International Journal of Control, Automation, and Systems 21(10) (2023) 3324-3335
http://dx.doi.org/10.1007/s12555-022-0509-6
ISSN:1598-6446 eISSN:2005-4092
http://www.springer.com/12555

Learning and Leveraging Conventions in the Design of Haptic Shared Control Paradigms for Steering a Ground Vehicle

Vahid Izadi and Amir H. Ghasemi*

Abstract: The main objective of this paper is to establish a framework to study the co-adaptation between humans and automation systems in a haptic shared control framework. We specifically used this framework to design control transfer strategies between humans and automation systems to resolve a conflict when co-steering a semi-automated ground vehicle. The proposed framework contains three main parts. First, we defined a modular structure to separate partner-specific strategies from task-dependent representations and use this structure to learn different co-adaption strategies. In this structure, we assume the human and automation steering commands can be determined by optimizing cost functions. For each agent, the costs are defined as a combination of a set of hand-coded features and vectors of weights. The hand-coded features can be selected to describe task-dependent representations. On the other hand, the weight distributions over these features can be used as a proxy to determine the partner-specific conventions. Second, to leverage the learned co-adaptation strategies, we developed a map connecting different strategies to the outputs of human-automation interactions by employing a collaborative-competitive game concept. Finally, using the map, we designed an adaptable automation system capable of co-adapting to human driver's strategies. Specifically, we designed an episode-based policy search using the deep deterministic policy gradients technique to determine the optimal weights vector distribution of automation's cost function. The simulation results demonstrate that the handover strategies designed based on co-adaption between human and automation systems can successfully resolve a conflict and improve the performance of the human automation teaming.

Keywords: Conventions and co-adaption, haptic shared control, human-robot interaction, semi-automated vehicles.

1. INTRODUCTION

Given that both humans and robots are subject to faults, the hand-off problem – how to exchange control between a human and robot— plays a critical role in ensuring the performance of a human-robot teaming [1,2]. Humans seamlessly resolve conflicts by co-adapting to each other through repeated interactions. A hypothesis behind the seamless human-human collaboration is that humans can adaptively form conventions [3]. A convention is defined as low-dimensional shared representations that capture the interaction and can change over time [3]. In other words, in a multi-agent repeated game context, there can be several equilibria, with some more preferable than others. The conventions narrow to a subset of these equilibria to which the team might more naturally gravitate. Conventions can encompass information such as team member roles, specific expertise, tacit knowledge, nomenclature and communication, and other dimensions [4,5]. Forming conventions in humans-robots teams is tricky because the human partner is a non-stationary agent meaning that each human partner may gravitate to a different equilibrium than the other human partner. Also, these equilibria may change over time for a human partner [6,7]. Therefore, to form conventions in humans-robots teams, two main questions shall be answered: 1) How to learn forms of conventions 2) How to influence the agents (driver and automation) to form a desirable convention?

To answer these questions, specific challenges shall be considered. For instance, the existing approaches for modeling human reasoning either suffer from tractability issues (e.g., theory of mind) or detect the emergence of cooperative behaviors qualitatively in an offline manner [8-11]. To leverage conventions, frameworks that separate partner-specific conventions from task-dependent representations shall be designed [3,12]. Algorithms developed within these frameworks are required to address uncertainty in human-automation dynamics, the spatiotemporal constraints imposed by the nature of the task (e.g., steering control), and the computational scalability for real-time implementation.

Furthermore, to adaptively form conventions, an au-

Manuscript received June 16 2022; revised January 31, 2023 and March 30, 2023; accepted April 3, 2023. Recommended by Associate Editor Yuhu Wu under the direction of Senior Editor Choon Ki Ahn.

Vahid Izadi and Amir H. Ghasemi are with the Department of Mechanical Engineering at the University of North Carlona at Charlotte, 9201 University Blvd, North Carolina 28223, USA (e-mails: izadi15092@gmail.com, ah.ghasemi@uncc.edu).
* Corresponding author.

tomation system should be able to learn complex policies automatically. While model-predictive-based approaches are powerful tools to deal with the uncertainty and complexity of human-automation teaming, they lack the learning capability [13-15]. On the other hand, traditional end-to-end learning algorithms require significant amounts of data (hundreds or even thousands of experiments) to achieve a desired level of performance that may not be feasible in the context of human-automation teaming problems. Therefore, for the automation system to effectively learn complex policies, algorithms shall be developed that leverage the advantages of both model-based and data-driven techniques [13,16].

This paper describes our solution to these challenges. In particular, we propose a framework wherein the human and automation actions are defined based on optimizing cost functions. Here, we model the human and automation cost function for driving a semi-automated vehicle (e.g., obstacle avoidance) as a weighted linear combination of a set of features that a human and automation care about (e.g., collision avoidance, staying on the road, or distance to the final goal). While these features represent the task, we argue that the distribution of the weights associated with these features and how they may evolve in time can be used as a proxy to learn and leverage the conventions formed between the human driver and the automation system. Additionally, by defining the concept of cooperative and competitive cost functions, we create a map to characterize human-automation interaction outputs under different conventions. Using such a map, an adaptable automation system is designed to adapt its behavior and form a desirable convention with a human driver. Specifically, we implement a deep deterministic policy gradients (DDPG)-based reinforcement learning method to select appropriate weights for the automation's cost function such that the automation can adapt its desired steering policy if needed. Finally, we test our convention formation framework's performance in resolving a human driver and automation conflict.

In summary, the main contributions of this paper are

(i) creating a modular structure that separates partner-specific conventions from task-dependent representations;

(ii) characterizing a map that can connect the space of conventions to outcomes of a human-automation interaction for resolving the reverse intent conflict;

(iii) development of an adaptable automation system that can reach to a desirable form of a convention with a human driver.

The outline of this paper is as follows: Section 2 presents the model of the adaptive haptic shared control paradigm. Section 3 presents the principles of convention formation in a haptic shared control paradigm. This section proposes a modular structure that can separate partner-specific conventions from task-dependent representations. Using this structure, we create a map to connect different forms of the conventions with the outputs of the human-automation interaction. We further develop a reinforcement learning (RL)-based model predictive controller (MPC) for the automation system to enable it to reach a desirable convention dynamically. Section 4 presents numerical results, followed by Section 5, which presents the conclusions and plan.

2. ADAPTIVE HAPTIC SHARED CONTROL FRAMEWORK

Fig. 1 shows a schematic of an adaptive haptic shared control paradigm. Three entities each impose a torque on the steering wheel: a driver through his arm and hand, an automation system through a motor, and the road through the steering linkage.

We adopt identical structures to model human and automation systems. We model the driver as a hierarchical two-level controller. The upper-level control represents the cognitive controller, and its output, θ_H, represents the driver's intent. The lower level represents the human's biomechanics, $z_H = [k_H \; b_H]$, and is considered back-drivable [17]. Here k_H and b_H are the stiffness and damping of the driver's biomechanics. To indicate that driver's biomechanic parameters vary with changes in grip on the steering wheel, use of one hand or two, muscle co-contraction, or posture changes, we have drawn an arrow through human z_H. Similarly, the automation system is modeled as a higher-level artificial intelligence (AI) with an intent θ_A coupled with a lower-level impedance controller. The automation system is also considered back-

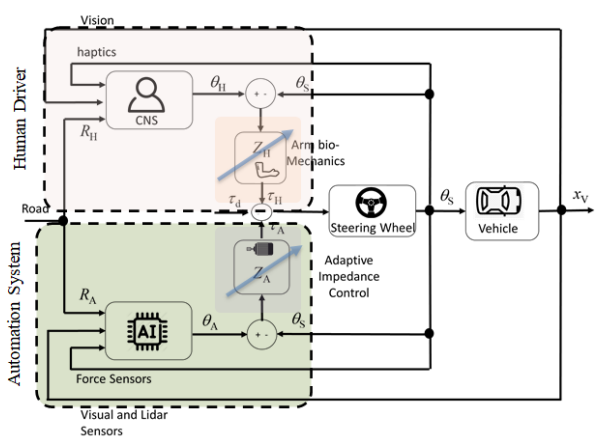

Fig. 1. A schematic of a haptic shared control paradigm. The human and automation are modeled as a two-level controller; their dynamics are coupled through the steering wheel.

drivable, and the gains of the impedance controller, $z_A = [k_A\ b_A]$, are designed to be modest rather than infinite. In other words, the automation is not intended to behave as an ideal torque source; instead, the automation imposes its command torque τ_A through an impedance z_A that is approximately matched to the human impedance z_H.

