Human-Automation Interaction for Assisting Novices to Emulate Experts by Inferring Task Objective Functions

Sooyung Byeon, Wanxin Jin, Dawei Sun, and Inseok Hwang
School of Aeronautics and Astronautics
Purdue University
West Lafayette, IN, USA
Email: {sbyeon, jin279, sun289, ihwang}@purdue.edu

2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC) | 978-1-6654-3420-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/DASC52595.2021.9594324

***Abstract*—In this paper, we propose a human-automation interaction scheme to improve the task performance of novice human users with different skill levels. The proposed scheme includes two interaction modes: *learn from experts mode* and *assist novices mode*. In the learn from experts mode, the automation learns from a human expert user such that the awareness of task objective is obtained. Based on the learned task objective, in the assist novices mode, the automation customizes its control parameter to assist a novice human user towards emulating the performance of the expert human user. We experimentally test the proposed human-automation scheme in a designed quadrotor simulation environment, and the results show that the proposed approach is capable of adapting to and assisting the novice human user to achieve the performance that emulates the expert human user.**

***Index Terms*—Human-automation interaction, Skill transfer, Remotely piloted quadrotor, Inferring task objective function, Inverse optimal control**

I. Introduction

Many aerospace applications such as air traffic management [1], unmanned aircraft systems (UAS) [2], remotely piloted systems (RPS) [3], and urban air mobility (UAM) rely on close human-automation collaboration to improve performance and safety of their operations since these human-automation systems seek to achieve a complementary competence of humans and automation. In this paper, we propose a human-automation interaction scheme to improve the task performance of novice human users with different skill levels, which helps enhance the performance of the entire system. In addition, the proposed human-automation interaction scheme could expedite human training or skill transfer. We consider remotely piloted aircraft or drone/quadrotor control as a driving example in this paper, but the proposed methodology is general enough to be applicable to other human-automation systems such as semi-autonomous cars [4], robotic rehabilitation systems [5], assistive exoskeletons [6], and teleoperation [7].

A key component in a human-automation system is the interaction scheme between humans and automation, which determines the performance of the whole system. Developing an efficient and human-friendly interactive control scheme is challenging, since it is usually mission-specific, and yet in general requires the integration of human motion prediction [8], [9], mission objective awareness [10], and shared control decision making [11]. Existing interactive control techniques developed in the field of human-automation interaction have achieved significant progress [12]–[14], but most of them typically assume a fixed human role in the interaction paradigm; in other words, the interaction models the human to be either a leader that provides situational guidance to the automation [8], [12]–[16], or a follower who yields to the automation that is pre-trained/programmed [6], [17]–[20]. Consequently, a limitation of these interactive control schemes is the lack of adaptivity and customization to different human users with possibly different knowledge or skill levels. To overcome this limitation, this paper develops a new human-automation interaction scheme, which can provide a personalized assistance based on the different skill levels of human users.

The proposed scheme assumes two types of human users: *experts* and *novices*. The expert knows how to operate the system optimally based on his/her own knowledge or experiences by minimizing an implicit task objective function. On the other hand, the novice's control inputs are optimal with respect to his/her own understanding of the task objective function, which is usually different from that of the expert. The automation interacts with both groups of human users via two interaction modes: in the *learn from expert mode*, the automation learns the task objective function of the expert human user using his/her demonstrations. The task objective function underlying the expert's demonstration is inferred using the inverse optimal control technique. In the *assist novice mode*, the automation adjusts the control parameters of the system such that the novice's demonstration minimizes the learned expert's task objective function as we consider it as a task performance measure. To that end, we iteratively integrate the inverse optimal control and the gradient decent approach to provide personalized assistance. We experimentally test and demonstrate the proposed human-automation scheme using a quadrotor landing simulation testbed. The test results show that the proposed approach is capable of adapting to different skill

The authors would like to acknowledge that this work is supported by NSF CNS-1836952. The Institutional Review Board at Purdue University approved the study.

levels of novice human users and assisting them to achieve the performance that emulates the expert human user.

