Stochastic Model Predictive Control for Coordination of Autonomous and Human-driven Vehicles

Sanzida Hossain * **Jiaxing Lu** ** **He Bai** * **Weihua Sheng** **

* *Mechanical and Aerospace Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: {sanzida.hossain, he.bai}@okstate.edu)*
** *Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078 USA (e-mail: {jiaxing.lu,weihua.sheng}@okstate.edu)*

Abstract: In this paper, we investigate coordination of an autonomous vehicle (AV) and an intelligent human vehicle (IHV). The IHV is a human-driven vehicle that can communicate and collaborate with other vehicles while also providing advisory directives to the driver to optimize its maneuver. The objective is to optimize control inputs for the AV and advisory directives for the driver on the IHV to coordinate their motions. We consider a coordinated lane merging example where the two vehicles need to reach a prescribed separation before the lane merging maneuver. We model the motion of the IHV and the AV using a Discrete Hybrid Stochastic Automata (DHSA) and formulate a model predictive control (MPC) problem to generate optimal inputs to the two vehicles. In particular, the input to the IHV is advisory commands that stochastically transition the human state. Since solving the MPC involves mixed-integer programming (MIP), we leverage a machine learning approach to predict optimal integer values, thereby reducing the computational time of the optimization. Preliminary simulation results and experimental findings from a driving simulator reveal successful coordination between the IHV and the AV and enhanced merging performance when compared to the 'no advising' scenario.

Keywords: Model predictive control, cooperative driving, mixed integer programming, hybrid system

1. INTRODUCTION

According to Statista (2020), the number of AVs in operation in the United States will be more than 2.1 million by 2025 and 20.8 million by 2030. The phenomenon of AVs rapidly increasing inevitably exists as a result of the development of technologies in several areas, such as artificial intelligence, machine learning, and automatic control. Although the number of AVs is expected to significantly increase, the public will inevitably have doubt about AVs' safety and vulnerability. Both human-driven vehicles and AVs are expected to co-exist in the next few decades. Therefore, utilizing cooperative driving between human-driven and AVs have the potential to make transportation more efficient and safe in the near future. According to Kim et al. (2015), installing autonomous cars with communication networks considerably improves their performance. As a result, transportation system efficiency also improves.

Researchers have been interested in cooperative driving for the past decade. There are a few studies that include a human-driven vehicle in the coordinating process. Chiang et al. (2010) describes a longitudinal automation system using human-in-the-loop technologies. Their proposed system has a hierarchical structure consisting of an adaptive sensory processor, a supervisory control and a regulation control. Lam et al. (2015) models the interaction between the driver and the vehicle in an assistive driving system using hidden mode stochastic hybrid systems. They also demonstrate that by monitoring both human behavior and vehicle status, the human state may be inferred, improving decision-making quality. Lam and Sastry (2014) presents how the partially observable Markov decision process (POMDP) can be used as a unified framework for the human model, the machine dynamic model and the observation model in a human-in-the-loop control. Driggs-Campbell et al. (2015) presents a testbed for gathering realistic driving data while retaining safety and control of the environmental surroundings. They demonstrate that the realistic data may be used to construct a precise and accurate driver model matched to an individual for semi-autonomous systems. Tian et al. (2021) proposes that robots use confidence-aware game-theoretic models of human behavior when assessing the safety of human-robot interaction. In all of these references, the human driving behavior is predicted and optimized but AVs are not considered together with the human-driven vehicle in a connected environment.

In this paper, we investigate cooperative driving between an IHV and an AV in a connected environment. In particular, we consider vehicle coordination for lane merging, where the two vehicles need to achieve a certain safe distance before merging. The objective is to design control inputs for the AV and advisory commands for the driver on the IHV and coordinate their behaviors. As the IHV is controlled by a human driver, its dynamics involve continuous variables such as position and velocity of the vehicle, discrete variables such as human states, and stochastic transitions of the human states. We employ DHSA Bemporad and Di Cairano (2010) to model the motion of the IHV into an MPC problem to optimize the AV's input and the IHV's advisory commands. Because of the discrete variables in the DHSA, solving the MPC problem inherently requires mixed-integer programming (MIP). To speed

⋆ This work is supported by the National Science Foundation (NSF) Grants CPS 2212582 and CISE/IIS 1910933.

up the computation of MIP, motivated by Bertsimas and Stellato (2022) we train a neural network to predict optimal integer values for the MPC, thereby reducing the MIP into non-integer programming that can be solved faster. We conduct human-in-the-loop experiments on a driving simulator to examine the effectiveness of our approach. The preliminary findings show that with the advisory commands, the two vehicles can reach the prescribed separation faster.

