similar curvature characteristics to the reference. Before the curve, there's a slight positive curvature deviation, indicating that the driver initiates cornering before reaching the curve entry. This behavior is reflected in the planned path's curvature. Throughout the cornering phase, there's a consistent negative curvature deviation, aligning with the reference path. In a left curve, a negative deviation suggests that the driver applies less steering angle than the mid-lane requires, made possible by the higher steering angle before entering the curve. At the peak point, the curvature begins to decrease, marking the start of the right curve. Both the reference and planned path curvatures remain lower than the corridor curvature, indicating that the driver pulls the vehicle more to the right and employs a higher steering angle than the corridor necessitates.

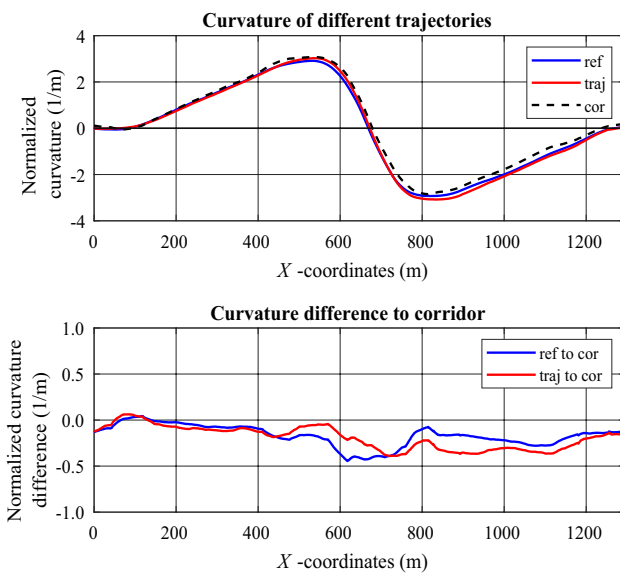


Fig. 9 Case study in an S-curve on country road (Curvature comparison of the planned ('traj') and the reference ('ref') path curvatures to the corridor ('cor'))

This case study demonstrates that in the S-curve with the highest curvature, the model effectively planned a path closely resembling the reference data in terms of both lateral offset and curvature. This implies successful reproduction of human curve path selection.

4.4 Performance Analysis

The statistical performance analysis includes two evaluation aspects: safety; deviation from the human paths.

To judge the safety of the planned path, the number of sample points is calculated where the planned path would result in the vehicle leaving the corridor. The ratio between violating points and the total number of sampling points are shown. If there are no violating points, the minimal distance of the vehicle body (edge points) to the corridor border is given, calculated separately for curve segments. Then, the worst case of the minimum border distances for the entire route is calculated. The lower this distance is, the less safe the path is.

To determine the performance of the planner, the following performance indicators are calculated:

- Euler distance: This involves measuring the Euclidean distance between the planned path points and the corresponding human path points. Both the maximum and average distances are computed, providing insights into the alignment of the planned and human paths.
- Side correctness: This metric evaluates the number of points where the planned path aligns with the same side

Table 1 Safety results of different drivers

Driver ID	Border violation (%)	Minimum border distance
Driver 1	0	0.458 m
Driver 2	0	0.422 m
Driver 3	0	0.627 m
Driver 4	0	0.313 m
Driver 5	0	0.156 m
Driver 6	0	0.578 m
Driver 7	0	0.143 m
Driver 8	0	0.448 m
Driver 9	0	0.473 m
Driver 10	0	0.380 m
Driver 11	0	0.439 m
Driver 12	0	0.025 m
Driver 13	0.19	0 m
Driver 14	0	0.439 m
Driver 15	0	0.438 m

of the mid-lane as the reference. The ratio of such points to the total number of sample points is determined.

It's important to note that these performance indicators are specifically calculated for curves. The test route is segmented into curve segments, and the evaluation focuses on these segments. A total of 10 curves per driver are considered for the analysis, contributing to a comprehensive assessment of the planner's performance in curve-related scenarios.