A. Related Work

1) Human Motion Characterization: A smooth or proactive interaction always requires a model of human motion such that efficient prediction and estimation can be performed by the automation. Existing models to characterize human motion can be classified into probabilistic graphical methods, including Gaussian mixture models [21]–[23], hidden Markov models [24], [25], and optimal control methods [8], [11], [15], [26]. Compared to probabilistic models, optimal control models fully account for the kinematics and/or dynamics constraints in human motion and the human motor control principle has well justified with the evidences in neuroscience and biomechanics [27], [28], thus has more generality and superiority [26].

2) Learning Task Objective Functions: A requirement in human-automation interaction is to enable the automation to be aware of the task objective. In general, the objective of complex tasks is difficult to be manually specified a priori. Inverse optimal control [29]–[31] or inverse reinforcement learning techniques [10], [32]–[34] can be applied to learn the task objective function from data. By assuming the task objective function to be a weighted sum of specified features, the inverse optimal control seeks to infer feature weights from the observational demonstrations.

3) Human-Automation Control Coordination: Another line of human-automation interaction research focuses on how to coordinate the control authority between automation and a human user. For example, in [13], the authors propose to generate the automation's decisions by accounting for a human user's corresponding response. In [8], automation planning is made by predicting a human user's motion based on the inverse optimal control. In [11], the adaption rules between automation and human users are developed based on game theory, which achieves continuous adaption. In our development, we choose to adapt the automation's control parameter using the gradient method.

The remaining paper is organized as follows: in Section II, we formulate the problem and propose an overall human-automation interaction scheme. Detailed techniques used for each mode of the interaction scheme are provided in Section III. In Section IV, an illustrative experiment is introduced to demonstrate the proposed scheme, and the experimental results are discussed. Finally, conclusions are drawn in Section V.

II. Proposed Human-Automation Interaction

A. Problem Definition

Our goal is to develop an interaction scheme where for the expert human user, the automation can learn his/her task objective and for the novice human user, the automation will automatically adjust its control parameter to assist him/her to emulate the expert. To deal with this problem, we define the following elements.

We model the dynamics of a human-automation system as

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \theta) \tag{1}$$

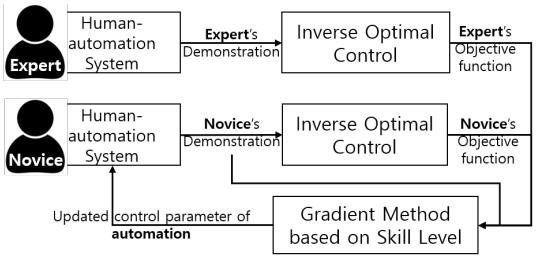

Fig. 1. Proposed Scheme for Human-Automation Interaction

where $\mathbf{x}_t \in \mathbb{R}^n$ is the system state; $\mathbf{u}_t \in \mathbb{R}^m$ is the human control input; $\theta \in \Theta \subset \mathbb{R}^p$ is the tunable control parameter in the automation, and Θ is a convex compact set including all feasible choices of parameters; $t = 0, 1, \cdots$ denotes the time instance; and the vector function f is assumed to be smooth. For the automation control parameter, the default value is denoted by $\bar{\theta}$, and we call the human-automation system with $\bar{\theta}$ as the *nominal configuration*. The task objective of the human-automation system is to minimize an accumulative task objective function, which is represented as a linear combination of features (or basis functions):

$$J(\mathbf{w}) = \sum_{t=0}^{T} \Phi(\mathbf{x}_t, \mathbf{u}_t) \cdot \mathbf{w} \tag{2}$$

where $\Phi(\mathbf{x}_t, \mathbf{u}_t) \in \mathbb{R}^r$ is the vector (function) whose elements are predefined relevant features; $\mathbf{w} \in \mathbb{R}^r$ is the vector of feature weights; and T is the time horizon. Note that this type of weight-feature objective is a typical formulation in human-automation research [8], [10], [12], [14], [29]. For convenience, we denote the above human-automation system as $\Sigma(\theta, \mathbf{w})$ if the system follows (1) and (2). We use $\boldsymbol{\xi}^{(\theta,\mathbf{w})}$ to denote the resulting (optimal) trajectory of $\Sigma(\theta, \mathbf{w})$:

$$\boldsymbol{\xi}^{(\theta,\mathbf{w})} = \left(\mathbf{x}_{0:T}^{(\theta,\mathbf{w})}, \mathbf{u}_{0:T}^{(\theta,\mathbf{w})}\right) \tag{3}$$

where $\mathbf{x}_{0:T}^{(\theta,\mathbf{w})}$ and $\mathbf{u}_{0:T}^{(\theta,\mathbf{w})}$ are the sequences of state and input with time from 0 to T, respectively. Here, we define $\mathbf{u}_T^{(\theta,\mathbf{w})}$ to be zero without loss of generality.

For the human-automation system $\Sigma\left(\bar{\theta}, \mathbf{w}\right)$ of nominal configuration, we consider two types of human users: expert human users and novice human users.

- The expert knows how to operate the system optimally based on his/her specific knowledge or experiences; that is, the expert's inputs are assumed to minimize an implicit task objective function of weights $\mathbf{w}_E$, whose specific value, however, is not explicitly known.
- The novice provides optimal inputs to the human-automation system, but such inputs are optimal with respect to his/her own understanding of the task objective function, characterized by $\mathbf{w}_N$, which is usually different from that of the expert with $\mathbf{w}_E$.

B. Proposed Human-Automation Interaction

As shown in Fig. 1, we propose a human-automation interaction scheme, which consists of two modes. The *learn*

from experts mode is used for expert human users. We assume that an expert human user is capable of operating the system efficiently based on specific knowledge or experiences, but typically the corresponding criterion underlying the expert's operation, i.e., the task objective function, is not explicitly given in a quantitative way [26], [35]. In this regard, the *learn from experts mode* is to learn the task objective function of the expert human user, which will be used as a performance measure of the whole system. Specifically, in the *learn from experts mode*, the automation records the trajectory/demonstration of the expert human user, $\boldsymbol{\xi}^{\left(\bar{\theta},\mathbf{w}_E\right)}$, and provides no intervention. Then, the task objective function weights $\mathbf{w}_E$ underlying the expert's trajectory are inferred using the inverse optimal control technique, which will be described in the next section.

The *assist novices mode* is designed for novice human users. Due to the lack of knowledge of the system, objective awareness, or operational experience, the task objective function weights $\mathbf{w}_N$ of the novice's operation are typically different from the weights $\mathbf{w}_E$ of the expert, and thus the trajectory $\boldsymbol{\xi}^{\left(\bar{\theta},\mathbf{w}_N\right)}$ of the novice human user, is different from the expert's trajectory/demonstration $\boldsymbol{\xi}^{\left(\bar{\theta},\mathbf{w}_E\right)}$. The goal of the *assist novices mode*, thus, is to adjust the control parameter θ of the automation such that the novice's trajectory resulting from $\Sigma\left(\theta,\mathbf{w}_N\right)$ minimizes

$$L=\sum_{t=0}^{T}\Phi\left(\mathbf{x}_t,\mathbf{u}_t\right)\cdot\mathbf{w}_E \tag{4}$$

as we consider the learned expert's task objective function as a task performance measure. To that end, we iteratively integrate the inverse optimal control and the gradient descent approach. Suppose that the current control parameter is θ, and the current trajectory of the novice human user results from $\Sigma\left(\theta,\mathbf{w}_N\right)$. First, the inverse optimal control is applied to infer $\mathbf{w}_N$ for the novice human user. Then, the gradient descent approach is applied to update the control parameter:

$$\theta\leftarrow\text{Proj}_{\Theta}\left(\theta-\eta\frac{dL}{d\theta}\right) \tag{5}$$

where Proj_{Θ} denotes the projection onto Θ; η is the step size; and $\frac{dL}{d\theta}$ is the gradient of L in (4) with respect to θ. By the chain rule, it is clear that

$$\frac{dL}{d\theta}=\frac{\partial L}{\partial\boldsymbol{\xi}^{(\theta,\mathbf{w}_N)}}\frac{\partial\boldsymbol{\xi}^{(\theta,\mathbf{w}_N)}}{\partial\theta} \tag{6}$$

The computation of the term $\frac{\partial\boldsymbol{\xi}^{(\theta,\mathbf{w})}}{\partial\theta}$ will be briefly reviewed in the next section.