The rest of this paper is organized as follows. The DHSA and the system modeling approach are discussed in Section 2. An MPC problem formulation for coordinated lane merging is presented in Section 3. The implementation process of using a NN to speed up the optimization speed is given in Section 4. In Section 5 we present our simulation results and demonstrate the speed up using the NN prediction. Preliminary experimental findings acquired from a driving simulation testbed using our optimization approach are discussed in Section 6. Conclusions and future work are discussed in Section 7.

2. SYSTEM MODELING

2.1 Review of DHSA

DHSA is a mathematical formulation for a family of hybrid dynamical systems. In a DHSA, the uncertainty appears on the discrete component of the hybrid dynamics in the form of stochastic events, which, together with deterministic events, define the discrete state transition. A DHSA consists of four components: the switched affine system (SAS), the event generator (EG), the mode selector (MS), and the stochastic finite state machine (sFSM) Bemporad and Di Cairano (2010).

The SAS is defined by the linear difference equations

$$x_c(k+1) = A_{i(k)}x_c(k) + B_{i(k)}u_c(k) + f_{i(k)} \tag{1}$$

where $k \in \mathbb{Z}_{0+} \triangleq \{0,1,...\}$ is the discrete time index, $x_c(k) \in \mathbb{R}^{n_c}$ is the vectors of continuous states, n_c is the number of continuous states, $u_c(k) \in \mathbb{R}^{m_c}$ is the vectors of continuous inputs, m_c is the number of continuous inputs and $i(k) \in I \triangleq \{1,2,\cdots,s\}$ is the current mode of the system. $\{A_i, B_i, f_i\}_{i\in I}$ are constant matrices of suitable dimensions that define the dynamics of the system in the current mode $i(k)$.

The EG produces binary event signals $\delta_e(k) \in \{0,1\}^{n_e}$ where n_e is the number of generated binary event signals. It is defined by

$$\delta_e(k) = f_{EG}(x_c(k), u_c(k)) \tag{2}$$

where $f_{EG} : \mathbb{R}^{n_c+m_c} \to \{0,1\}^{n_e}$ is the event generation function given by

$$[f^j_{EG}(x_c, u_c) = 1] \leftrightarrow [H^j_e x_c + J^j_e u_c + K^j_e \leq 0] \tag{3}$$

in which $H_e \in \mathbb{R}^{n_e\times n_c}$, $J_e \in \mathbb{R}^{n_e\times m_c}$, and $K_e \in \mathbb{R}^{n_e}$ are constant matrices defining linear threshold conditions. The superscript j denotes the j^{th} row.

The MS is defined by a Boolean function $f_{MS} : \{0,1\}^{n_b+m_b+n_e} \to I$ where n_b is the number of binary states and m_b is the number of binary inputs. It can be defined as

$$i(k) = f_{MS}(x_b(k), u_b(k), \delta_e(k)) \tag{4}$$

where $x_b \in \{0,1\}^{n_b}$ is the vector of binary states and $u_b \in \{0,1\}^{m_b}$ is the vector of binary inputs.

The sFSM is a Boolean function $f_{sFSM} : \{0,1\}^{2n_b+m_b+n_e} \to [0,1]$ satisfying

$$p[x_b(k+1) = \hat{x}_b] = f_{sFSM}(x_b(k), u_b(k), \delta_e(k), \hat{x}_b) \tag{5}$$

where, $p[\cdot]$ denotes the probability of a transition.

2.2 Application to vehicle coordination

In this paper, we consider coordination of an AV and an IHV. We assume that the AV can sense its surroundings, decide which path to take to its destination, and drive itself. Therefore, the dynamics of the AV and its autonomous inputs determine the vehicle's states. The IHV is a human-driven vehicle with an assisted driving device that can communicate with other vehicles through V2V communication and provide "advisory commands" to the human driver for coordinated driving. The driver has the option to follow or not follow the commands. Thus, the IHV is a hybrid system consisting of variables of both continuous and discrete nature. We adopt the DHSA formulation, where the discrete human states dictate the models of the vehicle states and actions and the transitions of human states are stochastic. The dynamics of the vehicle is modeled with a SAS and the stochastic human transitions are modeled with sFSM. The advisory commands enter the sFSM as a control input to stochastically transition the human states.

In the next section, we present the details of the DHSA formulation for coordinated lane merging between an AV and an IHV. The formulation can be extended to address more than two vehicles, which will be considered in our future work.

3. COORDINATION FOR LANE MERGING

We consider coordination for lane merging of one AV and one IHV. The goal is to reach d_r meters in the longitudinal distance between the two vehicles as fast as possible to allow a safe and efficient lane merging. While the motion of the AV can be directly controlled by the autonomous input, the motion of the IHV can be impacted only through the driver via advisory commands. To coordinate the motion of the two vehicles, we formulate a stochastic MPC problem with state and control constraints. The solution to the MPC problem provides the optimal advisory commands to the IHV and the optimal autonomous controls to the AV.