By combining the vehicle dynamics and the dynamics of the human-machine interaction on the steering wheel, the equation of motion for a haptic shared control can be expressed as

$$\dot{x}(t) = f(x(t), p(t), w(t)) + B_A(p(t))u_A(t) + B_H(p(t))u_H(t), \tag{1}$$

where $x = [\theta_{SW}\ \dot{\theta}_{SW}\ \theta_S\ \dot{\theta}_S\ x_V^T]^T$, are the state of the integrated system including the angle and angula velocity of the steering wheel, th angle and angular velocity of the steering shaft, and the state's of vehicle, $u_A = [\theta_A\ \dot{\theta}_A]^T$, and $u_H = [\theta_H\ \dot{\theta}_H]^T$ are the automation system and the human driver's control commands, $p(t) = [z_H^T\ z_A^T]^T$ are the time-varying parameters of the system, and $w(t) = \tau_V$ is exogenous signals, and

$$f = \begin{bmatrix} \dot{\theta}_{SW} \\ \frac{-b_H\dot{\theta}_{SW} - k_H - \theta_{SW} - K_T(\theta_{SW} - \theta_S)}{J_{SW} + J_H} \\ \dot{\theta}_S \\ \frac{-\left(\frac{r_S}{r_M}\right)^2 b_A\dot{\theta}_S - \left(\frac{r_S}{r_M}\right)^2 k_A\theta_S + K_T(\theta_{SW} - \theta_S) + \tau_v)}{J_S + \left(\frac{r_S}{r_M}\right)^2 J_M} \end{bmatrix}, \tag{2a}$$

$$B_H(p(t)) = \frac{1}{J_{SW} + J_H}\begin{bmatrix} 0 & 0 & 0 & 0 \\ k_H & b_H & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{2b}$$

$$B_A(p(t)) = \frac{\frac{r_S}{r_M}}{J_S + \left(\frac{r_S}{r_M}\right)^2 J_M}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & k_A & b_A \\ 0 & 0 & 0 & 0 \end{bmatrix}, \tag{2c}$$

where r_S/r_M is the mechanical advantages of a timing belt connecting the motor to the steering wheel, k_T is the stiffness of the torque sensor, J_H, J_M, J_{sw}, and J_s are the inertia of human's bio-mechanics, motor, steering wheel, and steering shaft, respectively. The details of the model are given in [18].

3. CONVENTION FORMATION THROUGH INTENTION NEGOTIATION

In a haptic shared control paradigm, there can be scenarios where a human and automation face a conflict. For instance, Fig. 2 shows a scenario when both human and automation systems see an obstacle and select a different path to avoiding it. In such a scenario, if both the human and automation select the same impedance ($z_H = z_A$), their control commands cancel out each other, and the vehicle hit an obstacle. In addition, to reverse intents, the other forms of conflicts can be considered when (i) one agent does not provide any control inputs (e.g., one agent does not detect an obstacle), (ii) too much or too few inputs (e.g., two agents have different perceptions from the size/position of an obstacle), (iii) control inputs arrive too early or too late, and (iv) additional inputs cause conflict (e.g., disturbance feedback from the road).

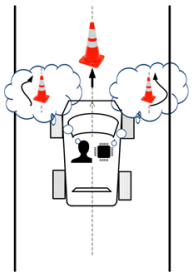

Fig. 2. Demonstration of a scenario when both human and automation systems select a different path for avoiding obstacle.

To potentially resolve a conflict such as having a reverse intent, the human and automation can adapt their control strategies by modulating their impedance parameters [18] and also by updating their steering commands θ_H and θ_A. While there might be multiple strategies for resolving a conflict (e.g., updating their steering commands or modulating their impedance parameters), some of these strategies may be preferable to the human driver. The idea behind the convention formation is to narrow the possible strategies for collaboration into a subset of these strategies to which the human partner might naturally be more gravitated. Below, we discuss the required steps for designing a set of adaptable and convention-based control transfer strategies to enhance joint driving performance.

3.1. Distinguishing partner-specific conventions from task-dependent representations

To learn and leverage conventions, we must create a modular structure separating partner-specific conventions from task-dependent representations. To this end, in this paper, we consider a structure where the human and automation's steering commands at the higher level can be determined by optimizing cost functions J_H, and J_A, respectively. These cost functions are defined as a combination of a set of hand-coded features $\phi_H = [\phi_{H,1}\ \cdots\ \phi_{H,n_H}]^T$ and $\phi_A = [\phi_{A,1}\ \cdots\ \phi_{A,n_A}]^T$ and vectors of the weights $w_H = [w_{H,1}\ \cdots\ w_{H,n_H}]$ and $w_A = [w_{A,1}\ \cdots\ w_{A,n_A}]$. In particular, $J_H = \phi_H w_H$ and $J_A = \phi_A w_A$. The hand-coded features can be defined as possible maneuvering paths and the control effort for each agent.

This paper focuses on developing a platform wherein the concept of conventions can be utilized for resolving a conflict between a human driver and an automation system. To this end, we select an example of conflict as shown

in Fig. 2. In this scenario, both humans and automation may see an obstacle, and they have two possible maneuvering trajectories r_{R} from the right side and r_{L} from the left side of the obstacle. To determine their steering commands, they both solve an optimization problem as follows:

$$\min_{\theta_{\mathrm{H}}} J_{\mathrm{H}}(x,u) = \sum_{k=1}^{N_p} \Big(\|y(k) - r_{\mathrm{R}}(k)\|^2_{w_{\mathrm{HR}}} + \|y(k) - r_{\mathrm{L}}(k)\|^2_{w_{\mathrm{HL}}} + \|\theta_{\mathrm{H}}(k)\|^2_{w_{\mathrm{H}\theta}} \Big), \tag{3a}$$

$$\min_{\theta_{\mathrm{A}}} J_{\mathrm{A}}(x,w) = \sum_{k=1}^{N_p} \Big(\|y(k) - r_{\mathrm{R}}(k)\|^2_{w_{\mathrm{AR}}} + \|y(k) - r_{\mathrm{L}}(k)\|^2_{w_{\mathrm{AL}}} + \|\theta_{\mathrm{A}}(k)\|^2_{w_{\mathrm{A}\theta}} \Big), \tag{3b}$$

$$\text{s.t.} \quad x_{\mathrm{d}}(k+1) = f_{\mathrm{d}}(x_{\mathrm{d}}(k), p(k), d(k)) + B_{\mathrm{d,H}}(p(k)_{\mathrm{d}})u_{\mathrm{H}}(k) + B_{\mathrm{d,A}}(p(k))u_{\mathrm{A}}(k), \tag{3c}$$

where y is the lateral position of the vehicle around the obstacle, r_{R} and r_{L} are desired reference trajectories around the obstacle (determined at a path planning level). Equation (3c) describes the discrete dynamics of the haptic shared control framework. In this paper, we derived the discrete dynamics using zero-order hold on the inputs and a sample time of T_s and N_{p} is a horizon time. Here $\phi_{\mathrm{H},1} = \phi_{\mathrm{A},1} = \|y - r_{\mathrm{R}}\|$ and $\phi_{\mathrm{H},2} = \phi_{\mathrm{A},2} = \|y - r_{\mathrm{L}}\|$ represent possible strategies for maneuvering the vehicle from the right or left of the obstacles. The last term (i.e., $\phi_{\mathrm{H},3} = \|\theta_{\mathrm{H}}\|$ and $\phi_{\mathrm{A},3} = \|\theta_{\mathrm{A}}\|$) represent the control effort value. The weight distribution over these features determines the interaction behavior between humans and automation. Three examples of these behaviors are discussed below.