The flow chart given in Fig. 1 summarizes the integration of the *learn from experts mode* and the *assist novices mode* in order to gradually update the control parameter such that the novice human user emulates the performance of the expert human user.

III. Technical Tools

A. Inverse Optimal Control

In both the *learn from experts mode* and *assist novices mode*, the inverse optimal control is used to infer the task objective function behind the human input, which is characterized by the weights $\mathbf{w}$ of pre-defined features in (2). To infer $\mathbf{w}$ using $\boldsymbol{\xi}^{(\theta,\mathbf{w})}$, we apply the inverse optimal control techniques as follows. Suppose $\left(\mathbf{x}_{t:t+l-1}^{(\theta,\mathbf{w})},\,\mathbf{u}_{t:t+l-1}^{(\theta,\mathbf{w})}\right)$ is a piece of demonstration, where l denotes the observation length. According to [31], [36], the Karush–Kuhn–Tucker (KKT) condition of optimal control for the discrete-time nonlinear system implies:

$$\begin{aligned}\mathbf{F}_x^{(t,l)}\lambda_{t:t+l-1}-\boldsymbol{\Phi}_x^{(t,l)}\mathbf{w}&=\mathbf{V}^{(t,l)}\lambda_{t+l}\\ \mathbf{F}_u^{(t,l)}\lambda_{t:t+l-1}+\boldsymbol{\Phi}_u^{(t,l)}\mathbf{w}&=\mathbf{0}\end{aligned} \tag{7}$$

where $\lambda\in\mathbb{R}^n$ denotes the dual variable; $\mathbf{F}_x^{(t,l)}\in\mathbb{R}^{nl\times nl}$, $\mathbf{F}_u^{(t,l)}\in\mathbb{R}^{ml\times nl}$, $\boldsymbol{\Phi}_x^{(t,l)}\in\mathbb{R}^{nl\times r}$, $\boldsymbol{\Phi}_u^{(t,l)}\in\mathbb{R}^{ml\times r}$, and $\mathbf{V}^{(t,l)}\in\mathbb{R}^{nl\times n}$ are given as:

$$\begin{aligned}\mathbf{F}_x^{(t,l)}&=\begin{bmatrix}I & \frac{-\partial f^T}{\partial\mathbf{x}_t^{(\theta,\mathbf{w})}} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \frac{-\partial f^T}{\partial\mathbf{x}_{t+l-2}^{(\theta,\mathbf{w})}}\\ 0 & 0 & \cdots & I\end{bmatrix},\quad \boldsymbol{\Phi}_x^{(t,l)}=\begin{bmatrix}\frac{\partial\phi^T}{\partial\mathbf{x}_t^{(\theta,\mathbf{w})}}\\ \vdots\\ \frac{\partial\phi^T}{\partial\mathbf{x}_{t+l-1}^{(\theta,\mathbf{w})}}\end{bmatrix},\\ \mathbf{F}_u^{(t,l)}&=\begin{bmatrix}\frac{\partial f^T}{\partial\mathbf{u}_{t-1}^{(\theta,\mathbf{w})}} & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \frac{\partial f^T}{\partial\mathbf{u}_{t+l-2}^{(\theta,\mathbf{w})}}\end{bmatrix},\quad \boldsymbol{\Phi}_u^{(t,l)}=\begin{bmatrix}\frac{\partial\phi^T}{\partial\mathbf{u}_{t-1}^{(\theta,\mathbf{w})}}\\ \vdots\\ \frac{\partial\phi^T}{\partial\mathbf{u}_{t+l-2}^{(\theta,\mathbf{w})}}\end{bmatrix},\\ \mathbf{V}^{(t,l)}&=\begin{bmatrix}0 & \frac{\partial f^T}{\partial\mathbf{x}_{t+l-1}^{(\theta,\mathbf{w})}}\end{bmatrix}^T\end{aligned} \tag{8}$$