3.1 Dynamic models

We consider a linear state space model for the motion of an AV

$$x^r_{k+1} = A_r x^r_k + B_r u^r_k, \tag{6}$$

where the subscript $k \in \mathbb{Z}_+$ is the discrete-time index, the longitudinal position and velocity with respect to the origin are represented by $x^r_k \in \mathbb{R}^2$, A_r and B_r are matrices of suitable dimensions that define the AV dynamics, and $u^r \in \mathbb{R}$ is the input (acceleration) to the AV.

The IHV's motion is determined by whether the driver follows the advised commands. As a result, it switches between two dynamic systems based on a binary decision variable: not following the advisory command (0), which indicates that the IHV is under human control, and following the advisory command (1), which indicates that the IHV is under advisory control. We assume that the vehicle models are given by

$$\text{IHV under human control: } x^h_{k+1} = A_h x^h_k + B_h f^h_k \tag{7}$$

$$\text{IHV under advisory control: } x^h_{k+1} = A_h x^h_k + B_h u^a_k, \tag{8}$$

where A_h and B_h are matrices of suitable dimensions that define the IHV dynamics, and $f^h_k \in \mathbb{R}$ is the human input and $u^a_k \in \mathbb{R}$

is the advisory commands for the IHV. We assume that f_k^h is available through a driver monitoring system. For example, in Tran et al. (2018), we developed such a system based on vehicle data and video data of the driver.

Denote by $x_k^B \in \{0,1\}$ the event of the IHV following the advisory control (1) or not (0) and let $x_k = [(x_k^r)^T \ (x_k^h)^T]^T$. We also define

$$A_1 = A_2 = \begin{pmatrix} A_r & 0 \\ 0 & A_h \end{pmatrix}, \ B_1 = \begin{pmatrix} B_r & 0 \\ 0 & 0 \end{pmatrix}, \ B_2 = \begin{pmatrix} B_r & 0 \\ 0 & B_h \end{pmatrix},$$
$$f_1 = \begin{pmatrix} 0 \\ B_h f_k^h \end{pmatrix}, \ f_2 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{9}$$

Then (6)–(8) can be rewritten in the form of (1)

$$x_{k+1} = A_{i_k} x_k + B_{i_k} \begin{pmatrix} u_k^r \\ u_k^a \end{pmatrix} + f_{i_k} \tag{10}$$

where the MS is given by

$$i_k = x_k^B + 1. \tag{11}$$

Note that the MS is independent of an event generator.

An important step in the MPC formulation in Bemporad and Di Cairano (2010) is to write (10) into a single linear system using mixed-logic dynamical (MLD) systems Bemporad and Morari (1999). Towards this end, we further define $z_k^u = x_k^B(u_k^a - f_k^h)$ and rewrite (10) as

$$x_{k+1} = \begin{pmatrix} A_r & 0 \\ 0 & A_h \end{pmatrix} x_k + \begin{pmatrix} B_r & 0 \\ 0 & B_h \end{pmatrix} \begin{pmatrix} u_k^r \\ z_k^u \end{pmatrix} + \begin{pmatrix} 0 \\ B_h f_k^h \end{pmatrix}. \tag{12}$$

One can obtain the solution to x_k based on u_k^r, z_k^u, and f_k^h and the initial conditions x_0^r and x_0^h as

$$x_k^r = A_r^k x_0^r + \sum_{j=0}^{k-1} A_r^{k-j-1} B_r u_j^r, \tag{13}$$

$$x_k^h = A_h^k x_0^h + \sum_{j=0}^{k-1} A_h^{k-j-1} B_h f_j^h + \sum_{j=0}^{k-1} A_h^{k-j-1} B_h z_j^u. \tag{14}$$

The equality $z_k^u = x_k^B(u_k^a - f_k^h)$ will be enforced as a constraint in the MPC formulation.

3.2 Stochastic human state model

We model the stochastic transitions of the binary human state x_k^B by an sFSM. Let $u_k^B \in \{0,1\}$ denote the on/off of an advisory control at time step k. Based on the first-order Markov assumption, we prescribe the transition probability of x_{k+1}^B given x_k^B and u_k^B. Therefore, there are 8 different possibilities for transitions. Following Bemporad and Di Cairano (2010), we introduce $w^i \in \{0,1\}$ (an uncontrollable event) for each transition and constrain $w^i = 1$ if and only if the ith transition occurs.