First, let define ϵ_{comp}, ϵ_{coop},and $\epsilon_{\mathrm{undecided}}$ to be three design parameters. Also, let assume $w_{\mathrm{H}\theta} = w_{\mathrm{A}\theta} = \epsilon$, where ϵ is a positive constant. Furthermore, assume $w_{\mathrm{AR}} = 1 - w_{\mathrm{AL}}$ and $w_{\mathrm{HR}} = 1 - w_{\mathrm{HL}}$. Then, for a fixed $w_{\mathrm{H}} = [w_{\mathrm{HR}}\ w_{\mathrm{HL}} w_{\mathrm{H}\theta}]$, the automation systems can adopt different levels of cooperativeness by assigning how weight vectors $w_{\mathrm{A}} = [w_{\mathrm{AR}}\ w_{\mathrm{AL}} w_{\mathrm{A}\theta}]$ shall be distributed. If the driver selects one of the two paths ($|w_{\mathrm{HR}} - w_{\mathrm{HL}}| > \epsilon_{\mathrm{undecided}}$), then three human-automation interaction behavior at the higher-level can be defined as

1) Uncooperative automation: When automation selects a different path than the human driver. This behavior can be described when $|w_{\mathrm{HR}} - w_{\mathrm{AL}}| \leq \epsilon_{\mathrm{comp}}$. Similarly, it can be described when $|w_{\mathrm{HL}} - w_{\mathrm{AR}}| \leq \epsilon_{\mathrm{comp}}$.
2) Undecided automation: The automation assigns similar weights to the two paths around the obstacle. This behavior can be described when $|w_{\mathrm{AR}} - w_{\mathrm{AL}}| \leq \epsilon_{\mathrm{undecided}}$.
3) Cooperative automation: When automation selects a path similar to the human driver. This behavior can be described when $|w_{\mathrm{HR}} - w_{\mathrm{AR}}| \leq \epsilon_{\mathrm{coop}}$. Similarly, $|w_{\mathrm{HL}} - w_{\mathrm{AL}}| \leq \epsilon_{\mathrm{coop}}$.

Note, the driver can also be undeceived, meaning $|w_{\mathrm{HR}} - w_{\mathrm{HL}}| \leq \epsilon_{\mathrm{undecided}}$ but for the sake of brevity, we do not consider such a case in this paper.

3.2. Design an adaptable automation system

To form desirable conventions, the automation system shall be able to update the distribution of its weight vector w_{A} [19] and search for the optimal strategy. Here, we consider adjusting automation's strategies by solving

$$w_{\mathrm{A}} = \underset{w_{\mathrm{A}} \in \mathcal{W}_{\mathrm{A}}}{\operatorname{argmax}} \Big(\mathcal{R}_{\mathrm{A},1}(w_{\mathrm{A}}, w_{\mathrm{H}}, x), \cdots, \mathcal{R}_{\mathrm{A,n}}(w_{\mathrm{A}}, w_{\mathrm{H}}, x) \Big), \tag{4}$$

where $\mathcal{R}_{\mathrm{A},i}(w_{\mathrm{H}}, w_{\mathrm{A}}, x)$, for $i = 1, \cdots, n$, is a reward function describing the goodness of the formed convention. It should be noted that $\mathcal{R}_{\mathrm{A}} = [\mathcal{R}_{\mathrm{A},1}, \cdots, \mathcal{R}_{\mathrm{A,n}}]^{\mathrm{T}}$ is not necessarily the same as J_{A}. Instead, $\mathcal{R}_A$ shall be selected to consider human and automation's joint costs. To solve (4), we developed an episode-based policy search using deep deterministic policy gradients (DDPG) technique to determine automation's optimal policies (i.e., automation's model-predictive weights vector w_{A}– See Fig. 3). We selected DDPG since it is deemed particularly powerful in handling continuous action spaces and its relative simplicity. Our action space is naturally continuous, as the choice of the automation's weight vector can take any real value in a constrained range.

Fig. 3 shows the structure of the DDPG approach that includes two neural networks named critic and actor networks. At each time-step k, the DDPG algorithm receives a system states feedback $S_k = [x^{\mathrm{T}}(k)\ p^{\mathrm{T}}(k)]^{\mathrm{T}}$ as its observation, and generates action $A_k = \{\omega_{\mathrm{AR,i}}, \omega_{\mathrm{AL},k}\}$ from the action set $\mathbb{A}$ according to a policy $\pi(S_k)$. The undertaken action A_k (penalty weights) results in a scalar reward r_k, and the updated system state S_{k+1}.

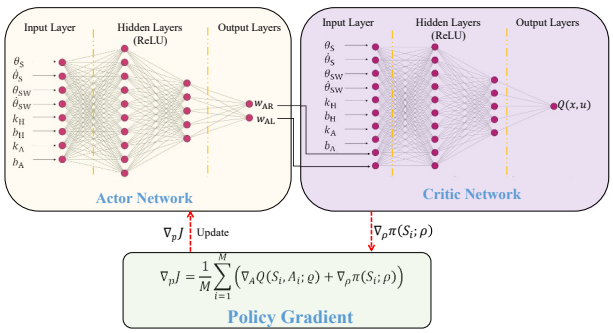

Fig. 3. Schematic diagram of the DDPG with the system states as the input for actor and critic networks.

A DDPG algorithm aims to determine an optimal policy such that the aggregated discounted future reward defined as $\mathcal{R}_{\mathrm{A,i}} = \sum_{i=0}^{\infty} \gamma^i \, r_{k+i}$ is maximized. Here, $\gamma = (0, 1]$ is the discount factor. To this end, the DDPG algorithm uses the Q-value function $Q(S,A)$ and the deterministic policy $\pi(S)$. Here, S and A are state and action spaces. In the learning phase, the DDPG algorithm updates the actor and critic network properties at each time step and stores the experiences in the previous time steps by a circular buffer. A mini-batch of randomly sampled experiences from the circular buffer updates the actor and critic [20]. The DDPG algorithm at each training step perturbs the action the policy selects using stochastic noise.

