

Inverse Model Predictive Control (IMPC) based Modeling and Prediction of Human-Driven Vehicles in Mixed Traffic

Longxiang Guo, *Student Member, IEEE* and Yunyi Jia, *Senior Member, IEEE*

Abstract—Modeling and predicting human-driven vehicle behaviors are critically important for the planning and control of autonomous vehicles in mixed traffic which includes both autonomous and human-driven vehicles. Despite the tremendous efforts on this problem, the traditional general driver model-based approaches are subject to prediction accuracy issues and the state-of-the-art data-driven heuristic approaches are subject to scalability issues. To this end, this paper proposes a novel inverse model predictive control (IMPC) based approach to address these issues. The approach can learn the internal control process of human drivers through the automatic learning of their cost functions in a novel IMPC setup, which could result in improvements in both the accuracy and scalability. The approach was implemented and validated with realistic human driver studies. The experiments illustrated that the proposed approach could achieve a better accuracy and a better scalability for unseen scenarios compared to existing approaches.

Index Terms—prediction, human-driven vehicle states, inverse model predictive control, learning

I. INTRODUCTION

AUTONOMOUS vehicles have the potential of improving driving safety and energy efficiency by incorporating the advanced sensing, control and communication technologies. [1]-[3]. In the foreseen future, autonomous vehicles will still need to share roads with human-driven vehicles to form a mixed-traffic, which introduces safety challenges due to uncertainties in human driving behaviors. Therefore, prediction of the states of human-driven vehicles becomes critical because such information could be shared with nearby autonomous vehicles for them to better plan and control their behaviors.

Predicting the states of human-driven vehicles is challenging since they are the results of human actions and vehicle dynamics. Furthermore, human actions are underlined by the internal mechanisms of environment perception, information processing and decision making. Over the years many different approaches have been developed to model this process. The most commonly used ones derive general human driver models and iterate them over the prediction horizon with the vehicle dynamics model. In the case of predicting vehicle longitudinal states, car following models such as the Tampère (TMP) model

[4], Optimal Velocity Model (OVM) [5], Intelligent Driver Model (IDM) [6], and feedforward and feedback integrated model [7] have been proposed. The parameters of these control models will be identified from prior observations to make them behave like a human driver [8][9]. However, these approaches are trying to derive models to generate a smooth control output or capture the general driving behaviors of a group of similar drivers instead of an individual. These models are generally simple and there are not enough parameters in these models to capture the control process of humans. Consequently, their accuracy is moderate when being applied to individual drivers although they can be applied to interpret general human driving behaviors.

In recent years, many data-driven heuristic approaches have been proposed to model the behaviors of human-driven vehicles. Gaussian Mixture Models (GMM) [15]-[17], Hidden Markov Models (HMM) [18] and Particle Filter (PF) based methods [19] have been adopted to model and predict behaviors/states of human-driven vehicles. The most popular heuristic approaches are Artificial Neural Networks (ANN) based approaches. There are lots of different types of ANN being used for this vehicle motion modeling and prediction task, such as Dynamic Bayesian Networks [13][14], multilayer feedforward neural network [11], radial basis function network (RBFN) [10] and recurrent neural network (RNN) [12]. The biggest advantage of these approaches also becomes a challenge. They employ enough parameters to model individual drivers, but those parameters require a large amount of carefully prepared data to get properly trained. More importantly, these approaches mainly aim to reproduce the same driving behaviors or trajectories of human-driven vehicles as demonstrated during training. Thus, the scalability of such approaches is limited by the scenarios covered by the training data, and they consequently have difficulties to handle never-seen situations. The main reason lies in the fact that these approaches still do not incorporate the internal control process of a human driver, which limits their performance when extended to unseen scenarios.

Therefore, this paper proposes an inverse model predictive control (IMPC) based approach to model and predict the

This work was partially supported by the National Science Foundation under Grants CNS-1755771 and IIS-1845779.

L. Guo and Y. Jia are with the Department of Automotive Engineering, Clemson University, Greenville, SC 29607 USA (e-mail: longxig@clemson.edu; yunyj@clemson.edu).

longitudinal states of human-driven vehicles. Model Predictive Control (MPC) [20][21] is an optimal control method that utilizes cost function as optimization object to reproduce the control process during controls. IMPC is based on Inverse Optimal Control (IOC) [22][23] which tries to derive the optimal cost functions. Normally such optimal cost functions are linear combinations of human-defined features. Recently, several latest research works have tried to extend IOC to IMPC to derive the cost functions from control behaviors [24]. This paper is to leverage these attempts and further extend them to the modeling and prediction of longitudinal behaviors of human-driven vehicles during car following. The major contribution of the paper can be summarized as following:

- We propose a novel inverse model predictive control (IMPC) based approach to model and predict the longitudinal behaviors of human-driven vehicles in mixed traffic to achieve better accuracy and scalability.
- We propose a novel and effective cost function learning approach to derive the best cost function from a set of primitive costs in the IMPC.
- We experimentally implemented the proposed approach using human-in-the-loop experiments and comprehensively compared it with existing approaches to validate the effectiveness and advantages of the proposed approach.

In the rest of the paper, section II will present the framework of the proposed IMPC method. Section III introduces the detailed design of the IMPC for predicting a human-driven vehicle. Section IV presents the setup of the human-in-the-loop experiment and results analysis.

II. OVERVIEW OF IMPC-BASED FRAMEWORK

The human-driven vehicle (HDV) is a critical part in mixed traffic. In such mixed traffic, unlike connected and automated vehicles, the nowadays HDV usually has no communication capabilities with surrounding vehicles. However, it can measure headway information such as the speed of the lead vehicle (LV) and headway distance using its onboard sensors such as radars. Without a loss of generality for mixed traffic, the lead vehicle can be assumed autonomous, named lead autonomous vehicle (LAV). The prediction of the HDV's states can then be made based on the measured current states of both the HDV and the LAV. Sharing such predictions with other surrounding vehicles can help improve the driving safety, fuel efficiency and riding

comfort of the entire fleet.

The structure of the proposed IMPC-based predictor is shown in Fig. 1. It consists of a motion model that describes the behavior of the human-vehicle system, a cost function evaluation process that select the most suitable primitive costs for the cost function, and a higher-level optimization process that learns the weights and references for the cost function. The latter two processes form the IMPC.

We propose to construct a generic motion behavior model in the form of MPC which considers controls in the near future horizon instead of at instantaneous current time and can thus represent a more comprehensive motion behavior control process. Such process is very similar to the mindset of a human. The system dynamic model used for prediction represents the dynamics of the system, and the cost function resembles the human's preferences in controlling the system. In the longitudinal vehicle control task, the system dynamic model is a description of the human's perceived vehicle-road system, and the cost function describes the human's preferences and tendencies during car following. It is intuitive that the preferences and tendencies will vary a lot between different human drivers while the perceived vehicle-road system would roughly remain the same. Thus, it would be effective to mimic and eventually predict a human driver's behavior by adopting IMPC method to learn the cost function of the MPC model. In this paper, the IMPC process is finding the best primitive costs in the cost function and adjusting the weights and references of them to fit the 'preferences' in a human driver's mind.

The references of each primitive cost are given as a priori in existing IOC approaches. That is because when they are working as controllers, the control tasks are known, and the targets are fixed. By adjusting the weights of the cost functions, the IOC can find a best weighted cost function for that given task. However, such a setup may not work well when being used as a predictor. Preset references can reflect the preferences of the designer of the controller but can be different from those of the driver it is applied to. It is also difficult to identify the references using data statistic methods from the driver's driving data since it is hard to separate the driver's driving purposes during driving. In this paper, we propose to train the references of the cost function together with the weights using a high-level optimization to make IMPC suitable for predicting human-driven vehicle longitudinal states. Another challenge is that in existing IOC approaches, the basic terms and/or features in the

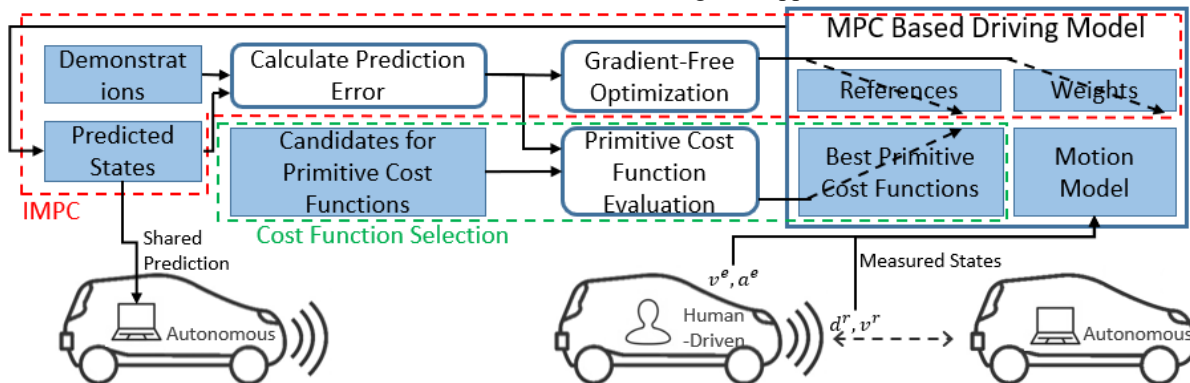


Fig. 1. IMPC-based predictor in mixed traffic

cost/reward function need to be selected through a trial-and-error process, which is troublesome and inefficient. Especially, a bad combination of primitive costs may make the higher-level optimization converge slowly or to a sub-optimal solution, or even fail to converge. In this paper we propose a cost function learning process that can derive the best cost function from a set of primitive costs.