The safety results are displayed in Table 1. The nominal lane width is 3.70 m. In general, the model successfully reproduced safe paths for all drivers, except for Driver 13. For Driver 12, while there were no explicit border violations, the planned path resulted in a very low minimum border distance, indicating a potential safety concern. For the remaining drivers, the minimum border distance always maintained a margin for potential controller inaccuracies. Further analysis is required for Drivers 12 and 13 to understand why the model struggles to plan safe paths for them.

The performance analysis results can be seen in Table 2. The nominal lane width is 3.70 m. The best seven drivers, who has an average error distance less than 4 cm, and more than 55% side correctness are denoted by green. In their cases, the driver model seems to be working properly, and their paths could be reproduced by the model.

Drivers, whose error distance is above 5 cm, or their side correctness is below 50% are denoted by red. The paths of these five drivers could not be successfully reproduced by the model. The offset – curvature correlation plot of one of the deviating drivers can be seen in Fig. 10.

Table 2 Statistical results of different drivers: performance evaluation

Driver ID	Average Distance	Side correctness
Driver 1	0.0204 m	66.56 %
Driver 2	0.0702 m	59.96 %
Driver 3	0.0529 m	52.57 %
Driver 4	0.0458 m	52.79 %
Driver 5	0.0391 m	55.45 %
Driver 6	0.0185 m	64.79 %
Driver 7	0.0519 m	56.42 %
Driver 8	0.0842 m	52.44 %
Driver 9	0.0379 m	59.17 %
Driver 10	0.0482 m	45.10 %
Driver 11	0.0230 m	68.32 %
Driver 12	0.0710 m	60.78 %
Driver 13	0.0577 m	53.28 %
Driver 14	0.0343 m	55.40 %
Driver 15	0.0290 m	53.00 %

In this plot, Δy_n is plotted against κ_{on} in each replanning cycle, over the entire test route. Blue solid lines are the linear regression over the left and right curve planes. There are two main differences compared to the behavior of the proposed offset model:

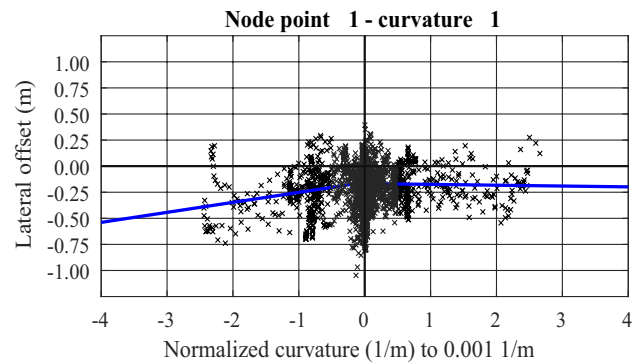
- There is an offset for zero curvature, which means they may have a side preference in straight road sections.
- The behavior is different on the left curvature plane than on the right curvature plane, which indicates that drivers may behave differently in right and left curves.

These effects must be analyzed further, and the model may be modified in the future to improve its accuracy.

5 Conclusions

This paper aims to propose a driver model coupled with a curve fitting algorithm for generating human-like paths. The problem is delineated into four main components: the node point model, the offset model, the retrigger model, and the curve fitting model. Environmental information is condensed into node points, and the offset relative to the mid-lane is computed at these points.

Three node points within a lookahead distance of 150 m are suggested, providing flexibility for planning various

**Fig. 10** Correlation plot of Driver 8

trajectory shapes while avoiding high computational complexity. Analysis of data from 15 drivers reveals a significant influence of road curvature on lane offset selection. The relation between road curvature and selected offset is assumed to be linear, leading to the proposal of a Linear Driver Model. This model calculates node point offsets as a linear combination of average curvature values between the points. Euler-curves are then fitted onto the node points, with the procedure repeated in even retrigger cycles, following the previously planned path between cycles.

The algorithm is implemented in MATLAB, and parameters for simulation are provided for the node point model, the offset model, and the retrigger model. Node point distances are determined based on previous work, offering the best geometric fit to human drivers' paths. The resulting distances exhibit increasing gaps. The offset model's parameter matrix is calculated via linear regression for each driver separately. The retrigger cycle is chosen for flexible adaptation to upcoming preview information and to exploit planned path geometry.

While the model generally performed well, there are gaps in the results. For some drivers, the average distance error was too high, or the side correctness of the path was inaccurate. For a few drivers, the resulting path was deemed unsafe.