The probability of each event $p[w_k^i]$ needs to be specified. In particular, we let $p[w_k^2] = p_t$ which is the probability of transitioning to an advisory action. We also let $p[w_k^6] = p_f$ which is the probability of continuously following the advisory control. We set $p[w_k^3] = 1$, $p[w_k^4] = 0$, $p[w_k^7] = 1$, $p[w_k^8] = 0$. Note that $p[w_k^i] + p[w_k^{i+1}] = 1$, $i = 1,3,5,7$. The p_t and p_f may be learned from human-in-the-loop experiments. The transition of x_k^B and the events w_k^i are illustrated in Fig. 1.

Let $C = [-1\ 0\ -1\ 0]$. One can verify that a linear model describing the transitions of the sFSM is given by

$$x_{k+1}^B = u_k^B + C[w_k^1 \ w_k^2 \ w_k^5 \ w_k^6]^\top \tag{15}$$

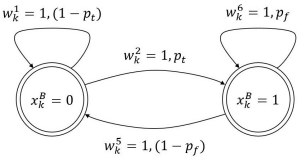

Fig. 1. Stochastic finite state machine for human state transition.

which leads to

$$x_k^B = \sum_{j=0}^{k-1} u_j^B + C \sum_{j=0}^{k-1} w_j, \forall k \geq 1. \tag{16}$$

3.3 Optimization constraints

There are five sets of constraints considered in our MPC formulation. Denote by M_u and m_u the upper and lower limit of the input acceleration for the IHV, respectively.

First set of constraints: The equality $z_k^u = x_k^B(u_k^a - f_k^h)$ together with the upper and lower bounds M_u and m_u leads to

$$z_k^u \leq (M_u - f_k^h)x_k^B, \ z_k^u \geq (m_u - f_k^h)x_k^B \tag{17}$$
$$z_k^u \leq (u_k^a - f_k^h) - (m_u - f_k^h)(1 - x_k^B) \tag{18}$$
$$z_k^u \geq (u_k^a - f_k^h) - (M_u - f_k^h)(1 - x_k^B). \tag{19}$$

Second set of constraints: The state transitions defined in the sFSM are enforced in these constraints. Let $\delta_k^1 = x_k^B u_k^B$. Then the transitions in Fig1 produce the following inequalities

$$w_k^1 + w_k^2 \leq (1 - x_k^B)u_k^B = u_k^B - \delta_k^1, \quad w_k^1 + w_k^2 \geq u_k^B - \delta_k^1, \tag{20}$$
$$w_k^5 + w_k^6 \leq x_k^B u_k^B = \delta_k^1, \quad w_k^5 + w_k^6 \geq \delta_k^1, \tag{21}$$
$$-x_k^B + \delta_k^1 \leq 0, \quad -u_k^B + \delta_k^1 \leq 0, \quad u_k^B + x_k^B \leq 1 + \delta_k^1. \tag{22}$$

Third set of constraints: For collision avoidance during lane merging, the longitudinal position between the two vehicles needs to be larger than a threshold $d_r > 0$, i.e.,

$$|x_{k,1}^r - x_{k,1}^h| \geq d_r \tag{23}$$

where $x_{k,1}$ denotes the position state. We introduce two binary variables $b_{1,k}$ and $b_{2,k}$ and two inequalities

$$x_{k,1}^r - x_{k,1}^h \leq -d_r + \bar{M} b_{1,k} \tag{24}$$
$$x_{k,1}^r - x_{k,1}^h \geq d_r - \bar{M} b_{2,k} \tag{25}$$

with a sufficiently large $\bar{M}$. When $b_{1,k} = 0$ and $b_{2,k} = 1$, (24) becomes $x_{k,1}^r - x_{k,1}^h \leq -d_r$ and (25) becomes $x_{k,1}^r - x_{k,1}^h \geq d_r - \bar{M}$, which holds trivially. Similarly, when $b_{1,k} = 1$ and $b_{2,k} = 0$, (25) becomes $x_{k,1}^r - x_{k,1}^h \geq d_r$ and (24) becomes $x_{k,1}^r - x_{k,1}^h \leq -d_r + \bar{M}$, which holds trivially. Thus, $b_{1,k} + b_{2,k} = 1$ ensures that (23) is satisfied at time step k.

To reduce the time to reach the condition in (23), we introduce a constraint

$$b_{1,k} + b_{2,k} \geq 1 \tag{26}$$

and minimize $b_{1,k} + b_{2,k}$ (among other objectives) in the objective function.

Fourth set of constraints: All the limits on the states of the IHV and AV are enforced using these set of constraints, i.e.,

$$x_k^r \leq M, \ x_k^r \geq m \tag{27}$$
$$x_k^h \leq M, \ x_k^h \geq m \tag{28}$$

for some M and m.