respectively, where $\frac{\partial f^T}{\partial\mathbf{x}}$ denotes the transpose of the Jacobian matrix. Alternatively, Equation (7) can be written in a compact form:

$$\mathbf{H}^{(t,l)}\begin{bmatrix}\mathbf{w}\\ \lambda_{t+l}\end{bmatrix}=\mathbf{0} \tag{9}$$

where $\mathbf{H}^{(t,l)}=\left[\mathbf{H}_1^{(t,l)}\,\mathbf{H}_2^{(t,l)}\right]\in\mathbb{R}^{lm\times(r+n)}$ (named as *recovery matrix* in [31]) is defined as:

$$\begin{aligned}\mathbf{H}_1^{(t,l)}&=\mathbf{F}_u^{(t,l)}\left(\mathbf{F}_x^{(t,l)}\right)^{-1}\boldsymbol{\Phi}_x^{(t,l)}+\boldsymbol{\Phi}_u^{(t,l)}\\ \mathbf{H}_2^{(t,l)}&=\left(\mathbf{F}_x^{(t,l)}\right)^{-1}\mathbf{V}^{(t,l)}\end{aligned} \tag{10}$$

Note that the definition of the recovery matrix involves an inversion of a large matrix, but it is not a problem because of the following iterative property:

$$\mathbf{H}^{(t,l+1)}=\begin{bmatrix}\mathbf{H}_1^{(t,l)} & \mathbf{H}_2^{(t,l)}\\ \frac{\partial\phi^T}{\partial\mathbf{u}_{t+l}} & \frac{\partial f^T}{\partial\mathbf{u}_{t+l}}\end{bmatrix}\begin{bmatrix}I & \mathbf{0}\\ \frac{\partial\phi^T}{\partial\mathbf{x}_{t+l}} & \frac{\partial f^T}{\partial\mathbf{x}_{t+l}}\end{bmatrix} \tag{11}$$

with

$$\begin{aligned}\mathbf{H}^{(t,1)}&=\begin{bmatrix}\mathbf{H}_1^{(t,1)} & \mathbf{H}_2^{(t,1)}\end{bmatrix}\\ &=\begin{bmatrix}\frac{\partial f^T}{\partial\mathbf{u}_t}\frac{\partial\phi^T}{\partial\mathbf{x}_t}+\frac{\partial\phi^T}{\partial\mathbf{u}_t} & \frac{\partial f^T}{\partial\mathbf{u}_t}\frac{\partial f^T}{\partial\mathbf{x}_t}\end{bmatrix}\end{aligned} \tag{12}$$

By observation, the unknown weight $\mathbf{w}$ in the task objective function can be uniquely determined as long as the piece of demonstration is informative and the features are chosen in an distinguishable manner. In practice, however, it is possible that there is no $\mathbf{w}$ satisfying (9) due to the noise. In this case, $\mathbf{w}$ can be estimated by solving

$$\begin{bmatrix}\mathbf{w}^*\\ \lambda^*\end{bmatrix} = \arg\min_{[\mathbf{w}^T\ \lambda^T]^T} \left\| \mathbf{H}^{(t,l)} \begin{bmatrix}\mathbf{w}\\ \lambda\end{bmatrix} \right\| \tag{13}$$

where the variable is restricted to be normalized such that $\sum_{i=1}^{r} w_i = 1$.

B. Gradient Method

In the *assist novices mode*, applying (5) to update the control parameter requires the evaluation of gradient (6), which is challenging. To address this, we resort to the Pontryagin differentiable programming (PDP) method [37], where an auxiliary control system is constructed in order to compute $\frac{\partial \boldsymbol{\xi}^{(\theta,\mathbf{w})}}{\partial \theta}$. Specifically, given a fixed $\mathbf{w}$, we consider $\boldsymbol{\xi}^{(\theta,\mathbf{w})}$ is an optimal trajectory of the human-automation system $\boldsymbol{\Sigma}(\theta, \mathbf{w})$. Then an auxiliary control system of $\boldsymbol{\Sigma}(\theta, \mathbf{w})$ at $\boldsymbol{\xi}^{(\theta,\mathbf{w})}$ is a matrix linear system:

$$\bar{\mathbf{x}}_{t+1} = \mathbf{A}_t \bar{\mathbf{x}}_t + \mathbf{B}_t \bar{\mathbf{u}}_t + \mathbf{E}_t \tag{14}$$

with a quadratic task objective function:

$$\begin{aligned}\bar{J} = &\sum_{t=0}^{T-1} \mathrm{Tr}\left(\frac{1}{2} \begin{bmatrix}\bar{\mathbf{x}}_t\\ \bar{\mathbf{u}}_t\end{bmatrix}^T \begin{bmatrix} Q_t^{xx} & Q_t^{xu} \\ Q_t^{ux} & Q_t^{uu}\end{bmatrix} \begin{bmatrix}\bar{\mathbf{x}}_t\\ \bar{\mathbf{u}}_t\end{bmatrix} + \begin{bmatrix} Q_t^{xe}\\ Q_t^{ue}\end{bmatrix}^T \begin{bmatrix}\bar{\mathbf{x}}_t\\ \bar{\mathbf{u}}_t\end{bmatrix} \right) \\ &+ \mathrm{Tr}\left(\frac{1}{2} \bar{\mathbf{x}}_T^T Q_T^{xx} \bar{\mathbf{x}}_T + \left(Q_T^{xe}\right)^T \bar{\mathbf{x}}_T \right)\end{aligned} \tag{15}$$

where $\bar{\mathbf{x}}_t \in \mathbb{R}^{n\times p}$ and $\bar{\mathbf{u}}_t \in \mathbb{R}^{m\times p}$ are the auxiliary state and input, respectively; $\mathbf{A}_t \in \mathbb{R}^{n\times n}$, $\mathbf{B}_t \in \mathbb{R}^{n\times m}$, $\mathbf{E}_t \in \mathbb{R}^{n\times p}$, $Q_t^{xx} \in \mathbb{R}^{n\times n}$, $Q_t^{xu} \in \mathbb{R}^{n\times m}$, $Q_t^{ux} = \left(Q_t^{xu}\right)^T \in \mathbb{R}^{m\times n}$, $Q_t^{uu} \in \mathbb{R}^{m\times m}$, $Q_t^{xe} \in \mathbb{R}^{n\times p}$, and $Q_t^{ue} \in \mathbb{R}^{m\times p}$ are defined as:

$$\begin{aligned}\mathbf{A}_t &= \frac{\partial f}{\partial \mathbf{x}_t^{(\theta,\mathbf{w})}}, & \mathbf{B}_t &= \frac{\partial f}{\partial \mathbf{u}_t^{(\theta,\mathbf{w})}}, \\ \mathbf{E}_t &= \frac{\partial f}{\partial \theta}, & Q_t^{xx} &= \frac{\partial^2 h}{\partial \mathbf{x}_t^{(\theta,\mathbf{w})} \partial \mathbf{x}_t^{(\theta,\mathbf{w})}}, \\ Q_t^{xu} &= \frac{\partial^2 h}{\partial \mathbf{x}_t^{(\theta,\mathbf{w})} \partial \mathbf{u}_t^{(\theta,\mathbf{w})}}, & Q_t^{uu} &= \frac{\partial^2 h}{\partial \mathbf{u}_t^{(\theta,\mathbf{w})} \partial \mathbf{u}_t^{(\theta,\mathbf{w})}}, \\ Q_t^{xe} &= \frac{\partial^2 h}{\partial \mathbf{x}_t^{(\theta,\mathbf{w})} \partial \theta}, & Q_t^{ue} &= \frac{\partial^2 h}{\partial \mathbf{u}_t^{(\theta,\mathbf{w})} \partial \theta}\end{aligned} \tag{16}$$

where h denotes the discrete-time Hamiltonian function:

$$\begin{aligned}&h(\mathbf{x}_t^{(\theta,\mathbf{w})}, \mathbf{u}_t^{(\theta,\mathbf{w})}, \lambda_{t+1}; \theta) \\ &= \boldsymbol{\Phi}\left(\mathbf{x}_t^{(\theta,\mathbf{w})}, \mathbf{u}_t^{(\theta,\mathbf{w})}\right) \cdot \mathbf{w} + f(\mathbf{x}_t^{(\theta,\mathbf{w})}, \mathbf{u}_t^{(\theta,\mathbf{w})}, \theta)^T \lambda_{t+1}\end{aligned} \tag{17}$$