The DDPG agent contains four function approximators name: actor $\pi(S;\rho)$, target actor $\pi_{\mathrm{trg}}(S;\rho_{\mathrm{trg}})$, critic $Q(S,A;\varrho)$, and target critic $Q_{\mathrm{trg}}(S,A;\varrho_{\mathrm{trg}})$ to estimate the value function and policy. Here $\{\rho,\ \rho_{\mathrm{trg}},\ \varrho,\ \varrho_{\mathrm{trg}}\}$ are the parameters of the networks. At the actor network, the policy $\pi(S;\rho)$ generates action A to maximize the long-term reward based on the states S. At the critic network, the $Q(S,A;\varrho)$ function generates the long-term reward expectation based on the states S and action A. The target actor and target critics with the same structure and parameterization as the actor and critics, respectively, are employed to improve the stability of the optimization. During the training phase, the DDPG agent adjusts the parameter values in $\{\rho,\ \rho_{\mathrm{trg}},\ \varrho,\ \varrho_{\mathrm{trg}}\}$ and these parameters remain at their tunned value after the training phase. Algorithm 1 described the training of the DDPG network at each time step [20]. The DDPG algorithm updates the critics' network parameters ϱ by minimizing the following loss function

$$L = \frac{1}{M}\sum_{k=1}^{M}\left(Q(S_k,A_k;\varrho) - \ell_k\right)^2, \tag{5}$$

where M is the number of DDPG's training episodes.

$$\begin{aligned} \ell_k &= r(S_k,A_k) + \gamma Q\left(S_{k+1}, \kappa(S_{k+1});\varrho\right), \\ \kappa(S_{k+1}) &= \underset{\mathrm{A}}{\operatorname{argmax}}\ Q(S,A). \end{aligned} \tag{6}$$

Here, $\kappa(S)$ is a greedy policy from the Q-learning algorithm. The sampled policy gradient $\nabla_p J$ for maximizing the discounted reward $\mathcal{R}$ is

$$\nabla_p J = \frac{1}{M}\sum_{i=1}^{M}\Big(\nabla_A Q(S_i,A;\varrho) + \nabla_\rho \pi(S_i;\rho)\Big). \tag{7}$$

Here, $\nabla_A Q$ and $\nabla_\rho \pi$ are gradients of the critic and actor, respectively, for the action computed by the actor A and the actor parameters ρ. These gradients are evaluated for states S_k. The sampled policy gradient $\nabla_p J$ updates the actors' network parameters ρ. The target actor ρ_{trg} and critic, ϱ_{trg} parameters in the DDPG agent are updated based on the smoothing method at every time sample with a smoothing factor $\mathscr{K}$.

$$\varrho_{\mathrm{trg}} = \mathscr{K}\varrho + (1-\mathscr{K})\varrho_{\mathrm{trg}}, \tag{8a}$$

Algorithm 1: DDPG agents training algorithm.

- Initialization of actor $\pi(S;\rho)$ and critic $Q(S,A;\vartheta)$ networks with random weights ρ and ϱ.
- Initializing target networks $\pi_{trg}(S;\rho_{trg})$ and critic $Q_{trg}(S,A;\varrho_{trg})$ with weights $\rho_{trg} = \rho$ and $\varrho_{trg} = \varrho$.
- Set up an empty experience buffer R.

for episode= 1 to M **do**

1: Begin with an Ornstein-Uhelnbeck (OU) noise $\mathcal{N}$ for exploration.
2: Receive initial observation state.
3: Apply action A, Observe the reward R and next observation S'.
4: Store transitions $(S_i,\ A_i,\ R_i,\ S_{i+1})$ into experience buffer R.
5: Sample a random mini-batch of M experiences from the experience buffer.
6: Value function target $y_i = R_i + \gamma Q_{trg}(S'_i, \pi_{trg}(S'_i;\rho_{trg});\varrho_{trg})$.
7: Update the critic parameters by minimizing the loss L across all sampled experiences.
8: Update the actor policy using the sampled policy gradient $\nabla_\rho J$.
9: Update the target networks by smoothing factor $\mathscr{K}$.

end

$$\rho_{\mathrm{trg}} = \mathscr{K}\rho + (1-\mathscr{K})\rho_{\mathrm{trg}}. \tag{8b}$$

The aggregated reward and the state errors are stored in their dedicated buffer in each episode. These buffers supply the observation and the reward value to the DDPG algorithm. The update rate of the automation systems' penalty weights in the training phase on the DDPG agent is the same as the episode length. In this paper, the nonlinear MPC of the automation system is executed 100 times for each set of weights. In each execution time, the model is propagated to cover the view horizon.

3.3. Characterization of convention maps

To determine the desirable $\mathcal{R}_{\mathrm{A}}$, it can be argued that a desirable convention forms when humans and automation adapt their behaviors to minimize their combined cost functions. Therefore, $\mathcal{R}_{\mathrm{A}}$ shall be selected to consider human and automation's joint costs. However, solving an optimization such as, $\min_{\theta_{\mathrm{H}},\theta_{\mathrm{A}}}\Big(J_{\mathrm{H}}(\theta_{\mathrm{H}},\theta_{\mathrm{A}}) + J_{\mathrm{A}}(\theta_{\mathrm{H}},\theta_{\mathrm{A}})\Big)$ may result in solutions that may favor one agent much more than the other ($J_{\mathrm{A}}(\theta^{\#}_{\mathrm{H}},\theta^{\#}_{\mathrm{A}}) \ll J_{\mathrm{H}}(\theta^{\#}_{\mathrm{H}},\theta^{\#}_{\mathrm{A}})$) which may not be agreeable to one agent. To address this issue, the cooperative-competitive (co-co) solution concept has been established [21]. The co-co concept models a situation

where one agent pays/receives an incentive to implement a strategy that minimizes the combined cost function. Specifically, employing a co-co game, the original game can be split as the sum of a purely cooperative game, where both players have the same cost function, and a purely competitive (i.e., zero-sum) game, where the players have opposite cost functions. An issue regarding the traditional form of the cooperative-competitive game is that the incentive amount shall be known or iteratively calculated, which may not be practical or computationally tractable.

To address this issue, in this paper, instead of solving the co-co game, we split the combined cost function of the human and automation systems into two competitive J_{comp} and cooperative J_{coop} cost functions and calculate their values at their Nash equilibrium [22]. In particular,

$$J_{\text{coop}}(\theta_{\text{H}}^*, \theta_{\text{A}}^*) = \frac{J_{\text{H}}(\theta_{\text{H}}^*, \theta_{\text{A}}^*) + J_{\text{A}}(\theta_{\text{H}}^*, \theta_{\text{A}}^*)}{2}, \tag{9a}$$

$$J_{\text{comp}}(\theta_{\text{H}}^*, \theta_{\text{A}}^*) = \frac{J_{\text{H}}(\theta_{\text{H}}^*, \theta_{\text{A}}^*) - J_{\text{A}}(\theta_{\text{H}}^*, \theta_{\text{A}}^*)}{2}. \tag{9b}$$

It follows from (9) that $J_{\text{H}} = J_{\text{coop}} + J_{\text{comp}}$ and $J_{\text{A}} = J_{\text{coop}} - J_{\text{comp}}$. The steering angle pair θ_{H} and θ_{A} is a Nash solution if the following holds.

1) The control θ_{H}^* solves the optimal control problem of the human driver's cost function. Specifically, $\theta_{\text{H}}^* = \underset{\theta_{\text{H}}}{\text{argmax}}\Big(J_{\text{H}}(x, \theta_{\text{H}}, \theta_{\text{A}}^*)\Big)$, where θ_{A}^* is the optimal solution of automation's cost function.
2) The control θ_{A}^* provides a solution to the optimal control problem of the automation's cost. Specifically, $\theta_{\text{A}}^* = \underset{\theta_{\text{A}}}{\text{argmax}}\Big(J_{\text{A}}(x, \theta_{\text{H}}^*, \theta_{\text{A}})\Big)$, where θ_{H}^* is the optimal solution of automation's cost function.

The two optimization problems will be solved iteratively until the Nash optimal solution $(\theta_{\text{H}}^*, \theta_{\text{A}}^*)$ is reached. Specifically, $J_{\text{H}}(\theta_{\text{H}}^*, \theta_{\text{A}}^*) \leq J_{\text{H}}(\theta_{\text{H}}^*, \theta_{\text{A}})$ and $J_{\text{A}}(\theta_{\text{H}}^*, \theta_{\text{A}}^*) \leq J_{\text{A}}(\theta_{\text{H}}, \theta_{\text{A}}^*)$ We solve the two optimization problem using the C-GMRES technique [23]. The details of the C-GMRES are described in the [18].