Fifth set of Constraints: Chance constraints are used to reject trajectories that only occur infrequently from the set of possible solutions. In our DHSA formulation, the possible transition events are given by $w_k = \begin{bmatrix} w_k^1 & w_k^2 & w_k^5 & w_k^6 \end{bmatrix}^\top$ and the transition probabilities are $p = \begin{bmatrix} p[w_k^1] & p[w_k^2] & p[w_k^5] & p[w_k^6] \end{bmatrix}^\top$. Following Bemporad and Di Cairano (2010), the probability of the state trajectory can be computed as

$$\begin{bmatrix} \pi_0 \\ \vdots \\ \pi_{K-1} \end{bmatrix} = \begin{bmatrix} w_0^\top \\ \vdots \\ w_{K-1}^\top \end{bmatrix} p \tag{29}$$

where K is the look ahead window in the MPC. At step k, the π_k indicates the likelihood of taking the transition described by w_k. The probability of the whole w trajectory, $\pi(w)$, is given by

$$\pi(w) = \pi(w_0, w_1, \ldots, w_k) = \prod_{k=0}^{K-1} \pi_k. \tag{30}$$

Then the chance constraint can be defined as

$$\pi(w) \geq \tilde{p} \tag{31}$$

with $\tilde{p} \in [0,1]$ being a probability bound. This chance constraint (31) enforces that w realizes with at least $\tilde{p}$ probability. From (31), for our system, we can compute the constraint as

$$\ln \pi(w) = \sum_{k=0}^{K-1} \sum_{i=1,2,5,6} w_k^i \ln(p[w_k^i]) \geq \ln(\tilde{p}). \tag{32}$$

3.4 The MPC formulation

Let K be the length of the look-ahead window in the MPC. At each time step k, the decision variables consist of continuous and binary variables, summarized as

$$\theta_k = \begin{bmatrix} \mathbf{u}_k^r & \mathbf{u}_k^a & \mathbf{z}_k^u & \mathbf{u}_k^B & \mathbf{w}_k & \delta_k^1 & \mathbf{b}_k \end{bmatrix} \tag{33}$$

where $\mathbf{u}_k^r = [u_k^r,\ u_{k+1}^r,\ \cdots,\ u_{k+K-1}^r]$ and $\mathbf{u}_k^a$, $\mathbf{z}_k^u$, $\mathbf{u}_k^B$, $\mathbf{w}_k$, δ_k^1 and $\mathbf{b}_k$ are defined similarly to $\mathbf{u}_k^r$. The continuous variables are $\mathbf{u}_k^r$, $\mathbf{u}_k^a$, and $\mathbf{z}_k^u$ while the other variables are binary.

The objective function of the MPC is designed to be the weighted sum of five quadratic or linear functions of θ_k, each with a different objective discussed below.

- Function 1 minimizes the energy of the control inputs to the AV and the IHV based on their respective weights. Function 1 is a quadratic function of θ_k;
- Function 2 maximizes the speed of both vehicles within a speed limit for fast lane merging. Function 2 is a linear function of θ_k;
- Function 3 minimizes the number of advisory actions so that the merge can happen with reduced advisory actions. Function 3 is a linear function of θ_k;
- Function 4 maximizes the probability of the stochastic events so that at least one of the four events occurs in a time instance. Function 4 is a linear function of θ_k;
- Function 5 minimizes $b_{1,k} + b_{2,k}$ so that the distance d_r can be reached quickly. Function 5 is a linear function of θ_k.

The objective function of the MPC is the sum of the aforementioned five functions, which can be represented as

$$J(\theta_k) = \theta_k^\top Q \theta_k + c^\top \theta_k \tag{34}$$

where $Q \in \mathbb{R}^{n_t \times n_t}$ and $c \in \mathbb{R}^{1 \times n_t}$ are the designed objective weights for the system. n_t is the total number of decision variables. The constraints of the MPC are (13), (14), (16) to (22), (24) to (28), (32), which are linear in θ_k, and thus can be written as $\mathbf{G}_k \theta_k \leq \mathbf{g}_k$. Then the MPC is given by

$$\min_{\theta_k} \ J(\theta_k), \quad s.t. \ \mathbf{G}_k \theta_k \leq \mathbf{g}_k. \tag{35}$$

Applying this optimization in a receding horizon fashion, we obtain the MPC algorithm where the optimal control inputs for each step k are applied to the AV and the advisory commands are conveyed to the IHV in each time step.

4. MACHINE LEARNING TO SPEED UP INTEGER OPTIMIZATION

The solutions to the MPC formulations can be obtained from a standard mixed-integer programming solver, e.g., the Gurobi optimizer Gurobi Optimization, LLC (2022). However, the computational time needed by the Gurobi solver may not be suitable for online implementation or scalable to a larger number of vehicles. We investigate a simplified version of the approach inspired by Bertsimas and Stellato (2022) to speed up the computation via machine learning and offline training.