Future improvements to the model could include:

- Developing a sophisticated node point model that calculates points dynamically based on environmental information.
- Implementing an event-based retrigger model.
- Conducting a trade-off analysis regarding the number of node points.

Despite these gaps, the results are considered satisfactory, contributing to human-like curve path modeling with a simple, robust solution. The approach is seen as potentially valuable for real-time applications in the future.

Acknowledgements The research was supported by the European Union within the framework of the National Laboratory for Autonomous Systems. (RRF-2.3.1-21-2022-00002).

Funding Open access funding provided by Széchenyi István University (SZE).

Declarations

Conflict of interest On behalf of all the authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Igneczki, G., Horvath, E.: Node point optimization for local trajectory planners based on human preferences. In: Paper presented at the IEEE 21st world symposium on applied machine intelligence and informatics, Herlany, Slovakia, 19–21 (2023)
- Xiong, L., Fu, Z., Zeng, D., Leng, B.: An optimized trajectory planner and motion controller framework for autonomous driving in unstructured environments. *Sensors* **21**, 1–16 (2021). <https://doi.org/10.3390/s21134409>
- Takayuki, O.: Multimodal trajectory optimization for motion planning. *Int. J. Robot. Res.* **39**, 983–1001 (2020). <https://doi.org/10.1177/0278364920918296>
- Liu, X., Liang, J., Zhang, H.: Dynamic motion planner with trajectory optimisation for automated highway lane-changing driving. *IET Intel. Transp. Syst.* **14**, 2133–2140 (2021). <https://doi.org/10.1049/iet-its.2020.0465>
- Bae, I., Moon, J., Seo, J.: Toward a comfortable driving experience for a self-driving shuttle bus. *Electronics* **9**, 943–955 (2019). <https://doi.org/10.3390/electronics8090943>
- He, X., Xu, D., Zhao, H., et al.: A human-like trajectory planning method by learning from naturalistic driving data. In: Paper presented at the IEEE intelligent vehicles symposium (IV), Suzhou, China, 4–9 (2022)
- Lambert, E., Romano, R., Watling, D.: Optimal path planning with clothoid curves for passenger comfort. In: Paper presented at the 5th international conference on vehicle technology and intelligent transport systems, Heraklion, Crete, 3–5 (2019)
- Gu, T., Dolan, J.M.: Toward human-like motion planning in urban environments. Paper presented at IEEE IV. Symposium on intelligent vehicle, Dearborn, USA, 8–11 (2014)
- Braghin, F., Cheli, F., Melzi, S., Sabbioni, E.: Race driver model. *Comput. Struct.* **86**, 1503–1516 (2008). <https://doi.org/10.1016/j.compstruc.2007.04.028>
- Kiencke, U., Majjad, R., Kramer, S.: Modeling and performance analysis of a hybrid driver model. *Control. Eng. Pract.* **7**, 985–991 (1999). [https://doi.org/10.1016/S0967-0661\(99\)00053-2](https://doi.org/10.1016/S0967-0661(99)00053-2)
- McAdam, C.C.: An optimal preview control for linear systems. *J. Dyn. Syst. Meas. Contr.* **102**, 188–190 (1980). <https://doi.org/10.1115/1.3139632>
- Yu, C., Ni, A., Luo, J., et al.: A novel dynamic lane-changing trajectory planning model for automated vehicles based on reinforcement learning. *J. Adv. Transp.* **1**, 1–16 (2022). <https://doi.org/10.1155/2022/8351543>
- Zhang, J., Chen, H., Song, X., et al.: Reinforcement learning-based motion planning for automatic parking system. *IEEE Access Comput. Sci.* **8**, 154485–154486 (2020). <https://doi.org/10.1109/ACCESS.2020.3017770>
- Chen, L., Jiang, Z., Cheng, L., et al.: Deep reinforcement learning based trajectory planning under uncertain constraints. *Front. Neurobot.* **16**, 1–10 (2020). <https://doi.org/10.3389/fnbot.2022.883562>
- Siebinga, O., Zgonnikov, A., David, A.: A human factors approach to validating driver models for interaction-aware automated vehicles. *ACM Trans. Hum. Robot Interact.* **11**, 1–21 (2022). <https://doi.org/10.1145/3538705>
- Ungoren, A.Y., Peng, H.: An adaptive lateral preview driver model. *Veh. Syst. Dyn.* **43**, 245–259 (2005). <https://doi.org/10.1080/00423110412331290419>
- Conlter, C.R.: Implementation of the pure pursuit path tracking algorithm, The Robotics Institute Carnegie Mellon University, Pittsburgh, Pennsylvania (1992)
- Horvath, E., Kőrös, P., Hajdu, C.: Novel pure-pursuit trajectory following approaches and their practical applications. Paper presented at the 10th IEEE international conference on cognitive infocommunications, Naples, Italy, 23–25 (2019)
- Liu, C., Lee, S., Varnhagen, S., Tseng, E.H.: Path planning for autonomous vehicles using model predictive control. Paper presented at the IEEE intelligent vehicles symposium (IV), Redondo Beach, CA, USA, 11–14 (2017)
- Wang, R., Li, Y., Fan, J., et al.: A novel pure pursuit algorithm for autonomous vehicles based on salp swarm algorithm and velocity controller. *IEEE Access* **8**, 166525–166540 (2020). <https://doi.org/10.1109/ACCESS.2020.3023071>
- Li, A., Jiang, H., Li, Z., Zhou, J., Zhou, X.: Human-like trajectory planning on curved road: learning from human drivers. *IEEE Trans. Intell. Transp. Syst.* **21**, 3388–3397 (2020). <https://doi.org/10.1109/TITS.2019.2926647>