The values of θ_{H}^* and θ_{A}^* depends on the distribution of human's weight vector ($w_{\text{H}} = [w_{\text{HR}}\ w_{\text{HL}}\ w_{\text{H}\theta}]$) and automation's weight vector ($w_{\text{A}} = [w_{\text{AR}}\ w_{\text{AL}}\ w_{\text{A}\theta}]$). To be able to design an adaptable automation system such that a desirable convention is formed (i.e., the cooperative cost is minimum and the competitive cost is zero), a map should be created that shows the values of J_{comp} and J_{coop} as a function of adopted weights by human w_{H} and automation w_{A}. To this end, we created the convention map by evaluating the values of J_{coop} and J_{comp} for a range of weights w_{H} and w_{A}.

4. NUMERICAL SIMULATIONS AND DISCUSSION

In this section, we present a series of simulation studies demonstrating the effectiveness of convention formation for resolving a conflict between a human driver and an automation system. The following simulations consider a scenario where the human driver and the automation system detect an obstacle and negotiate on controlling the steering wheel to avoid the obstacle safely. We consider the two cost functions in (3) for the human driver and automation system. Table 1 shows the numerical values used in the simulation. Here, we select different values for the parameters of the human driver and automation's impedance controllers to demonstrate different lower-level interaction modes (e.g., active safety vs. assistive mode).

Fig. 4 shows the competitive-cooperative cost functions values for a range of w_{A} and w_{H} in three lower-level interaction modes named active safety, neutral and assistive modes. Specifically, we define active safety mode when the parameters of the automation's impedance controller are larger than the parameters of the human driver's bio-mechanics ($z_{\text{A}} - z_{\text{H}} > \epsilon_1$), where ϵ_1 is a positive constant. The assistive mode is when the parameters of the automation's impedance controller are smaller than the parameters of the human driver's bio-mechanics ($z_{\text{H}} - z_{\text{A}} > \epsilon_1$). Finally, the neutral is when the human and automation's impedance parameters are almost the same ($|z_{\text{H}} - z_{\text{A}}| < \epsilon_1$). Here, we considered $z_{\text{A}} = 0.1 z_{\text{H}}$ in the assisitve mode, $z_{\text{A}} = z_{\text{H}}$ in the neutral and $z_{\text{A}} = 10 z_{\text{H}}$ in the active-safety mode. To create the conventions map, we considered $w_{\text{HR}} = 1 - w_{\text{HL}}$ and $w_{\text{H}\theta} = 1$. Similarly, we considered $w_{\text{AR}} = 1 - w_{\text{AL}}$ and $w_{\text{A}\theta} = 1$.

Fig. 4 shows that the cooperative surfaces' convention maps have two maximum points. These two maximum points are when $[w_{\text{HR}}\ w_{\text{AR}}] = [0\ 0]$ representing a scenario when both agents choose the left path to avoid the obstacle or when $[w_{\text{HR}}\ w_{\text{AR}}] = [1\ 1]$ representing a scenario when both agents choose the right path to avoid the obstacle. The competitive cost surfaces also have two maximum points. Specifically, the competitive cost value is maximum when $[w_{\text{HR}}\ w_{\text{AR}}] = [0\ 1]$ representing a scenario when the human driver chooses the left path, but automation chooses the right path to avoid the obstacle or when $[w_{\text{HR}}\ w_{\text{AR}}] = [1\ 0]$ representing a scenario when the human driver chooses the right path but automation choose the left path to avoid the obstacle.

Comparing the shape of cooperative and competitive surfaces for the three lower-level interaction modes, it can be seen that by changing the lower-level interaction mode, the flatness of the convention map for the cooperative/competitive value surfaces and the direction of curvature of the competitive value surface varies. Furthermore, it should be noted that since the competitive surface de-

Table 1. Numerical values for the system parameters in the simulation.

Parameters	Description	Haptic interaction mode		Units
		Active-safety	Assistive	
k_{H}	Driver arm's stiffness	0.5	3	N.m/rad
b_{H}	Driver arm's damping	0.2	0.5	N.m.s/rad
k_{A}	Automation's initial value of the arm's stiffness	0.5	3	N.m/rad
b_{A}	Automation's initial value of the arm's damping	0.2	0.5	N.m.s/rad
$\beta_{\mathrm{k_A}}$	Activation coefficient of k_{A}	1		-
$\beta_{\mathrm{b_A}}$	Activation coefficient of b_{A}	1		-
$\alpha_{\mathrm{k_A}}$	Memory coefficient of k_{A}	-1		-
$\alpha_{\mathrm{b_A}}$	Memory coefficient of b_{A}	-1		-
J_{H}	Driver arm's inertia	1×10^{-3}		kg.m^2
J_{SW}	Steering wheel inertia	1×10^{-2}		kg.m^2
J_{S}	Steering column inertia	1×10^{-2}		kg.m^2
J_{M}	Motor's inertia	1×10^{-3}		kg.m^2
K_{rmT}	Torque sensor stiffness	1000		N.m/rad
$r_{\mathrm{S}}/r_{\mathrm{M}}$	Timing belt mechanical advantage	1		-
m	Total mass of vehicle	1385		kg
I_{z}	Vehicle yaw moment of inertia	2065		kg.m^2
l_{f}	Distance from CG to front axle	1.114		m
l_{r}	Distance from CG to rear axle	1.436		m
r_{sw}	Steering ratio	15		
C_{f}	Front cornering stiffness	85,000		N/rad
C_{t}	Rear cornering stiffness	123,000		N/rad
v_{x}	Vehicle longitudinal velocity	20		m/sec
$N_{\mathrm{P_{Imp}}}$	Prediction horizon for impedance control	10		-
$N_{\mathrm{P_{H\_L}}}$	Prediction horizon for higher-level controller	100		-
$N_{\mathrm{C_{Imp}}}$	Control horizon for impedance control	2		-
$N_{\mathrm{C_{H\_L}}}$	Control horizon for higher-level controller	20		-
T_{s}	Simulation time step	0.002		sec
$I_{\mathrm{max\_out}}$	Maximum index for outer iteration C/GMRES algorithm	5		-
$I_{\mathrm{max\_in}}$	Maximum index for inner iteration C/GMRES algorithm	10		-
δ	KKT vector norm range	1×10^{-2}		-
λ_{rate}	Learning rate	0.001		-
γ	Discount factor	0.9		-
-	Mini-Batch size	128		-
-	Reply buffer size	1×10^{5}		-
-	Reply start size	300		-
k	Target update smoothing factor	0.01		-
M_{sub}	Time steps for fixe weights	200		-

fines the payoff of one agent to the other (zero-game part), zero competition is desirable in the interaction between two agents. Therefore, in defining the reward function for the RL agent (10), the second norm of the aggregated competitive value is employed in addition to the differential torque and the cooperative value.

Fig. 4 can be used as a map to connect forms of conventions to the outputs of human-automation interaction. For instance, Fig. 5 shows the human and automation interaction outputs associated with the three points shown with red, blue, and orange circles in Fig. 4 when both human and automation have similar impedance parameters. These three circles demonstrate three interaction modes where the automation is cooperative (red circle), undecided (orange circle), and uncooperative (blue circle), as discussed in Subsection 3.1. For all these three cases, the human's desired path for maneuvering the obstacle is from the right of the obstacle (i.e., $w_{\mathrm{HR}} = 1$). Therefore, the red circle represents a case where automation's desired path is from the left side (i.e., $w_{\mathrm{AL}} = 1$). The orange circle rep-

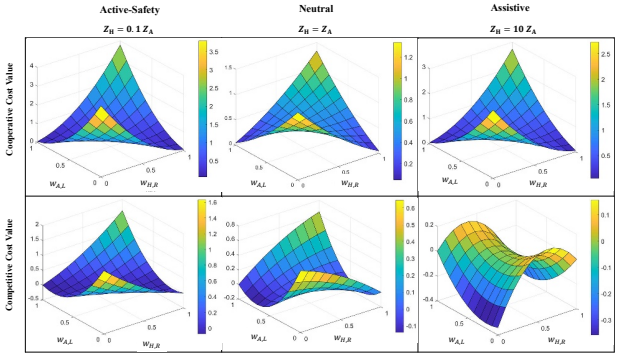

Fig. 4. Competitive-Cooperative cost functions values for lower-level interaction modes. The columns represent the interaction mode, and the rows depict the cooperative/competitive cost values from the Nash solution. In each surface has w_{H}, w_{H} and $V_{\mathrm{Coop}}/V_{\mathrm{Comp}}$ coordinates axis. The color bars demonstrate the range for each surface based on its minimum and maximum values.