We generate binary output results from Gurobi for different initial conditions and store them as training data sets for a neural network (NN). Once trained, the NN will predict the optimal integer values, thereby reducing the integer program to a non-integer program. The input parameters to the NN are defined as the initial speeds of the IHV sp_h and the AV sp_r, the position difference between the two vehicles dif, the probability of transitioning to advisory action p_t, and the initial human state x_0^B. For these parameters, the optimized binary outputs for $K = 30$ time steps are calculated. A total of 13,770 different data sets are collected as the training data.

For each data set, the optimal values of the binary variables are obtained from the Gurobi and recorded. Each unique combination of the optimal binary variables is called a strategy and it is given a strategy index. Calculating all different strategies, we obtain 28 unique strategies for our problem formulation. Then we denote the 28 strategies with a one-hot encoding where all the elements of a 28×1 vector are 0 except the one corresponding to the strategy index equal to 1. We use the one-hot encoded outputs as the target training data to train a classification NN model. Table 1 below lists the parameters used for the NN training model. Once trained, the NN predicts probable strategies among the 28 options, and the optimal integer values are extracted from the probable strategies.

Table 1. NN specifiations

Parameter	Used setting
Number of hidden layers	3
Width of hidden layers	50,150,150
Learning rate	0.001
Batch size	1000
Number of epochs	500
Inner layer activation function	ReLu
Output layer activation function	Softmax
Loss function	Binary cross-entropy

The trained NN model predicts the probability of a strategy to be the correct optimal strategy. The top 4 probable strategies are selected and compared. For each strategy, we optimize the cost function with the binary variables fixed to the values in that strategy, which simplifies the optimization to a convex optimization problem without any integers. The strategy with the lowest cost function is chosen as the optimal solution.

5. SIMULATION RESULTS

We implement the MPC method in simulations for the initial conditions as $sp_h = 15\ m/s$, $sp_r = 15\ m/s$, $dif = 0\ m$, and the initial Human state $x_0^B = 0$. The size of the look-ahead window was chosen as $K = 30$ and each time step is 0.8 seconds. We use a double integrator model for the vehicle dynamics. The simulations were run in Python on a computer with Intel(R) Core i7-3770 CPU @ 3.40GHz with 16GB RAM and NVIDIA GeForce GT 630 graphics card.

5.1 Simulated optimization results

This simulation is formulated following Section 3.4. Assuming that $p_t = 0.9$, i.e., the human starts to follow the advisory command after the first time step with a probability of 90%, and $p_f = 0.99$, i.e., the human will continue to follow the advisory command after the first time step with a probability of 99%, we obtain the results shown in Fig. 2(left). We can see that the merging action is distributed between the AV and the IHV for optimal merging. When we lower p_t and p_f to $p_t = 0.4$ and $p_f = 0.4$, we obtain the MPC results shown in Fig. 2(right). Note that for the optimal merging with high p_t

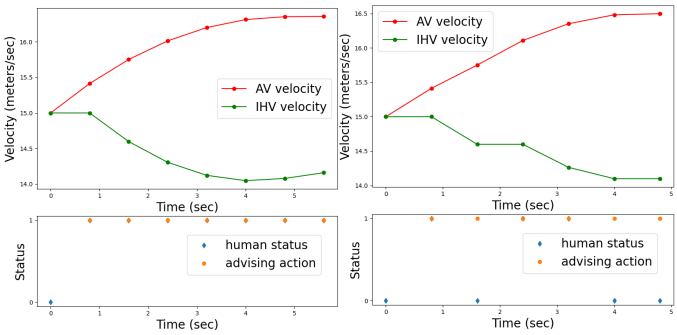

Fig. 2. Simulated optimization results for $p_t = 0.9$ and $p_f = 0.99$ (left column) for $p_t = 0.4$ and $p_f = 0.4$ (right column)

and p_f the merging action is distributed between the AV and the IHV. With a lower p_t and p_f, the optimized trajectory for the IHV is changed to accommodate for the human's probability of not following advisory commands. For different probabilities, different optimal trajectories are expected. If p_t is further decreased, no advising actions will be generated for the IHV and only the AV will be responsible for creating the separation. The average time for each step of calculation by Gurobi is 0.32 sec.

We conduct 20 Monte Carlo simulations with a varying p_t, $p_f = 0.99$, and all other initial conditions fixed. The average time for merge t_{avg}, the average distance traveled by IHV avd_{IHV} and AV avd_{AV}, and the average speed change by IHV avs_{IHV} and AV avs_{AV} are presented in Table 2. The t_{avg}, avd_{IHV}, and avd_{AV} decrease as p_t increases, which indicates improved efficiency. The avs_{IHV} and avs_{AV} reveal that the speedup and slowdown are more distributed among the AV and IHV with increasing p_t.