Gergo Ferenc Igneczki has graduated as Control Engineer at Budapest University of Technology and Economics and learnt master studies in Vehicle Engineering at Széchenyi Istvan University of Győr, Hungary. He is a PhD student since 2021. As the part of his research, he studies the motion planning and control strategies of human drivers. Also, he is an active software engineer at Robert Bosch Kft, Budapest, Hungary since 2016, developing driver models and human-like motion planners and controllers for Advanced Driver Assistance Systems.



Erno Horvath received his MSc as Computer Engineer in System Engineering specialization from Széchenyi István University, Győr, Hungary in 2016 and his PhD in 2020. Since 2014 he has been working at the Department of Computer Science, where he was Laboratory engineer from 2014, assistant lecturer from 2016. From 2020 he is a research scientist at the Research Center of Vehicle

Industry. His research interests include robotics, autonomous vehicles, AI, optimization, perception and planning. He currently holds lectures about robotics and ROS in BSc and MSc level. Now he is actively working on autonomous vehicle perception and control.



Roland Toth received his Ph.D. degree with Cum Laude distinction at the Delft Center for Systems and Control (DCSC), Delft University of Technology (TUDelft) in 2008. He was a Post-Doctoral Research Fellow at TUDelft in 2009 and at the Berkeley Center for Control and Identification, University of California, Berkeley in 2010. He held a position at DCSC, TUDelft in 2011-12, then he joined to the Control Systems (CS) Group at the Eindhoven University of Technology (TU/e). Currently, he is an Associate Professor at the CS Group, TU/e and a senior researcher at the Systems and Control Laboratory, Institute for Computer Science and Control

(SZTAKI) in Budapest, Hungary. He is an Associate Editor of the IEEE Transactions on Control Systems Technology and he was the general chair of the 3rd IFAC Workshop on Linear Parameter-Varying Systems in 2019. His research interests are in identification and control of linear parameter-varying (LPV) and nonlinear systems, developing machine learning methods with performance and stability guarantees for modeling and control, model predictive control and behavioral system theory. On the application side, his research focuses on advancing reliability and performance of precision mechatronics and autonomous robots/vehicles with LPV and learning-based motion control.



Krisztian Nyilas has master's degree (2003) and Ph.D. (2010) in Physics from university Eötvös Loránd. He is currently a section manager in development at Robert Bosch Kft., working in driver experience domain within Cross-Domain Computing Solutions. He has experience in vehicle controls systems especially in the field of active and passive safety and driver assistance. His research interest is vehicle safety systems and autonomous vehicles.