As the auxiliary control system has a linear quadratic structure, the state and input trajectories of the auxiliary system can be solved efficiently via Riccati-type recursion [37] and the final result is $\frac{\partial \boldsymbol{\xi}^{(\theta,\mathbf{w})}}{\partial \theta} = \{\bar{\mathbf{x}}_{0:T}^*, \bar{\mathbf{u}}_{0:T-1}^*\}$, which can be directly used to compute the gradient with (6).

IV. Human Subject Experiments

We design a quadrotor simulation environment and human subject experiments to demonstrate how the proposed scheme works interactively with human users.

A. Experiment Design

The quadrotor simulation environment employs a typical scenario; control a quadrotor to safely land it on a touch-pad with the appropriate final position, speed, and attitude. In this work, we develop a two-dimensional quadrotor landing simulator (Fig. 2). The human users are requested to control the quadrotor with a joystick. The human control inputs are related to the roll angle and the total thrust of the quadrotor.

The system dynamics is given as follows with the states consisting of the position, velocity, and attitude of the quadrotor $\mathbf{x}_t = (x_t, y_t, \dot{x}_t, \dot{y}_t, \phi_t, \dot{\phi}_t)^T \in \mathbb{R}^6$, the human control inputs in 2-axis $\mathbf{u}_t = (u_t^x, u_t^y)^T \in \mathbb{R}^2$, and the control parameter in the k-th update $\boldsymbol{\theta}_k = (\theta_{k,1}, \theta_{k,2})^T \in \mathbb{R}^2$:

$$\begin{aligned}T_{\text{total}} &= \alpha u_t^y + b + \theta_{k,1} \dot{y}_t \\ T_{\text{diff}} &= \beta u_t^x + P\phi_t + D\dot{\phi}_t + \theta_{k,2}\dot{x}_t \\ \ddot{x} &= T_{\text{total}} \sin(\phi_t) \\ \ddot{y} &= T_{\text{total}} \cos(\phi_t) - mg \\ \ddot{\phi} &= T_{\text{diff}}/I\end{aligned} \tag{18}$$

where $\alpha, \beta, P, D, b, I$ and m are design parameters and g is a gravitational acceleration. The control parameters $(\theta_{k,1}, \theta_{k,2})^T$ are kinds of D-gain, which help the vertical and horizontal speed be stabilized, respectively. By observations, adding D-gains can effectively assist novices to handle the quadrotor more easily. The proposed scheme can find out how much D-gain values should be given to individual novice human users to assist them to control the quadrotor like an expert. The total 8 quadratic basis features are employed for this testbed, i.e., $\boldsymbol{\Phi}(\mathbf{x}_t, \mathbf{u}_t) = (x_t^2, y_t^2, \dot{x}_t^2, \dot{y}_t^2, \phi_t^2, \dot{\phi}_t^2, (u_t^x)^2, (u_t^y)^2)^T$. For the quadrotor landing simulator, the parameters in Table I are used.

The experiment is conducted by an expert, who is requested to control the quadrotor as stable as possible. This is because in a realistic scenario, landing a quadrotor safely is more critical than landing it quickly. 10 novices are divided into

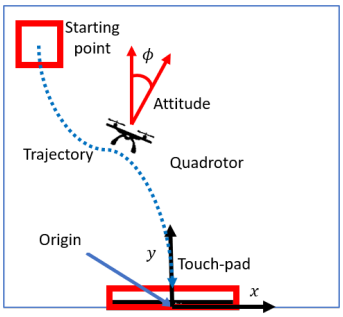

Fig. 2. Experimental testbed: Quadrotor landing simulator

TABLE I
QUADROTOR LANDING SIMULATOR PARAMETERS (SI UNITS)