Table 2. Monte Carlo simulation results

p_t	t_{avg}	avd_{IHV}	avd_{AV}	avs_{IHV}	avs_{AV}
0.1	6.40	12.00	20.19	0.00	1.50
0.5	5.75	11.41	19.79	-0.71	1.42
0.9	5.55	11.29	19.98	-0.85	1.37

5.2 Speed-up by neural network

We report preliminary results on using NN to speed up the MIP in the MPC optimization. We consider the formulation in Section 5.1. The details of the NN training are discussed in Section 4. The results produced by the Gurobi and the NN-based optimization are compared against each other for 1050 test data points. For different initial conditions, the Gurobi optimization and the NN-based optimization results are evaluated.

- The average time to solve the MPC using Gurobi is 0.325 sec. The average time to solve the MPC using NN-based optimization (including NN inference) is 0.1524 sec, which means that the NN-based optimization is more than twice as fast as the Gurobi solver in this example.
- The accuracy of the NN prediction of the integer values is 99.7%. Even for the 0.3% wrong predictions, the MPC produces sub-optimal results satisfying all the constraints.

6. EXPERIMENTAL RESULTS

Multiple experiments were conducted to validate and demonstrate the effectiveness of our proposed MPC algorithm.

A Carnetsoft simulator Carnetsoft Inc. (2016) is adopted as the simulation testbed, in which data of an ego vehicle that a human is driving through the steering wheel, pedals, and shifter is collected, as shown in Fig. 3, and AVs can be implemented and controlled. Lane-merging scenarios are established based on this simulation testbed. In Fig. 3, the copilot attached to the steering wheel plate consists of a screen, a speaker and a Raspberry Pi 4B, which is responsible for the data communication and processing between the simulator and the Gurobi server. Two vehicles are involved in the merging scenarios. The ego vehicle is driven by a human subject and its data including velocity and location are obtained every 0.8 sec in real-time. The black AV on the right starts at the same speed as the human-driven vehicle and tries to merge into the human-driven vehicle's lane at a random time point. During the merging process, the commands for the AV are generated by the MPC algorithm and implemented in the simulator in real-time. We conducted

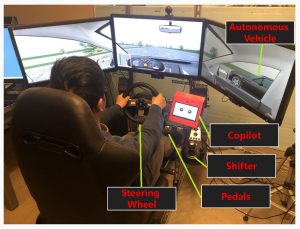

Fig. 3. The simulation testbed.

2 sets of experiments. The first set was with the advisory command 'Off' and the second set was with the advisory command 'On'. The goal of these experiments was to assess how the merging performs with and without advisory commands to the human driver. Each set of studies was repeated twice with 5 different drivers, totaling 10 experiments per set. Out of these 10 experimental data 4 merging tests from 4 different drivers are shown in Fig. 4 and Fig. 5, respectively.

For the test, we instructed the human driver to drive with the AV. For the 'On' scenarios, when the AV was within the merging range, the IHV was sent the advising command through speakers to speed up, slow down or maintain the speed based

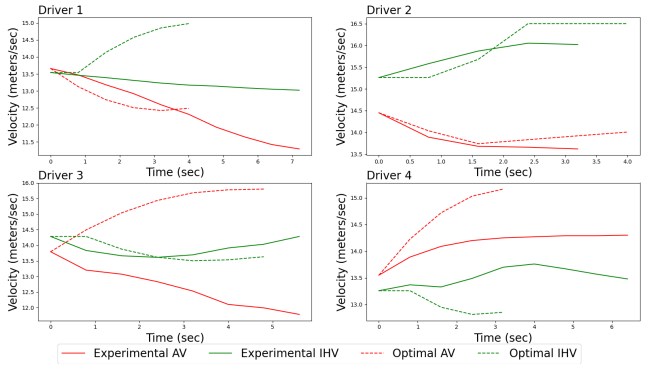

Fig. 4. Experimental velocity profiles for AV and IHV with advisory command 'Off' for different drivers.