Therefore, this paper will introduce an IMPC-based approach and its application in modeling and predicting the longitudinal states of human-driven vehicles in car-following scenario.

III. IMPC-BASED MODELING AND PREDICTION OF HUMAN-DRIVEN VEHICLES

A. Modeling of Human Driven Vehicle States in IMPC

This section describes the predictive model of the HDV. In physical traffic, the motion model of a vehicle can be represented by

$$x(k+1) = f(x(k), u(k)) \quad (1)$$

where x is the vehicle's states such as position and orientation, u is the vehicle's motion control input such as desired speed/acceleration and turning/steering rate, and k is discrete time index. Based upon this model, the HDV's motion behavior control in the form of MPC is to find a control sequence u_h^* over a horizon of N steps to minimize the HDV's cost function J . The motion behavior model of the HDV can be expressed by

$$\begin{aligned} u_h^* = \arg \min_{u_h} \sum_{\kappa=k}^{k+N} J(x_h(\kappa), x_a(\kappa), r(\kappa), u_h(\kappa)) \\ \text{s.t.: } x_h(k+1) = f_h(x_h(k), u_h(k)) \\ x_a(k+1) = f_a(x_a(k), u_a(k)) \\ x_h \in C_{x_h}, u_h \in C_{u_h}, x_a \in C_{x_a}, u_a \in C_{u_a} \end{aligned} \quad (2)$$

where $x_h(\kappa)$ is the states of the HDV, $x_a(\kappa)$ is the states of the LAV that interacts with the HDV, $r(\kappa)$ is a set of references of the outputs of the HDV-LAV system such as reference speed and reference headway distance. C_{x_h} , C_{u_h} , C_{x_a} , and C_{u_a} are the admissible sets for the state and control input (constraints) for the HDV and LAV.

Since the motions of vehicles and pedestrians are mostly on roads which usually do not have many terrain uncertainties to affect the motions, for simplification purpose, we use kinematic model to model their motion response. Specifically, for the longitudinal model of an HDV, (1) can be concretized as

$$\begin{bmatrix} s_h(k+1) \\ v_h(k+1) \\ a_h(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t_c & \frac{\Delta t_c^2}{2} \\ 0 & 1 & \Delta t_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_h(k) \\ v_h(k) \\ a_h(k) \end{bmatrix} + \begin{bmatrix} \frac{\Delta t_c^3}{6} \\ \frac{\Delta t_c^2}{2} \\ \Delta t_c \end{bmatrix} u_h(k) \quad (3)$$

where s_h , v_h and a_h represent the predicted displacement, speed, and acceleration of the HDV respectively, Δt_c is the iteration step time of the controller, u_h is the input to the HDV and its physical meaning is the longitudinal jerk of the vehicle. For the longitudinal model of the LAV, (1) can be concretized as (4):

$$\begin{bmatrix} s_a(k+1) \\ v_a(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t_c \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_a(k) \\ v_a(k) \end{bmatrix} \quad (4)$$

where s_a and v_a represent the predicted displacement and speed of the LAV, respectively. In this paper, we chose to use constant acceleration model which is a commonly used prediction model in (4). to describe the future motion of the LAV in the human driver's mind. Although this prediction may be different with the actual LAV motion in some cases during experiments, it represents how human driver thinks the LAV will move and using this prediction will help better model and predict human-driven vehicle behaviors than using the actual LAV motion which is also impossible for human driver to precisely obtain. The admissible sets, C_{x_h} and C_{u_h} , can be defined using the following constraints:

1) *Vehicle physical acceleration limits*: The maximum acceleration and minimum possible deceleration of the HDV:

$$a_{min} \leq a_h \leq a_{max} \quad (5)$$

2) *Speed limit*: The maximum possible speed of the HDV on open roads:

$$0 \leq v_h \leq v_{max} \quad (6)$$

3) *Relative distance limit*: The relative distance between the two vehicles should always be greater than the length of a single vehicle L to ensure feasibility of the solution:

$$d \geq L \quad (7)$$

where d is given by $d(k) = s_a(k) - s_h(k)$.

4) *Safety constraints*: Based on our previous work [9], inverse time to collision is a good indicator of driving safety. It should fulfill the condition:

$$TTCi \leq TTCi_{max} \quad (8)$$

where $TTCi$ is given by $TTCi(k) = [v_h(k) - v_a(k)]/d(k)$.

(3) and (4) specify the motion models, (5)-(8) specify the admissible sets, the last and the most important part to be specified in order to complete the MPC model is the cost function J . The process of finding J is the IMPC process to be introduced in the rest parts of section III.

B. Prediction of Human-Driven Vehicle States using IMPC and Perdition Performance Evaluation

This section mainly introduces how to predict human-driven vehicle states using IMPC and to evaluate the IMPC-based prediction performance once the IMPC-based model has been obtained, which is required for the cost function learning in the following section. Assuming the cost function J has been obtained, at each time t_m , the initial condition of MPC can be initialized with the current human driving states $x_h(t_m)$. $x_h(t_m) = [s_h(t_m), v_h(t_m), a_h(t_m), \dots]^T$ is a column state vector that contains multiple states at time t_m . Then the MPC can predict a trajectory by iterating the optimal control problem (OCP) defined in (2) and motion model in (1):

$$X_m^P = [x^P(t_m), \dots, x^P(t_m + n_p \Delta t_p)] \quad (9)$$

where n_p is the number of the prediction steps, Δt_p is the length of the prediction/control interval, X_m^P is the state matrix that contains all predicted vehicle states $x^P(\tau) = [s^P(\tau), v^P(\tau), a^P(\tau), \dots]^T$ in time order. Predicted trajectories

can be partially overlapped, or in other words, t_{m+1} can be smaller than $t_m + n_p \Delta t_p$.

The performance of the prediction can be evaluated by comparing the predicted trajectories to the actual human driving demonstrations. During an actual human driving cycle, the vehicle states are recorded as demonstration data and augmented into many short trajectories:

$$X_m^H = [x^H(t_m), \dots, x^H(t_m + n_H \Delta t_H)] \quad (10)$$

where t_m is the starting time of the demonstrated trajectory, Δt_H is the sampling time during data recording, n_H is the length of a recorded short trajectory, and X_m^H is a matrix that contains the values for all states between starting time t_m and end time $t_m + n_H \Delta t_H$. For the convenience of comparison, the time duration of the demonstration trajectory, $T_H = n_H \Delta t_H$, is selected to be the same as the prediction horizon, $T_P = n_P \Delta t_P$. Δt_H and Δt_P , n_H and n_P are not necessarily the same.

The error between the predicted trajectory and the reference trajectory is:

$$X_m^E = [x^E(t_m), \dots, x^E(t_m + n_E \Delta t_E)] \quad (11)$$

where $x^E(\tau) = |x^H(\tau) - x^P(\tau)|$ is the error matrix between the predicted trajectory and reference. Δt_E is the time step length of the error trajectory. $x^H(\tau)$ and $x^P(\tau)$ are the interpolated values from X_m^H and X_m^P at time $\tau = t_m + n \Delta t_E$. The total error between all predicted trajectories and reference trajectories is:

$$E = \frac{1}{M} \sum_{m=1}^M \frac{w}{n_E} \cdot \sum_{n=1}^{n_E} x^E(t_m + n \Delta t_E) \quad (12)$$

where w is a row weight vector for different states in vector $x^E(\tau)$, M is the total number of trajectories augmented from the complete human demonstration. The total error E can be used as a performance indicator that guides the learning of the cost function.