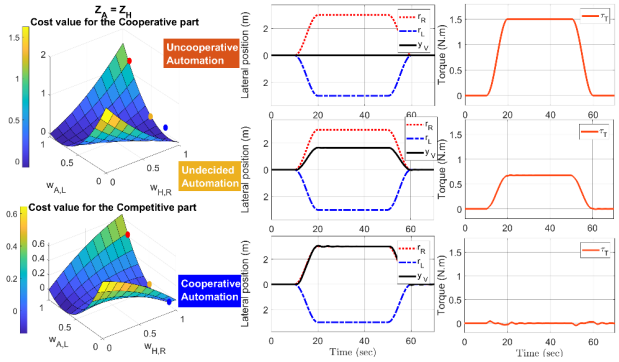

Fig. 5. The outputs of the human and automation interaction associated with the three points shown with red, blue, and orange circles in the neutral interaction mode ($Z_{\mathrm{H}} = Z_{\mathrm{A}}$). The surfaces represent the convention map's cooperative (middle-top) and competitive (middle-bottom) surfaces. The plots on the second column represent the lateral deviation of the vehicle from the centerline of the road. The last column represents the differential torque between the human driver and the automation system. The human drivers' behavior identifies each row based on their weight for the right directionw_{HR}.

resents a case where the automation's wights both right and left paths the same (i.e., $w_{\mathrm{AR}} = w_{\mathrm{AL}} = 0.5$). Finally, the blue circle represents a case where the automation's desired path is also from the right side (i.e., $w_{\mathrm{AR}} = 1$).

The first column of Fig. 5 shows the two possible paths for avoiding the obstacles and the vehicle's r_{R} and r_{L} and the vehicle's lateral position y_{V}. The second column shows the differential torque measured by the torque sensor τ_{T}.

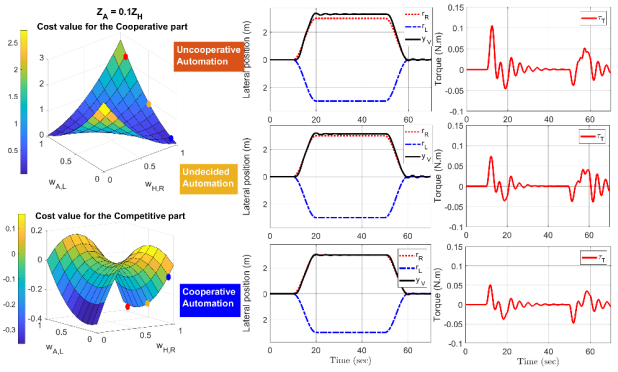

Fig. 6. The outputs of the human and automation interaction associated with the three points shown with red, blue, and orange circles in the assistive interaction mode ($Z_{\mathrm{H}} = 10Z_{\mathrm{A}}$). The surfaces represent the convention map's cooperative (right-top) and competitive (right-bottom) surfaces. The plots on the second column represent the lateral deviation of the vehicle from the centerline of the road. The last column represents the differential torque between the human driver and the automation system. The human drivers' behavior identifies each row based on their weight for the right direction w_{HR}.

It is demonstrated that when humans and automation have opposite paths since their impedance is the same, their control commands cancel out, and the vehicle hits the obstacle. To avoid such a conflict, two possible solutions can be presented. First, we can modulate the automation's impedance controller's parameters to yield or gain control as studied in our previous work [18]. Also, the automation's intent can be adapted to select a path similar to the human driver, as demonstrated in Fig. 5. In this paper, we focus on the latter approach.

When the human driver and the automation system have the same intent ($w_{\mathrm{AR}} = w_{\mathrm{HR}} = 1$ shown in the third row of Fig. 5), the differential torque is much smaller compared to the other two cases (the uncooperative automation shown and undecided automation). It should be noted that even though the competitive value for the blue and orange points are approximately the same since the cooperative value is different, the differential torque for the undecided automation is not zero. Also, the vehicle's lateral position is not the same as the right reference path for the undecided automation system.

Fig. 6 shows the outputs of the human and automation interaction in the assistive mode ($z_{\mathrm{A}} = 0.1z_{\mathrm{H}}$). Fig. 6 shows that for the three cases of uncooperative automation, undecided automation, and cooperative automation, the vehicle path is close to the human's desired path. Also, the differential torque is relatively small for all three cases. This is because automation's impedance is relatively small, meaning it only applies a low torque on the

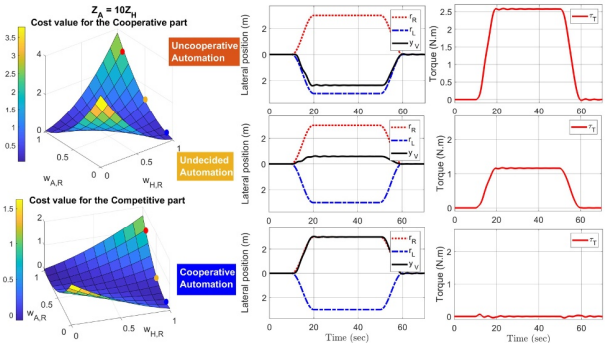

Fig. 7. The outputs of the human and automation interaction associated with the three points shown with red, blue, and orange circles in the active-safety interaction mode ($Z_H = 0.1Z_A$). The surfaces represent the convention map's cooperative (left-top) and competitive (left-bottom) surfaces. The plots on the second column represent the lateral deviation of the vehicle from the centerline of the road. The last column is for the differential torque between the human driver and the automation system. The human drivers' behavior identifies each row based on their weight for the right direction w_{HR}.

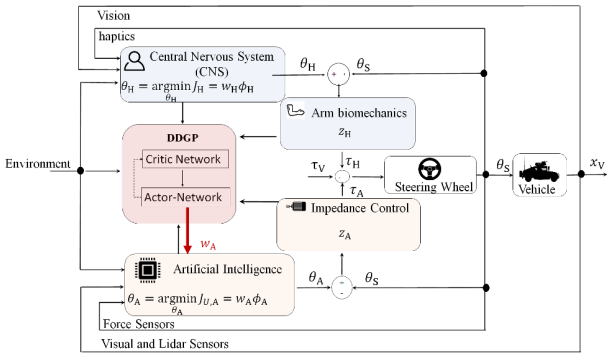

Fig. 8. Schematic diagram of the DDPG-based intent adaptation approach. The DDPG agents receive the observations from the model, lower-level, and higher-level controller, generating updated w_A.

steering wheel. In this scenario, the human driver mainly controls the vehicle.

Fig. 7 shows the outputs of the human and automation interaction in the assistive mode ($z_A = 10z_H$). Fig. 7 shows that the vehicle path is close to the automation for the three cases of uncooperative automation, undecided automation, and cooperative automation's desired path. When humans and automation have a reverse intent, the differential torque is relatively high in the active safety mode. This is because automation's impedance is relatively high, meaning it applies a high torque in the opposite direction as the human driver, which can cause discomfort to the driver but it ensures the safety of the vehicle.