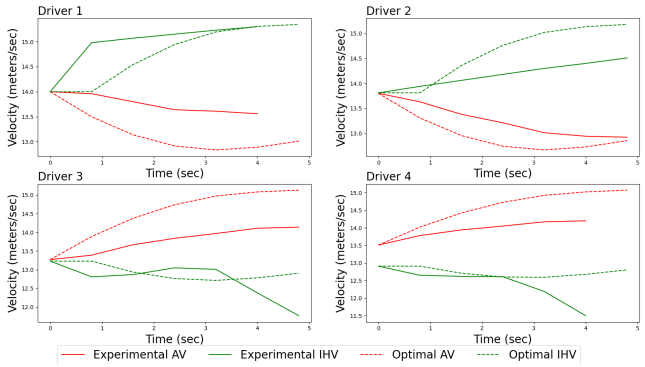

Fig. 5. Experimental velocity profiles for AV and IHV with advisory command 'On' for different drivers

on the optimization. The AV was sent the input command based on the MPC formulation. Because of the MPC formulation, the AV automatically changes its speed based on the IHV speed and behavior. Note that in our current setup, the speakers can only announce 'speedup', 'slow down', or 'maintain speed' without specifying a magnitude of acceleration or deceleration.

The optimal velocity profiles from the MPC are also plotted with the experimental results in Fig. 4 and Fig. 5. These profiles are the trajectories that would be the optimal solution for the two vehicles if the human was following the advisory command. With the advisory command 'Off', the human's behavior is less predictable. For example, in the cases of Driver 1 and Driver 4, the driver's behavior was the opposite of the optimal behavior. Among all of the test data sets from the 5 drivers, the driver has only a 40% probability of moving towards the advisory command and attempting to execute the optimal action for merging. The average merging time is 5.52 sec and the standard deviation of the merging times is 2.16 sec.

With the advisory command 'On', the human ultimately attempts to follow the advisory command. Even though the velocity profile of the IHV doesn't follow the optimal MPC solution in Fig. 5, it is driving towards the optimal solution compared with Fig. 4 The human driver has a 100% probability of moving toward the advisory command and attempting to execute the optimal action required for merging in all of the test data sets from 5 different drivers with advisory command 'On'. The average merging time is 4.64 sec with a standard deviation of 1.59 sec, which is faster than with the advisory command 'Off'. We conjecture that with a better user interface, the human driver could follow the advised commands better.

7. CONCLUSIONS AND FUTURE WORK

We present a framework for modeling the motion of the IHV and the AV using a DHSA, and an MPC problem for coordinated merging to generate optimal inputs for the AV and optimal advisory commands for the IHV. The simulation results show that the MPC algorithm takes into account the stochastic transitions of the human states and provides optimal merging actions. Preliminary experiments from a driving simulator testbed show promising results for successful coordination between an IHV and an AV. Future work includes extending the formulation to more vehicles, developing a recognition model to estimate the human driver's intention state, and experimenting with more complex driving scenarios.

REFERENCES

Bemporad, A. and Di Cairano, S. (2010). Model-predictive control of discrete hybrid stochastic automata. *IEEE Transactions on Automatic Control*, 56(6), 1307–1321.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *IEEE Transactions on Automatic Control*, 35, 407–427.

Bertsimas, D. and Stellato, B. (2022). Online mixed-integer optimization in milliseconds. *INFORMS Journal on Computing*.

Carnetsoft Inc. (2016). Research Driving Simulator. URL `https://cs-driving-simulator.com/`.

Chiang, H.H., Wu, S.J., Perng, J.W., Wu, B.F., and Lee, T.T. (2010). The human-in-the-loop design approach to the longitudinal automation system for an intelligent vehicle. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 40(4), 708–720.

Driggs-Campbell, K., Shia, V., and Bajcsy, R. (2015). Improved driver modeling for human-in-the-loop vehicular control. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 1654–1661.

Gurobi Optimization, LLC (2022). Gurobi Optimizer Reference Manual. URL `https://www.gurobi.com`.

Kim, S.W., Liu, W., Ang, M.H., Frazzoli, E., and Rus, D. (2015). The impact of cooperative perception on decision making and planning of autonomous vehicles. *IEEE Intelligent Transportation Systems Magazine*, 7, 39–50.

Lam, C.P. and Sastry, S.S. (2014). A POMDP framework for human-in-the-loop system. In *53rd IEEE Conference on Decision and Control*, 6031–6036.

Lam, C.P., Yang, A.Y., Driggs-Campbell, K., Bajcsy, R., and Sastry, S.S. (2015). Improving human-in-the-loop decision making in multi-mode driver assistance systems using hidden mode stochastic hybrid systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5776–5783.

Statista (2020). Projected number of autonomous vehicles in operation in the United States in 2025 and 2030. URL `https://www.statista.com/statistics/750149`.

Tian, R., Sun, L., Bajcsy, A., Tomizuka, M., and Dragan, A.D. (2021). Safety assurances for human-robot interaction via confidence-aware game-theoretic human models. *ArXiv*, abs/2109.14700.

Tran, D., Du, J., Sheng, W., Osipychev, D., Sun, Y., and Bai, H. (2018). A human-vehicle collaborative driving framework for driver assistance. *IEEE Transactions on Intelligent Transportation Systems*, 20(9), 3470–3485.