C. Learning of the Cost Function in IMPC

This section introduces the learning process in IMPC, including the cost function formulation, primitive cost learning and evaluation, learning of weights and references, and final determination of the best cost function.

1) Cost Function Formulation in IMPC

The cost function J is the core to describe the internal control process of a human drive vehicle i . Previous results in inverse reinforcement learning (IRL) approaches [25][26] have shown that the reward function in IRL can be approximated by a linear combination of a set of primitive reward functions. Inspired by this finding, we also propose to approximate the cost function in IMPC by a linear combination of primitive costs

$$J = \sum \Omega_h^T \Phi_h \quad (13)$$

where $\Phi_h = (\phi_1, \phi_2, \dots)^T$ are a set of primitive costs for the HDV and each specifies the cost on a particular motion objective such as tracking the reference speed, maintaining the look-ahead distance gap and minimizing the control efforts, etc., as shown in (14), and $\Omega_h = (\omega_1, \omega_2, \dots)^T$ are the associated weights.

$$\phi_j = g(x_h, x_a, r_j, u_h) = \sum_{\kappa=k}^{k+N} (y_j(\kappa) - r_j)^2 \quad (14)$$

In (14), y_j is the outputs of the HDV-LAV system, r_j is the reference if the output has one. It represents what the human driver wants y_j to be. If a human driver is not 'defining' r_j for y_j , then ϕ_j is not a good primitive cost to be included in the cost function. To learn this cost function, we first propose to evaluate a complete set of primitive costs for an HDV by an analysis of their motion behavior data. We then propose to down select the most important primitive costs for an HDV and then learn the associated weights of them to accomplish the learning of the entire cost function for the HDV.

2) Learning and Evaluating Primitive Costs in IMPC

If we consider all the possible outputs of the system (including all states, inputs, and combinations of states and inputs) to be valid y_j in the primitive costs, there will be consequently many weights and references to learn, which requires a large amount of data and training time for optimal converging. From our preliminary studies [29], we have observed that usually only a few primitive costs are important for the HDV. This section describes the selection of such important cost functions.

When a human driver is conducting a driving task, he/she may focus on and try to maintain some of the system outputs at desired target values while leaving the rest unattended. In this paper we propose to evaluate the primitive costs by using each of the system outputs independently as stand-alone cost functions, which can be written as:

$$J_{\phi_j} = \phi_j = \sum_{\kappa=k}^{k+N} (y_j(\kappa) - r_j)^2 \quad (15)$$

and then learning the reference r_j with a higher-level optimization:

$$\begin{aligned} r_j^* &= \arg \min_{r_j} E \\ \text{s.t.: } r_j &\in \mathcal{C}_{r_j} \end{aligned} \quad (16)$$

In this paper, the reference r_j is set as a constant parameter. The human drivers may sometimes have randomness and obscureness in operations, but these randomness and obscureness are usually occasional because human drivers usually have their own developed driving styles. Our proposed approach is trying to learn such styles including the reference r_j , which is the expectation of the distribution to represent the human driver in most of the time. Later in our experiments, it has also shown that human drivers may have some variations in their driving behaviors, which is where the errors come from. However, the overall performance has been shown good in our approach.

When the higher-level optimization finishes, a minimum prediction error E_{ϕ_j} over demonstrations will be obtained for primitive cost ϕ_j . If the human driver is focusing on ϕ_j and trying to maintain y_j at a specific target value during driving, then the resultant E_{ϕ_j} should be small, which means ϕ_j can be

a ‘good’ primitive cost in the final cost function. Otherwise, the resulted E_{ϕ_j} should be large, and ϕ_j might better be excluded from the cost function. All primitive costs can be ranked based on their E_{ϕ_j} values. By including the ‘good’ primitive costs and excluding the ‘bad’ primitive costs, the cost function can be formulated.

3) Learning of Cost Function in IMPC

Once the proper primitive costs have been selected, the weights and references of the cost function in MPC will be learned from human driving demonstration using a higher-level optimization. Denote the set of references r_j by R , the optimization can be expressed by (17). The total error E can be reduced by optimizing the weights Ω_h and references R in the cost function. Since only the relative values of weights are important, it is practical to fix one weight to 1 and optimize the remaining weights [27].

$$\begin{aligned} (\Omega_h^*, R^*) &= \arg \min_{\Omega_h, R} E \\ \text{s.t.: } \Omega_h &\in C_{\Omega_h}, R \in C_R \end{aligned} \quad (17)$$

The object function of this higher-level optimization is the evaluation result of the MPC which has a complex dynamic over a long horizon, its Jacobian is very difficult and nearly impossible to obtain. Thus, a gradient-free optimization method needs to be adopted. To solve the optimization problem, the Nelder-Mead Simplex (NMS) method [29] is used in this paper. NMS is an optimization method that requires no derivatives and does not require the objective function to be smooth. It has an advantage in the speed of convergence compared to other gradient-free optimization methods. The measurement of convergence used in this paper is the standard deviation of the evaluated error of each apex in the simplex:

$$\sigma = \sqrt{\frac{\sum_{p=1}^{N+1} (E_p - \bar{E})^2}{N+1}} \quad (18)$$

where E_p is the total error of apex p evaluated over the training data, $\bar{E}$ is the average error of all apexes, N is the number of parameters to be optimized, or in other words the number of apexes of the simplex. When standard deviation σ is no larger than a threshold value σ_{ter} , all the apexes of the simplex are very close to each other and the optimization process will be terminated.

4) Final Determination of Cost Functions in IMPC

In the last part of section III.C 2), the cost function is formulated using only the ‘good’ primitive costs. However, defining ‘good’ in this case can be very ambiguous. There can be various of different combinations of primitive costs. This section introduces how to select the best cost function from those different combinations.

We assume that $\Phi_h^* = (\phi_1^*, \phi_2^*, \dots, \phi_j^*)^T$ is the set of all available primitive costs that has been ranked from good to bad, with ϕ_1^* being the best and ϕ_j^* being the worst, as described in section III.C 2). Then we propose to formulate the cost function by first including the best primitive cost ϕ_1^* only, then

adding the other primitive costs one at a time from ϕ_2^* to ϕ_j^* , which can be described by (19).

$$\begin{aligned} J_1 &= \omega_1 \phi_1^* \\ J_2 &= \omega_1 \phi_1^* + \omega_2 \phi_2^* \\ J_3 &= \omega_1 \phi_1^* + \omega_2 \phi_2^* + \omega_3 \phi_3^* \\ &\dots \\ J_j &= \omega_1 \phi_1^* + \omega_2 \phi_2^* + \dots + \omega_j \phi_j^* \end{aligned} \quad (19)$$

Since humans normally focus on more than one aspect during driving, it is reasonable to skip J_1 and start with a combination of the top two or three best primitive costs in the cost function first, then try adding the next best primitive cost to the cost function in the following attempts. Every cost function J_j will learn its parameters using the method given in section III.C 3) and obtain an evaluation prediction error E_j . Adding an effective primitive cost ϕ_j^* should improve the prediction accuracy and reduce the error E_j while adding an ineffective primitive cost will not bring any benefit but affect the optimization convergence, which will result in a larger prediction error. Thus, the adding of primitive costs will be repeated until the evaluated performance of the predictor starts to decrease, then the previous cost function can be selected to be the best cost function.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experiment setup

This section introduces the equipment, environment, and design of the experiment. A 3D driving environment that includes two vehicles was constructed in Simulink. The lead car is autonomous and the one behind is controlled by the human subject in real-time. Both vehicles were built with complete longitudinal dynamics that is more comprehensive than (3) and (4). The communication between the ego vehicle and the lead vehicle is assumed to be ideal with no delays and packet losses. The impact on the autonomous vehicles behind the human driven vehicle is not studied in this paper. Instead, this section will focus on the performance of the predictor independently. The simulator used in this paper is shown in Fig. 2.