As demonstrated in Figs. 5-7, the interaction between the human driver and automation system depends on the weights of the nonlinear MPC for each of them. A point in the convention maps demonstrates the agent's decision to choose the preferred path in the obstacle avoidance task. Based on human behavior, the automation system can adapt w_{AR} and w_{AL} to minimize the conflict in the interaction. To resolve a conflict between the automation system and the human driver, impedance modulation in the lower-level controller or intent adaption of the automation system in the higher-level controller can be used as a solution. The impedance modulation was studied in detail in [18]. In this paper, we discuss intent adaptation for resolving a conflict.

Fig. 8 shows how the weights of the nonlinear model predictive controller in the automation system are adjusted dynamically with the DDPG agent to minimize the conflict. For the DDPG agent, each actor and critic network has an input layer, an output layer, and three hidden layers of 100 units. In the hidden layer, the rectified linear unit (ReLU) is employed as the activation function, which projects the input to the output signal. The reward function in the DDPG algorithm is defined to minimize the integrated differential torque and cooperative value while maintaining the competitive value to zero:

$$r_k = \frac{1}{M_{sub}}\left(\sum_{k=1}^{M_{sub}}(-100\left\|J_{coop}\right\| - 100\left\|J_{comp}\right\| - \left\|\tau_T\right\|)\right), \tag{10}$$

which M_{sub} is the number of time steps with fixed weights in the cost function of the automation system. In the DDPG agent training phase, the number of the time step in each episode includes $100M_{sub}$. J_{coop} and J_{comp} are the equilibrium point's cooperative and competitive cost values. The hyperparameters of the DDPG agent are presented in Table 1.

Fig. 9 demonstrates the performance of the RL-based intent adaptation approach when the human driver wants to go more in the left direction to avoid the obstacle with weight $[w_{HR}\ \ w_{HL}] = [0.2\ \ 0.8]$ and they have equal impedance. On the contrary, the automation system preferred the right direction for avoiding the obstacle ($[w_{AR}\ \ w_{AL}] = [0.8\ \ 0.2]$). The human driver's and automation system's initial weight value is depicted by a red circle on the cooperative and competitive surfaces of the convention map (the first column). The lateral deviation of the vehicle (y_V) and the reference paths for the right (r_R) and left (r_L) sides are depicted in the first row of the second column. The measured differential torque is depicted in the second row of the second column, and the units of the y-axis are N.m. The weight value of the human driver

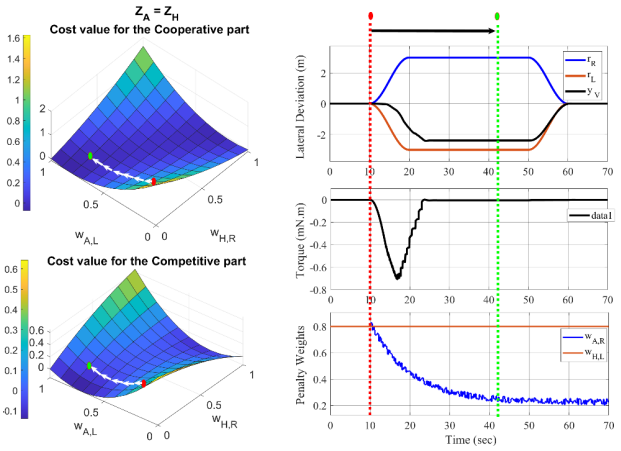

Fig. 9. RL-based intent adaptation for the case that the human and automation system have the same impedance. The first column demonstrates the convention map with the cooperative and competitive cost values. The second column shows the lateral deviation, differential torque, and the cost weights ωAR and ωHL. The red circle on the convention map is the initial weight value, and the green circle is the adapted weight value.

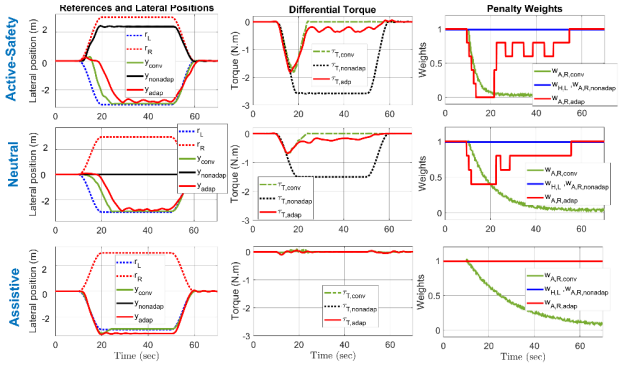

Fig. 10. RL-based intent adaptation for the different cases between the human and automation system. Each row represents the interaction mode (automation's impedance level). The first column is for references r_{L}, r_{R} and lateral positions for the non-adaptive $y_{nonadap}$, adaptive with a predefined rule y_{adap} and the proposed convention-based intent adaptation y_{conv}. The second and third columns represent the differential torques $\tau_{\mathrm{T,nonadap}}$, $\tau_{\mathrm{T,adap}}$, $\tau_{\mathrm{T,conv}}$ and the weights $\omega_{\mathrm{A,R,nonadap}}$, $\omega_{\mathrm{A,R,adap}}$, $\omega_{\mathrm{A,R,conv}}$, respectively.

and the automation system is demonstrated in the second column of the last row. By approaching the obstacle, the trained DDPG agent adopts the weight in the cost function of the automation system to minimize conflict between the automation and the human driver. The red dashed line demonstrates the start of the intent adaptation, and the green dashed line depicts the green circle on the convention map as the terminal weight of the automation system. After this line, the conflict is minimized. The value of the differential torque retained approximately zero after the intent adaption, which shows zero fight. Also, the competitive cost function value is zero. Therefore, the DDPG agent handled the intent adaptation to minimize the conflict while the vehicle avoided the obstacle.

Fig. 10 demonstrates the performance of the non-adaptive, adaptive with a predefined rule and the proposed convention-based intent adaptation for Active-Safety ($Z_{\mathrm{H}} = 0.1\ Z_{\mathrm{A}}$), Neutral ($Z_{\mathrm{H}} = Z_{\mathrm{A}}$) and assistive ($Z_{\mathrm{H}} = 10\ Z_{\mathrm{A}}$) cases. In the non-adaptive case, the cost weights of the automation system are constantly set to predefined values. In the adaptive with a predefined rule method, the cost weights are tuned based on the measured differential torque on the steering, which projects the measured torque to the six equally divided ranges [0, 2] N.m of the $\tau_{\mathrm{T,adap}}$ (11).

$$\omega_{\mathrm{A,R,adap}} = \begin{cases} 1 - \left[3|\tau_{\mathrm{T,adap}}|\right]/6, & |\tau_{\mathrm{T,adap}}| < 2\ \mathrm{N.m}, \\ 0, & |\tau_{\mathrm{T,adap}}| > 2\ \mathrm{N.m}. \end{cases} \tag{11}$$

In the first column, the references r_{L}, r_{R} and lateral positions for the non-adaptive y_{nonadap}, adaptive with a predefined rule y_{adap} and the proposed convention-based intent adaptation y_{conv} are demonstrated. In the second column, the differential torques between the human driver and the automation system cases with/without adaptive intent adaptions ($\tau_{\mathrm{T,nonadap}}$, $\tau_{\mathrm{T,adap}}$, $\tau_{\mathrm{T,conv}}$) are demonstrated. In the last column, the weight for the human cost $\omega_{\mathrm{H,L}}$, non-adaptive automation cost $\omega_{\mathrm{A,R,nonadap}}$, automation system with predefined adaptive weights $\omega_{\mathrm{A,R,adap}}$, and adaptive convention-based intent adaptions $\omega_{\mathrm{A,R,conv}}$ are demonstrated. In the case without the adaptive intent adaptions, the automation's weight $\omega_{\mathrm{A,R,nonadap}}$ is the same as the $\omega_{\mathrm{H,L}}$. Based on the results for the lateral deviation of the vehicle, the proposed RL-based intent adaption results in better cooperation between the human driver and the automation system. By approaching the obstacle, the trained DDPG agent adopts weights [$\omega_{\mathrm{A,R}}$, $\omega_{\mathrm{A,L}}$] in the cost function of the automation system to minimize conflict between the automation and the human driver. The differential torque in the transition period is decreased significantly in the active-safety and neutral modes. The error between the vehicle's lateral position and the reference path is decreased in the Assistive mode. Therefore, RL-based intent adaption by changing the automation's cost weights improves the performance of cooperation between the human driver and the automation system for all interaction modes.