Fig. 2. A human driver operating our driving simulator

TABLE I
PARAMETERS USED BY MPC

Parameter	Value	Parameter	Value	Parameter	Value
Δt_p	0.5s	t_p	10s	Δt_s	0.2s
a_{min}	$-8m/s^2$	a_{max}	$4.5m/s^2$		
v_{min}	$0m/s$	v_{max}	$40m/s$		

The test track used in the simulation is a circuit with mainly straights and several turns. In each simulation, the HDV starts at the starting point of the straightway of the track, and the LAV starts 10 meters ahead of the HDV. Both vehicles' initial speed is zero. The LAV tracks three different driving cycles autonomously after the simulation starts. The first one is the EPA Highway Fuel Economy Test Cycle (HWFET), which is a mild highway cycle. The second is the Artemis Motorway 130 cycle which is an aggressive motorway cycle that requires heavier braking and wider open throttle. The last one is the New York City Cycle (NYCC) which is an urban driving cycle. Three human subjects (referred to as driver A, B and C hereafter) were required to drive the following vehicle in their preferred way and maintain a comfortable distance from the lead vehicle. Two of the participants were age 23 and the third participant was age 27. All three drivers had more than 3 years of clean driving records and have formed their own different driving styles. Two sets of data were collected from HWFET cycle, one set of data was collected from the Artemis cycle, and one set of data collected from NYCC cycle from each driver. The proposed IMPC based prediction approach is applied to each driver's data and generate dedicated predictor for them. The performance of the proposed IMPC based predictor is compared with IDM and ANN based predictors that are also trained for these three drivers separately. During the learning process for all approaches in the following, the first set of HWFET cycle data is used as training data. The other set of HWFET data is used to test the prediction performance in the seen situation. The data for Artemis and NYCC cycles is used to test the performance of predictors in unseen situations. In both learning and testing phases, the predicting horizon is chosen to be 10 seconds for all methods. In this paper, the MPC problem is solved using ACADO toolkit [28].

B. Results of Learning of IMPC-based Human-Driven Vehicle Models in Mixed Traffic

This section introduces the implementation and results of the proposed IMPC learning process. During longitudinal driving, two of the states of the HDV, speed v_h and acceleration a_h , and the control input u_h can be y_j in primitive costs ϕ_j . Vehicle displacements s_h is neglected due to the obvious fact that a human driver will not focus on the travelled distance of the ego vehicle only. The outputs of the two-car system, such as relative speed v_r , which is given by $v_r(k) = v_h(k) - v_a(k)$, headway distance d , time headway inverse $THWi$, which is given by $THWi(k) = v_h(k)/d(k)$, and time to collision inverse $TTCi$ are also potential primitive costs for y_j . These seven primitive costs are evaluated with the method described in section III using the HWFET cycle training data. The parameters used by MPC are listed in TABLE I. Δt_s is the sampling time-step. $\Delta t_s = 0.2s$ means that a predicted state trajectory is made every 0.2s in the test data. The evaluated E_{ϕ_j} values and references of these primitive costs for all three drivers are shown in TABLE II. The references for vehicle acceleration, relative speed and time to collision inverse could sometimes be negative. A negative acceleration reference indicates that the driver applies deceleration more aggressively than acceleration, a negative relative speed reference indicates

that the driver prefers to maintain a lower speed than the LAV, and a negative time to collision inverse reference indicates that the driver tries to avoid the possibility of colliding with the LAV.

One can see that a_h and u_h are the two 'good' primitive costs since their reference can be optimized such that the total prediction error E_{ϕ_j} is reduced to very low values for all three drivers. a_r and $TTCi$ are two decent primitive costs since their minimum E_{ϕ_j} are larger than those of a_h and u_h , but still relatively small. v_h , $THWi$ and d are three unsatisfactory primitive costs since they result in quite large minimum E_{ϕ_j} . TABLE II and Fig. 3 also show that for different drivers, their preferences during driving could be different. Driver A cares more about driving smoothness during driving since the E_{ϕ_j} values for his cost primitive costs a_h and u_h are the smallest among the three drivers, and the corresponding reference values are also the closest to zero among the three drivers. Driver C meanwhile is quite the opposite. The E_{ϕ_j} value for his primitive cost u_h is the largest among the three drivers. Meanwhile, the E_{ϕ_j} values for v_h , $THWi$ and d are the smallest among the three drivers. These indicate that driver C is trading driving smoothness for headway and speed tracking accuracy. Driver B's preferences during longitudinal driving is somewhere between driver A and C. v_h and $TTCi$ seem to be two commonly and equally considered primitive costs for all drivers since they have very similar E_{ϕ_j} and terminal reference values.

It is worth mentioning that the final reference value for v_h of driver C is 84.18m/s, which is clearly abnormally large during car following. However, that does not mean that the primitive cost evaluation is not valid. When ego vehicle speed is the only term in the cost function, the reference speed needs to be very large such that the MPC can generate a large enough control input to mimic driver C's aggressive driving style.

TABLE II
EVALUATION RESULTS OF DIFFERENT PRIMITIVE COSTS

Primitive costs	Driver A		Driver B		Driver C	
	r_j	E_{ϕ_j}	r_j	E_{ϕ_j}	r_j	E_{ϕ_j}
$v_h(m/s)$	23.098	1.284	22.387	1.430	84.180	1.096
$a_h(m/s^2)$	-0.011	0.553	0.088	0.542	1.768	0.633
$v_r(m/s)$	0.143	0.897	-0.252	0.901	-0.041	0.856
$d(m)$	70.678	2.553	33.885	2.560	24.445	1.410
$THWi(s^{-1})$	0.214	2.346	0.544	1.766	0.843	1.268
$TTCi(s^{-1})$	0.0039	0.946	-0.015	0.947	-0.021	0.971
$u_h(m/s^3)$	0.006	0.504	0.0107	0.698	3.700	0.652

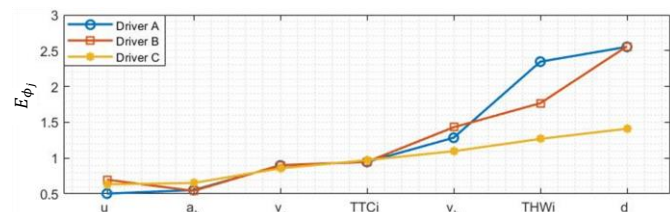


Fig. 3. Evaluation results of different primitive costs

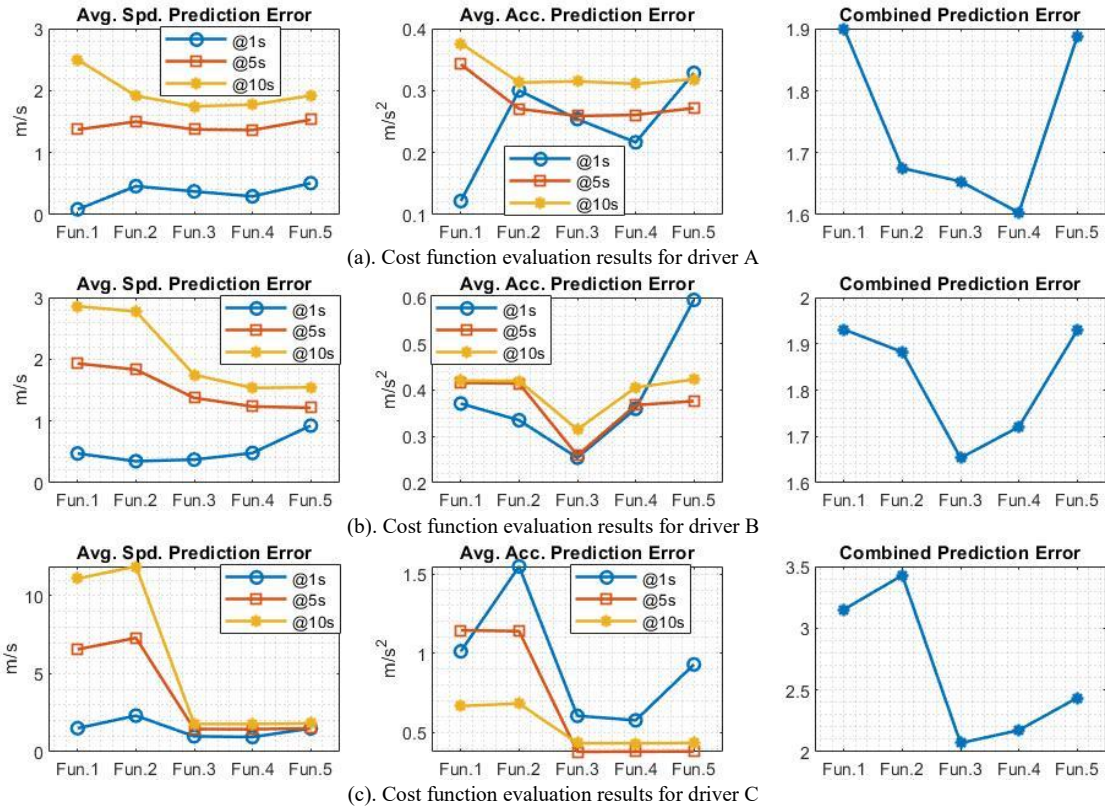


Fig. 4. Evaluation results of different cost functions for different drivers

Based on the principle described in III.C, we can gradually obtain five different combinations of different primitive costs based on their rankings as following:

- Cost function 1:
 $J_1 = \sum_{\kappa=k}^{k+N} w_{u_h}(u_h(\kappa) - r_{u_h})^2 + w_{a_h}(a_h(\kappa) - r_{a_h})^2$
- Cost function 2:
 $J_2 = J_1 + \sum_{\kappa=k}^{k+N} w_{v_r}(v_r(\kappa) - r_{v_r})^2$
- Cost function 3:
 $J_3 = J_2 + \sum_{\kappa=k}^{k+N} w_{TTCi}(TTCi(\kappa) - r_{TTCi})^2$
- Cost function 4:
 $J_4 = J_3 + \sum_{\kappa=k}^{k+N} w_{v_2}(v_2(\kappa) - r_{v_2})^2$
- Cost function 5:
 $J_5 = J_4 + \sum_{\kappa=k}^{k+N} w_{THWi}(THWi(\kappa) - r_{THWi})^2$

These five different cost functions are trained for each driver with the method described in section III.A 3) using the same MPC parameters in TABLE I. The initial guesses of the references are using the values obtained in TABLE II. The initial weights are all chosen to be 1 and the weight of the last primitive cost in the cost function is fixed during the training. The termination condition is selected to be $\sigma_{ter} = 3 \times 10^{-4}$. The performance of these five cost functions is evaluated using the HWFET cycle test data set. The terminal speed, acceleration, and combined prediction errors at 1s (1 prediction step), 5s (5 prediction steps) and 10s (10 prediction steps) are compared. The combined prediction error is defined by (20), where E_t^x is the trajectory of prediction error at horizon t for state x obtained from the whole test data. k_{Ops}^x are the normalization weights for the averages, standard deviations, maximum values, and minimum values of prediction error trajectories. k_{Ops}^x is selected based on the complete evaluation

data from all three drivers with all 5 cost functions at 3 different prediction horizons, such that $k_{Ops}^x Ops(E^x)$ has the same value as $mean(E^x)$. The reason for choosing this combined error is that we value the accuracy, stability, maximum positive error, and minimum negative error of the predictor equally. The evaluation results are shown in Fig. 4.

$$E_{combine} = \frac{E_{1s} + E_{5s} + E_{10s}}{3}$$

$$E_t = mean(E_t^v) + k_{std}^v std(E_t^v) + k_{max}^v max(E_t^v) + k_{min}^v min(E_t^v) + k_{mean}^a mean(E_t^a) + k_{std}^a std(E_t^a) + k_{max}^a max(E_t^a) + k_{min}^a min(E_t^a) \quad (20)$$

From the figures, one can see that for driver A, cost function 4 outperforms other cost function options in terms of combined error. It can provide the best speed and acceleration prediction accuracy at 5s and 10s horizons, and the 2nd best prediction accuracy at 1s horizon. It also provides more stable predictions and the smallest maximum prediction errors. For driver B, cost function 3 is the best among these five options. While it performs similarly as cost function 4 and 5 in terms of speed prediction, it is doing significantly better in acceleration prediction. It is also providing the smallest maximum prediction error. For driver C, cost function 3 again is the best. Its performance in speed and acceleration prediction accuracy is on par with cost function 4 and 5, but its ability in making stable prediction and controlling maximum prediction error makes it stand out.

From the results, the proposed approach can automatically derive that cost function 3 is the best for driver B and C, and cost function 4 is the best for driver A. The results have shown

TABLE III
PREDICTION PERFORMANCE FOR HWFET CYCLE (TRAINED CYCLE), DRIVER A

Model Type Error Type	IMPC-based Approach			Typical MPC based Approach			Intelligent Driver Model			Neural Network		
	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error
Predicted @1s	0.37	0.6	6.21/-2.2	0.06	0.22	4.34/-3.81	0.35	0.65	4.51/-10.25	1.06	2.66	10.68/-20
Speed @5s	1.37	1.95	10.5/-6.1	12.84	3.92	20/-2.07	1.67	2.32	12.62/-7.55	2.2	3.27	20/-20
Error @10s	1.75	2.46	13.72/-9.74	8.21	8.83	20/-9.09	2.52	3.55	16.12/-12.3	2.35	3.55	20/-20
Predicted @1s	0.25	0.43	5.3/-1.84	0.31	0.52	6.48/-1.18	0.34	0.59	2.68/-9.72	1.06	2.67	10.69/-20
Accelerat @5s	0.26	0.47	6.14/-1.06	4.04	1.3	11.15/-0.95	0.33	0.55	6.43/-1.29	0.86	2.78	20/-20
ion Error @10s	0.31	0.57	6.45/-6.01	5.75	3.3	5.95/-9.76	0.34	0.57	6.55/-1.22	0.67	1.91	20/-20

TABLE IV
PREDICTION PERFORMANCE FOR ARTEMIS CYCLE (NEW CYCLE), DRIVER A

Model Type Error Type	IMPC-based Approach			Typical MPC based Approach			Intelligent Driver Model			Neural Network		
	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error
Predicted @1s	0.54	0.89	6.54/-2.46	0.12	0.41	5.08/-6.39	0.51	0.81	5.77/-7.88	3.09	4.29	19.38/-20
Speed @5s	1.8	2.58	12.9/-9.37	11.71	4.54	20/-4.91	2.4	3.38	15.73/-11.5	8.2	9.29	20/-20
Error @10s	2.31	3.34	16.8/-12.03	14.84	16.8	20/-20	3.56	4.88	18/-16.12	6.81	8.75	20/-20
Predicted @1s	0.37	0.65	5.39/-1.72	0.45	0.82	6.48/-1.64	0.49	0.75	6.42/-6.64	3.1	4.3	19.46/-20
Accelerat @5s	0.44	0.83	6.45/-3.9	2.28	2.24	11.03/-2.81	0.51	0.9	6.54/-1.97	5.73	8.94	20/-20
ion Error @10s	0.51	0.95	6.78/-4.35	3.66	4.16	7.44/-10.32	0.53	0.94	6.49/-1.89	4.98	7.98	20/-20

TABLE V
PREDICTION PERFORMANCE FOR NYCC CYCLE (NEW CYCLE), DRIVER A

Model Type Error Type	IMPC-based Approach			Typical MPC based Approach			Intelligent Driver Model			Neural Network		
	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error
Predicted @1s	0.96	1.48	6.46/-2.84	0.61	1.41	5.54/-6.41	0.49	0.84	5.85/-2.09	2.55	1.8	5.07/-10.3
Speed @5s	1.9	2.51	10.23/-7.02	9.82	6.63	20/-7.24	2.6	3.58	11.36/-9.83	3.23	4.96	20/-20
Error @10s	2.96	3.91	12.2/-11.05	16.8	8.45	20/-10.6	3.58	4.65	12.34/-20	3.49	4.5	14.46/-20
Predicted @1s	0.91	1.63	8.87/-2.37	0.87	1.53	6.46/-2.38	0.89	1.58	6.42/-2.08	2.53	2.23	4.95/-9.02
Accelerat @5s	1.03	1.67	7.16/-2.77	4.03	2.52	10.9/-3.21	0.99	1.71	6.47/-2.04	3.04	5.01	20/-20
ion Error @10s	1.04	1.73	6.69/-4.31	1.94	2.39	8.82/-6.27	1.03	1.95	6.5/-20	1.6	2.32	20/-20

that the proposed IMPC learning process can effectively find the most suitable cost function for different human drivers. In section V.C., further prediction results will show that adding the ‘bad’ primitive costs to or removing an important ‘good’ primitive cost from the cost function will weaken the predictor’s performance.

C. Comparison of IMPC-based Approach to Typical MPC based Approach

In this section, the performance of the proposed IMPC-based approach is compared with the typical MPC based approach. In typical MPC-based approach, the form of the cost function is usually fixed and pre-determined based on heuristic experience. In this paper, we have chosen the commonly adopted cost function in MPC-based vehicle longitudinal control [31] as the cost function which incorporates ego vehicle speed, acceleration, jerk and headway information. The weights and other unknowns in this typical MPC cost function are learnt using the same way as proposed in Section III.C.3). We then base this typical MPC formulation to predict the vehicle behaviors using the same way as proposed in Section III.B and compare its prediction performance with our proposed IMPC-based approach. The performance comparison is given from TABLE III to TABLE XI in the first two data columns. The predicted speed error or predicted acceleration error sometimes may not be increased as the prediction window increases from 1s to 10s, which is because the predictor happened to match its predictions with actual values at a certain prediction horizon under one driving cycle for one driver.

TABLE III to TABLE V shows the predictors’ performance

for driver A. TABLE III shows the prediction performance over HWFET cycle. One can see that the while the typical MPC is having a slight advantage over IMPC at 1s prediction horizon, our IMPC-based approach out-performs the typical MPC in every aspect in both speed and acceleration prediction at 5s and 10s prediction horizons by a huge margin, meaning that it is not only more accurate but also more stable than the typical MPC. TABLE IV and TABLE V show the predictors’ performance over Artemis cycle and NYCC cycle. Artemis cycle has more aggressive braking and acceleration profiles than the HWFET cycle. The NYCC cycle is a totally different driving cycle. It has a lot of hard braking, heavy accelerating, stop-and-go, and the vehicle is operating in a speed range completely different from the HWFET cycle. The IMPC is showing significant advantages over the typical MPC in speed predictions over these two cycles again at 5s and 10s prediction horizons again. In fact, in all three cycles, the maximum speed prediction error of the typical MPC has reached the ceiling for this comparison (20m/s), which seems to indicate that the typical MPC is not making reasonable predictions at all. When we compare the prediction results of typical MPC with that of ANN, which will be discussed in detail in the next section, we see that the predicted acceleration of the typical MPC is relatively reasonable, but the predicted speed is outrageous. That means unlike the ANN, the MPC is not having scalability or training issues. The prediction error is purely caused by the improper primitive cost employed by the cost function.

TABLE VI to TABLE VIII show the predictors’ performance for driver B, TABLE IX to TABLE XI show the predictors’ performance for driver C. One can see that the

TABLE VI
PREDICTION PERFORMANCE FOR HWFET CYCLE (TRAINED CYCLE), DRIVER B

Model Type		IMPC-based Approach			Typical MPC based Approach			Intelligent Driver Model			Neural Network		
Error Type		Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error
Predicted	@1s	0.37	0.6	6.21/-2.2	0.16	0.4	4.55/-5.74	0.43	0.66	3.76/-6.3	0.5	0.93	5.04/-9.72
Speed	@5s	1.37	1.95	10.5/-6.1	12.82	4.06	20/-6.05	1.39	1.96	9.42/-7.36	1.34	1.95	9.64/-19.59
Error	@10s	1.75	2.46	13.72/-9.74	8.19	8.76	20/-19.57	1.85	2.8	13.6/-10.06	1.68	2.63	14.49/-20
Predicted	@1s	0.25	0.43	5.3/-1.84	0.39	0.63	6.59/-1.51	0.43	0.63	4.37/-5.88	0.53	1.01	5.47/-12.23
Accelerat	@5s	0.26	0.47	6.14/-1.06	4.02	1.4	11.13/-1.41	0.38	0.65	6.58/-1.46	0.48	1.21	20/-20
ion Error	@10s	0.31	0.57	6.45/-6.01	5.72	3.45	6.73/-9.56	0.4	0.67	6.64/-1.35	0.51	1.32	20/-20

TABLE VII
PREDICTION PERFORMANCE FOR ARTEMIS CYCLE (NEW CYCLE), DRIVER B

Model Type		IMPC-based Approach			Typical MPC based Approach			Intelligent Driver Model			Neural Network		
Error Type		Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error
Predicted	@1s	0.54	0.89	6.54/-2.46	0.18	0.59	5.45/-6.53	0.64	1.09	5.41/-20	0.81	1.6	17.96/-15.5
Speed	@5s	1.8	2.58	12.9/-9.37	11.5	4.48	20/-5.69	2.52	3.11	14.84/-20	2.61	3.44	19.07/-19.4
Error	@10s	2.31	3.34	16.82/-12	14.91	17.14	20/-20	3.72	4.55	17.97/-20	3.4	4.32	18.07/-20
Predicted	@1s	0.37	0.65	5.39/-1.72	0.49	0.9	7.1/-1.7	0.65	1.12	6.37/-20	0.84	1.69	18.85/-16.2
Accelerat	@5s	0.44	0.83	6.45/-3.9	2.27	2.29	11.13/-2.92	0.55	0.93	6.76/-1.89	0.87	2.26	20/-20
ion Error	@10s	0.51	0.95	6.78/-4.35	3.85	4.26	11.88/-9.95	0.55	0.96	6.87/-1.7	0.83	2.18	20/-20

TABLE VIII
PREDICTION PERFORMANCE FOR NYCC CYCLE (NEW CYCLE), DRIVER B

Model Type		IMPC-based Approach			Typical MPC based Approach			Intelligent Driver Model			Neural Network		
Error Type		Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error
Predicted	@1s	0.96	1.48	6.46/-2.84	0.99	1.72	5.28/-6.54	0.46	0.77	4.75/-1.87	3.13	3.65	19.02/-4.04
Speed	@5s	1.9	2.51	10.23/-7.02	10.38	5.76	20/-6.35	2.37	3.19	11.36/-7.48	4.98	5.8	17.48/-18.2
Error	@10s	2.96	3.91	12.18/-11	16.72	8.63	20/-14.48	3.45	4.41	13.07/-9.51	5.42	6.78	18.75/-20
Predicted	@1s	0.91	1.63	8.87/-2.37	1.15	1.74	6.48/-1.87	1.15	1.77	6.25/-2.97	3.1	3.32	20/-3.82
Accelerat	@5s	1.03	1.67	7.16/-2.77	4.07	2.64	11.65/-2.14	1.24	1.87	6.64/-2.04	6.29	8.46	20/-20
ion Error	@10s	1.04	1.73	6.69/-4.31	2.11	2.48	8.21/-4.86	1.24	1.88	6.66/-1.91	6.66	9.47	20/-20

IMPC again performs significantly better than the typical MPC in every aspect for both drivers, and under all cycles at 5s and 10s horizons. Such excellent performance is resulted from our optimal cost function being able to catch the human driver's preferences more precisely than the normally designed cost function.

The results show that a well-established MPC setup, which works very well for control, may not serve as a good predictor, and that our proposed IMPC-based approach can learn the best cost function for prediction and is superior to the typical MPC-based approach in terms of prediction performance.

D. Analysis of Prediction Accuracy and Scalability Compared to Existing Approaches

In this section, the state-of-the-art IDM and ANN models are used for performance comparison with proposed IMPC approach. We chose these two models since they achieved best speed prediction accuracy among existing driver model based and heuristic approaches according to [32].

Intelligent Driver Model is a widely used adaptive cruise control (ACC) model that can describe accelerations and decelerations in a satisfactory way. It has been used to simulate human driver behaviors in traffic simulation [33]. The acceleration function is given by (21), where v_0 is the desired velocity, d_0 is minimum desired spacing, T is desired time headway, a is maximum acceleration and b is comfort braking deceleration. These five parameters are tunable/trainable parameters of this model. The IDM model is trained for all three drivers with the HWFET cycle training data using NMS optimization method as described in section III.A.3).

$$a_{IDM} = a \left[1 - \left(\frac{v_h}{v_0} \right)^4 - \left(\frac{d^*}{d} \right)^2 \right] \quad (21)$$

$$d^* = d_0 + v_h T + \frac{v_h v_r}{2\sqrt{ab}}$$

The ANN model proposed in this paper is based on a feed-forward structure [11] with the hidden layer having 16 sigmoidal neurons and the output layer having linear neurons. The inputs to the network are the most basic system states $v_h(t)$, $v_a(t)$ and system output $d(t)$. The training is done by fitting the output of the network to the human demonstrated accelerations $a_h^R(t)$. The training data set is the same one that is used by all other 3 predictors. The training algorithm we used is Levenberg-Marquardt method [34]. The trained IDM and NN are making predictions in the same way as the IMPC based predictor. The prediction time step and reference evaluation time step are sharing the same settings as TABLE I. The performance of all predictions is shown in TABLE III to TABLE XI.