5. CONCLUSIONS

This paper established a platform that allows studying the principles of co-adaptation (i.e., convention formation) between a human and automation system in a haptic shared control framework wherein both humans and automation collaboratively control the steering of a semi-automated ground vehicle to determine optimal handover strategies in uncertain circumstances. The framework consists of three main parts. The first part is focused on establishing a modular structure that can be used for separations of partner-specific conventions and task-dependent representations. Using this structure, the second section focuses on creating a map that can connect different forms of conventions to the output of the human-automation system. Finally, the third part focuses on designing an RL-based model predictive controller to search for automation's optimal strategy so that a desired form of convention can be reached. We applied the proposed platform to the problem of intent negotiation for resolving a conflict in a haptic shared control paradigm. The simulation results demonstrate that the handover strategies designed based on co-adaption can successfully resolve a conflict and improve the performance of the human automation teaming.

In future studies, to test and validate the performance of the proposed platform, the first step is to employ inverse reinforcement learning approaches to capture the distribution of the human weight vectors in performing a task. By capturing the human weight vector, we can realize whether the weight vector distribution can be used as a proxy for identifying the partner-specific conventions. In addition to validating this hypothesis, we plan to improve the automation system capability by employing Bayesian optimization (BO) to determine automation's optimal policies. Moreover, we plan to employ a transfer learning approach, such as a meta-inverse algorithm, so that knowledge of learned conventions can be used for interacting with new users or on new tasks. Finally, we plan to test this platform with human subjects in the loop.

CONFLICT OF INTEREST

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

REFERENCES

[1] F. Flemisch, M. Heesen, T. Hesse, J. Kelsch, A. Schieben, and J. Beller, "Towards a dynamic balance between humans and automation: Authority, ability, responsibility and control in shared and cooperative control situations," *Cognition, Technology and Work*, vol. 14, no. 1, pp. 3-18, 2012.

[2] J. C. F. De Winter and D. Dodou, "Preparing drivers for dangerous situations: A critical reflection on continuous shared control," *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, pp. 1050-1056, 2011.

[3] A. Shih, A. Sawhney, J. Kondic, S. Ermon, and D. Sadigh, "On the critical role of conventions in adaptive human-ai collaboration," *arXiv preprint arXiv:2104.02871*, 2021.

[4] S. Muggleton and N. Chater, *Human-like Machine Intelligence*, Oxford University Press, 2021.

[5] H. C. Siu, J. Pe na, E. Chen, Y. Zhou, V. Lopez, K. Palko, K. Chang, and R. Allen, "Evaluation of human-AI teams for learned and rule-based agents in hanabi," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16183-16195, 2021.

[6] W. Z. Wang, A. Shih, A. Xie, and D. Sadigh, "Influencing towards stable multi-agent interactions," *Proc. of Conference on Robot Learning*, pp. 1132-1143, PMLR, 2022.

[7] A. Xie, D. P. Losey, R. Tolsma, C. Finn, and D. Sadigh, "Learning latent representations to influence multi-agent interaction," *arXiv preprint arXiv:2011.06619*, 2020.

[8] E. Gkeredakis, "The constitutive role of conventions in accomplishing coordination: Insights from a complex contract award project," *Organization Studies*, vol. 35, no. 10, pp. 1473-1505, 2014.

[9] D. Meutsch and S. J. Schmidt, "On the role of conventions in understanding literary texts," *Poetics*, vol. 14, no. 6, pp. 551-574, 1985.

[10] C. L. Baker, J. Jara-Ettinger, R. Saxe, and J. B. Tenenbaum, "Rational quantitative attribution of beliefs, desires and percepts in human mentalizing," *Nature Human Behaviour*, vol. 1, no. 4, pp. 1-10, 2017.

[11] C. Brooks and D. Szafir, "Building second-order mental models for human-robot interaction," *arXiv preprint arXiv:1909.06508*, 2019.

[12] R. D. Hawkins, M. Kwon, D. Sadigh, and N. D. Goodman, "Continual adaptation for efficient machine communication," *arXiv preprint arXiv:1911.09896*, 2019.

[13] Y. Song and D. Scaramuzza, "Learning high-level policies for model predictive control," *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 7629-7636, 2020.

[14] Q. Lu, R. Kumar, and V. M. Zavala, "Mpc controller tuning using bayesian optimization techniques," *arXiv preprint arXiv:2009.14175*, 2020.

[15] Z. Lu and W. Lou, "Bayesian approaches to variable selection: a comparative study from practical perspectives," *The International Journal of Biostatistics*, vol. 18, no. 1, pp. 83-108, 2022.

[16] P. T. Jardine, M. Kogan, S. N. Givigi, and S. Yousefi, "Adaptive predictive control of a differential drive robot tuned with reinforcement learning," *International Journal of Adaptive Control and Signal Processing*, vol. 33, no. 2, pp. 410-423, 2019.

[17] P. Boehm, A. H. Ghasemi, S. O'Modhrain, P. Jayakumar, and R. B. Gillespie, "Architectures for shared control of vehicle steering," *IFAC-PapersOnLine*, vol. 49, no. 19, pp. 639-644, 2016.

[18] V. Izadi and A. H. Ghasemi, "Modulation of control authority in adaptive haptic shared control paradigms," *Mechatronics*, vol. 78, 102598, 2021.

[19] V. Izadi and A. H. Ghasemi, "Intent negotiation in a shared control paradigm with cooperative-competitive game," *Proc. of American Control Conference (ACC)*, IEEE.

[20] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[21] M. Stryszowski, S. Longo, D. D'Alessandro, E. Velenis, G. Forostovsky, and S. Manfredi, "A framework for self-enforced optimal interaction between connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6152-6161, 2020.

[22] N. Mehr, M. Wang, and M. Schwager, "Maximum-entropy multi-agent dynamic games: Forward and inverse solutions," *arXiv preprint arXiv:2110.01027*, 2021.

[23] T. Ohtsuka, "A continuation/gmres method for fast computation of nonlinear receding horizon control," *Automatica*, vol. 40, no. 4, pp. 563-574, 2004.

Vahid Izadi received a B.S. degree in electrical engineering and electronics from Hamedan University of Technology in 2012, an M.S. degree in electrical engineering and electronics from Iran University of Science and Technology, and a Ph.D. Student from the University of North Carolina at Charlotte. His research interests include control systems, robotics, and optimization.

Amir H. Ghasemi received a B.S. degree in mechanical engineering from the Ferdowsi University of Mashhad in 2005, an M.S. degree in mechanical engineering from Amirkabir University in 2008, and a Ph.D. degree in mechanical engineering from the University of Kentucky in 2012. He is currently an Assistant Professor in the Department of Mechanical Engineering and Engineering Science at the University of North Carolina at Charlotte. His research interests include control systems, robotics, and human-machine interactions.