1) Comparison of Prediction Accuracy

TABLE III, TABLE VI and TABLE IX show the prediction results under HWFET cycle, which is the same cycle as the one used in the training data. Such results provide a straightforward comparison in prediction accuracy.

Under HWFET cycle, the speed and acceleration prediction accuracy of IDM and NN is worse than our IMPC based approach but still acceptable. In driver B' case, the average speed prediction accuracy of IDM and NN is almost comparable to that of IMPC. That is because although IDM and NN do not catch the internal preference of the human driver, they can still obtain a good prediction accuracy since they are

TABLE IX
PREDICTION PERFORMANCE FOR HWFET CYCLE (TRAINED CYCLE), DRIVER C

Model Type Error Type	IMPC-based Approach			Typical MPC based Approach			Intelligent Driver Model			Neural Network		
	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error
Predicted @1s	0.99	1.31	8.07/-4.36	0.12	0.28	1.92/-3.73	0.57	0.9	5.14/-7.1	0.88	1.39	5.59/-7.69
Speed @5s	1.45	1.94	9.6/-6.87	13.01	4.06	20/-7.59	1.87	2.54	11.97/-9.41	1.94	2.95	14.45/-20
Error @10s	1.79	2.64	11.59/-15	8.71	9.13	20/-20	2.21	3.14	15.53/-12.4	2.23	3.59	16.27/-20
Predicted @1s	0.6	0.8	5.37/-2.79	0.39	0.61	6.24/-1.4	0.56	0.94	6.56/-6.94	0.96	1.55	5.56/-9.49
Accelerat @5s	0.38	0.61	6.32/-3.13	4.16	1.25	11.16/-1.02	0.6	1.11	6.85/-1.66	0.9	1.83	9.46/-19.63
ion Error @10s	0.43	0.69	6.64/-4.91	6.06	3.3	4.33/-9.65	0.65	1.16	6.87/-2.4	0.87	1.91	20/-17.43

TABLE X
PREDICTION PERFORMANCE FOR ARTEMIS CYCLE (NEW CYCLE), DRIVER C

Model Type Error Type	IMPC-based Approach			Typical MPC based Approach			Intelligent Driver Model			Neural Network		
	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error
Predicted @1s	1.78	2.32	7.94/-7.65	0.25	0.7	4.13/-5.99	0.85	1.74	5.26/-20	1.67	2.64	14.9/-11.37
Speed @5s	2.34	3.00	10.97/-9.84	11.37	5.09	20/-7.4	2.71	3.74	15.44/-20	3.57	4.97	20/-20
Error @10s	2.96	3.88	16.43/-12.3	14.65	16.99	20/-20	3.47	4.79	17.5/-20	4.22	6.29	20/-20
Predicted @1s	1.12	1.49	5.81/-5.25	0.68	1.15	6.49/-1.86	0.87	1.81	6.79/-20	1.66	2.65	15.85/-13.8
Accelerat @5s	0.69	1.19	6.56/-4.97	2.63	2.34	11.13/-3.23	0.75	1.38	6.99/-2.69	1.87	3.78	20/-20
ion Error @10s	0.76	1.28	6.65/-5.32	4.14	4.54	10.7/-10.38	0.83	1.51	7.17/-2.96	1.97	4.04	20/-20

TABLE XI
PREDICTION PERFORMANCE FOR NYCC CYCLE (NEW CYCLE), DRIVER C

Model Type Error Type	IMPC-based Approach			Typical MPC based Approach			Intelligent Driver Model			Neural Network		
	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error	Avg. Error	Std. Error	Max. Error
Predicted @1s	2.17	2.59	5.6/-7.48	0.84	1.33	2.29/-5.09	0.91	1.37	6.62/-2.15	4.85	6.24	20/-10.16
Speed @5s	1.62	2.07	8.57/-6.99	10.44	5.8	20/-6.77	3.37	3.9	14.41/-9.72	11.55	10.22	20/-20
Error @10s	2.73	3.68	12.62/-12.5	20	10.44	20/-16.75	3.71	4.7	16/-15.35	15.87	12.48	20/-20
Predicted @1s	1.72	2.18	5.8/-4.96	1.15	1.46	4.97/-1.64	2.49	3.03	6.79/-2.1	4.31	5.87	20/-10.15
Accelerat @5s	1.27	1.6	6.4/-4.37	4.3	2.26	9.71/-1.76	2.57	3.1	6.93/-2.52	7.29	9.36	20/-20
ion Error @10s	1.39	1.76	6.69/-4.79	1.79	2.13	8.6/-5.3	2.6	3.09	6.98/-4.11	6.88	9.16	20/-20

trained to reproduce state trajectories under HWFET cycle. However, the IMPC is still showing noticeable advantages in average speed and acceleration prediction accuracy at 10s prediction horizon compared to other predictors thanks to our proposed cost function evaluation method, especially for driver A and C. Moreover, the IMPC is resulting in smaller standard deviation for prediction errors, as well as smaller maximum errors, which further proves the effectiveness of our proposed IMPC predictor formulation.

2) Comparison of Prediction Scalability

TABLE IV, TABLE VII and TABLE X show the prediction results under Artemis cycle, TABLE V, TABLE VIII and TABLE XI show the results obtained from NYCC cycle. Such results provide a comparison in not just prediction accuracy, but also prediction scalability.

Under Artemis cycle, the IDM and NN models' lack in scalability starts to appear. They fall behind IMPC by quite a lot in both speed and acceleration prediction accuracy, and the difference increases as the prediction horizon extends. This indicates that IMPC based approaches can catch the internal stimulus of human actions and perform better in unseen situations. One interesting finding is that the IDM performs better in speed prediction accuracy at 1s prediction horizon for driver C. That is possibly caused by driver C's driving style. During driving, driver C responded to the lead vehicle in an aggressive and delayed manner. Such driving style may cause the driver's behavior to go against common control logic. The IMPC could pick up driver C's preference from heavy acceleration and braking behaviors and generate opposite predictions when the driver is not responding timely. On the

other hand, since IDM is inherently a conservative collision-free model, it may not try as hard as the IMPC models to keep up with the lead vehicle within the predicting horizon, resulting in a lower prediction error when driver C is not driving very well. However, such effect is less obvious when the prediction horizon is extended to 5s or 10s. That is because as the prediction time goes by, driver C's unexpected driving behaviors will be less influential than his overall driving preferences.

Under NYCC cycle, the general observations are like those from Artemis cycle. The IDM and NN are performing much worse than IMPC approaches under NYCC cycle, and the gap increases as the prediction horizon extends. One exception is speed prediction accuracy at 1s horizon for all three drivers. The IMPC does not work well at 1s prediction horizon under NYCC cycle compared to IDM. For driver A, that is probably because of the presence of v_h in the cost function. Since the speed in NYCC cycle is much lower than that in HWFET cycle, the reference value for v_h might be too high and dominate the total cost value at low-speed range. The MPC might predict the vehicle to accelerate heavier than it really does. For driver B and C, although there is only v_r in the cost function, the relative value of speed and TTCi tracking error in the total cost will decrease when the overall speed range is low. The MPC might predict the vehicle to move in a more aggressive way under the increased influence from acceleration and control input tracking error. Such problem can be addressed by employing more comprehensive cost function design in future work. None the less, the IMPC approach surpasses other methods at 5s and 10s prediction horizons. It also needs to be noticed that the

acceleration prediction of NN is significantly worse than the other 3 predictors. That is because the training data set is not able to provide enough information to get the NN trained properly since these two driving cycles are almost entirely different from each other. The NN has basically lost predicting capability at all under NYCC cycle.

From these results, we can see that in general the IMPC based approach outperforms the other two approaches by providing a much higher prediction accuracy, especially at long prediction horizons, and a much better scalability. It can adapt to never-seen situations better than other approaches. The proposed cost function selection process can help our proposed IMPC approach win over typical MPC approach by better capturing the human driver's driving intentions and further improving the prediction accuracy.

V. CONCLUSION

In this paper a new IMPC based approach is proposed to model and predict the longitudinal behaviors of human-driven vehicles. A new cost function selection process is also proposed to determine the appropriate cost function in IMPC. The proposed approach can capture the internal control process of humans and thus result in better accuracy and scalability which is validated by the experimental results. The capability of predicting a human-driven vehicle's longitudinal states is tested on different drivers under different driving scenarios, and the performance is compared with existing approaches. The results illustrate the effectiveness and advantages of the proposed approaches in predicting the forthcoming behaviors/states and handling unseen situations compared with other existing approaches. The human-in-the-loop experiments have demonstrated the benefits of the proposed approach. The IMPC based predictor can reduce the headway tracking error while improving the riding comfort and fuel efficiency of the following autonomous vehicle at the same time.

As for future work, we plan to extend the proposed framework to the prediction of other behaviors/states of human-driving vehicles such as lane tracking and lane switching in addition to the studied longitudinal driving behaviors. We are also planning to apply the approach to studying how humans gain their driving styles and how such styles evolve as the driving experience increases.

ACKNOWLEDGMENT

This work was partially supported by the National Science Foundation under Grants CNS-1755771 and IIS-1845779.

REFERENCES

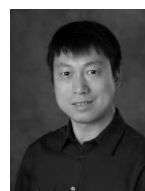
- [1] Brenner, Walter, and Andreas Herrmann. "An overview of technology, benefits and impact of automated and autonomous driving on the automotive industry." *Digital marketplaces unleashed*. Springer, Berlin, Heidelberg, 2018. 427-442.
- [2] Chan, Ching-Yao. "Advancements, prospects, and impacts of automated driving systems." *International journal of transportation science and technology* 6.3 (2017): 208-216.
- [3] Milakis D, Van Arem B, Van Wee B. Policy and society related implications of automated driving: A review of literature and directions for future research[J]. *Journal of Intelligent Transportation Systems*, 2017, 21(4): 324-348.
- [4] Tampère, Chris MJ. Human-kinetic multiclass traffic flow theory and modelling. With application to Advanced Driver Assistance Systems in congestion. Diss. 2004.
- [5] Bando, Masako, et al. "Dynamical model of traffic congestion and numerical simulation." *Physical review E* 51.2 (1995): 1035.
- [6] Kesting, Arne, and Martin Treiber. "Calibrating car-following models by using trajectory data: Methodological study." *Transportation Research Record: Journal of the Transportation Research Board* 2088 (2008): 148-156.
- [7] Schnelle, Scott, et al. "A feedforward and feedback integrated lateral and longitudinal driver model for personalized advanced driver assistance systems." *Mechatronics* 50 (2018): 177-188.
- [8] Bonsall, Peter, Ronghui Liu, and William Young. "Modelling safety-related driving behaviour—impact of parameter values." *Transportation Research Part A: Policy and Practice* 39.5 (2005): 425-444.
- [9] B., Drew, L. Guo, and Y. Jia. "Modeling and Characterization of Driving Styles for Adaptive Cruise Control in Personalized Autonomous Vehicles." *ASME 2017 Dynamic Systems and Control Conference*. American Society of Mechanical Engineers, 2017.
- [10] Panwai, Sakda, and Hussein Dia. "Neural agent car-following models." *IEEE Transactions on Intelligent Transportation Systems* 8.1 (2007): 60-70.
- [11] Khodayari, Alireza, et al. "A modified car-following model based on a neural network model of the human driver effects." *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 42.6 (2012): 1440-1449.
- [12] Morton, Jeremy, Tim A. Wheeler, and Mykel J. Kochenderfer. "Analysis of recurrent neural networks for probabilistic modeling of driver behavior." *IEEE Transactions on Intelligent Transportation Systems* 18.5 (2017): 1289-1298.
- [13] Gindele, Tobias, Sebastian Brechtel, and Rüdiger Dillmann. "A probabilistic model for estimating driver behaviors and vehicle trajectories in traffic environments." *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010.
- [14] Kumagai, Toru, and Motoyuki Akamatsu. "Prediction of human driving behavior using dynamic Bayesian networks." *IEICE TRANSACTIONS on Information and Systems* 89.2 (2006): 857-860.
- [15] Angkitrakul, Pongtep, Chiyomi Miyajima, and Kazuya Takeda. "Modeling and adaptation of stochastic driver-behavior model with application to car following." *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011.
- [16] Chernova, Sonia, and Manuela Veloso. "Confidence-based policy learning from demonstration using gaussian mixture models." *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM, 2007.
- [17] Lefevre, Stéphanie, et al. "Lane keeping assistance with learning-based driver model and model predictive control." *12th International Symposium on Advanced Vehicle Control*. 2014.
- [18] Qu, Ting, et al. "Modeling driver's car-following behavior based on hidden Markov model and model predictive control: A cyber-physical system approach." *2017 11th Asian Control Conference (ASCC)*. IEEE, 2017.
- [19] Hermes, Christoph, et al. "Long-term vehicle motion prediction." *2009 IEEE intelligent vehicles symposium*. IEEE, 2009.
- [20] Kouvaritakis, Basil, and Mark Cannon. "Model predictive control." Switzerland: Springer International Publishing (2016).
- [21] Liu, Jiechao, et al. "Combined speed and steering control in high-speed autonomous ground vehicles for obstacle avoidance using model predictive control." *IEEE Transactions on Vehicular Technology* 66.10 (2017): 8746-8763.
- [22] Mainprice, Jim, Rafi Hayne, and Dmitry Berenson. "Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces." *IEEE Transactions on Robotics* 32.4 (2016): 897-908.
- [23] Kalakrishnan, Mrinal, et al. "Learning objective functions for manipulation." *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013.
- [24] Ulusoy, Lütfi, Müjde Güzelkaya, and İbrahim Eksin. "Inverse optimal control approach to model predictive control for linear system models." *2017 10th International Conference on Electrical and Electronics Engineering (ELECO)*. IEEE, 2017.
- [25] Abbeel, Pieter, and Andrew Y. Ng. "Apprenticeship learning via inverse reinforcement learning." *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004.

- [26] Ziebart, Brian D., et al. "Maximum entropy inverse reinforcement learning." (2008).
- [27] Mombaur, Katja, Anh Truong, and Jean-Paul Laumond. "From human to humanoid locomotion—an inverse optimal control approach." *Autonomous robots* 28.3 (2010): 369-38.
- [28] Houska, Boris, Hans Joachim Ferreau, and Moritz Diehl. "ACADO toolkit—An open-source framework for automatic control and dynamic optimization." *Optimal Control Applications and Methods* 32.3 (2011): 298-312.
- [29] Guo, Longxiang, and Yunyi Jia. "Modeling, Learning and Prediction of Longitudinal Behaviors of Human-Driven Vehicles by Incorporating Internal Human DecisionMaking Process using Inverse Model Predictive Control." 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2019.
- [30] Nelder, John A., and Roger Mead. "A simplex method for function minimization." *The computer journal* 7.4 (1965): 308-313.
- [31] Li, Shengbo, et al. "Model predictive multi-objective vehicular adaptive cruise control." *IEEE Transactions on Control Systems Technology* 19.3 (2010): 556-566.
- [32] Lefèvre, Stéphanie, et al. "Comparison of parametric and non-parametric approaches for vehicle speed prediction." 2014 American Control Conference. IEEE, 2014.
- [33] Kesting, Arne, Martin Treiber, and Dirk Helbing. "Enhanced intelligent driver model to access the impact of driving strategies on traffic capacity." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368.1928 (2010): 4585-4605.
- [34] Lv, Chen, et al. "Levenberg–Marquardt Backpropagation Training of Multilayer Neural Networks for State Estimation of a Safety-Critical Cyber-Physical System." *IEEE Transactions on Industrial Cyber-Physical System*.
- [35] Dollar, Robert Austin, and Ardalan Vahidi. "Efficient and Collision-Free Anticipative Cruise Control in Randomly Mixed Strings." *IEEE Transactions on Intelligent Vehicles* 3.4 (2018): 439-452.



Longxiang Guo (S'17) received the B.S. degree in mechanical engineering from Tsinghua University, Beijing, China, in 2012, and the M.S. degree in mechanical engineering from Chinese Academy of Sciences, Shenzhen, China, in 2015. Currently, he is working toward

the Ph.D. degree in automotive engineering at Clemson University International Center for Automotive Research (CU-ICAR).



Yunyi Jia (S'08-M'15-SM'20) received the B.S. degree in automation from National University of Defense Technology, Changsha, China, in 2005, the M.S. degree in control theory and control engineering from South China University of Technology, Guangzhou, China, in 2008, and the Ph.D. degree in

electrical engineering from Michigan State University, Michigan, USA, in 2014.

He is currently the director of Collaborative Robotics and Automation Lab and an assistant professor in the Department of Automotive Engineering at Clemson University in Greenville SC USA. His research interests include robotics, automated vehicles and advanced sensing.