Parameter	*Value*	*Parameter*	*Value*
$\boldsymbol{\theta}_0$	$(0,0)^T$	$\mathbf{x}_0$	$(-0.28, 0.74)^T$
$\boldsymbol{\Theta}$	$[-0.2, 0], [-0.5, 0]$	$\mathbf{x}_f$	$(0,0)^T$
(α, β)	$(60, 60)$	Δt	0.034 sec
(P, D, b)	$(60, 30, 60)$	(I, m, g)	$(10, 8, 9.8)$

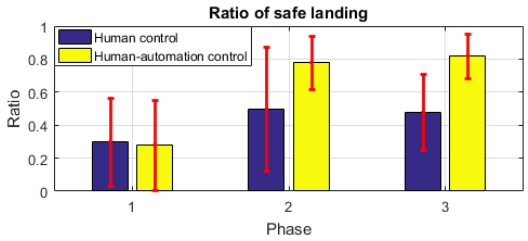

Fig. 3. Ratio of safe landing in three phases with total 10 novice human users. 10 trials for each phase and each novice. Error bars represent the ± 1 standard deviation of the mean.

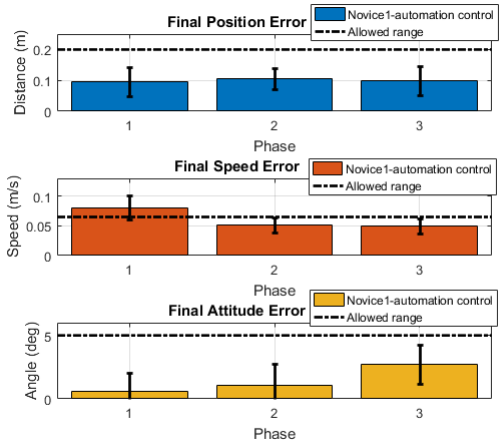

Fig. 4. Final position, speed, and attitude errors for a novice human user (in the human-automation control group) with step size $\eta = 1.0$. In particular, the final speed error is improved with the assistance of the automation. Error bars represent the ± 1 standard deviation of the mean.

two groups: *human control* and *human-automation control*, with 5 members each. The novices are requested to control the quadrotor 10 times for each phase over total 3 phases. Thus, they conduct the experiment 30 times in total. The experiments is conducted under the following conditions, depending on the phase of each group.

- *Phase 1*: no assistance for both groups.
- *Phase 2*: no assistance for the human control group and the automation assists the human-automation control group by iteratively updating control parameters.
- *Phase 3*: no assistance for the human control group and the human-automation control group assisted by the automation with the lastly updated control parameters in phase 2.

Final conditions for a safe landing are given as follows: final position within $0.2m$ radius from the origin (touch-pad), less than $0.065m/s$ final speed, and less than 5 degree roll angle at landing.

B. Results and Discussion

The results show that (Fig. 3) the novices can land the quadrotor safely with the assistance from the automation. This is clear from the fact that the ratio of safe landing for the human-automation control group significantly improves over the phases, while the human control group's performance remains more or less similar.

For investigating individual cases in more detail, Figure. 4 presents the final position, speed, and attitude errors of a novice from the human-automation control group. In the case of this novice, there is no problem in controlling the final position and attitude from the beginning of the experiment, but the final speed control is not satisfactory in Phase 1. However, with the assistance of the automation, the final speed control is remarkably improved from Phase 2, and this trend is maintained in Phase 3.

V. CONCLUSIONS

A skill level based human-automation interaction scheme was proposed and tested by human subject experiments. The results of the experiments demonstrated that the proposed scheme can help novices perform like experts by inferring the experts' task objective function and personalizing the automation's control parameters according to the skill levels of novices. The inverse optimal control and gradient method are employed to infer the task objective function and adjust the control parameters, respectively. The future work will consider uncertainties and variability of human behaviors and develop a theoretical framework which guarantees the system's optimality and stability under uncertainties and variability.

REFERENCES

[1] S. Lee, I. Hwang, and K. Leiden, "Intent inference-based flight-deck human-automation mode-confusion detection," *Journal of Aerospace Information Systems*, vol. 12, no. 8, pp. 503–518, 2015. [Online]. Available: https://doi.org/10.2514/1.I010331

[2] G. D'Intino, M. Olivari, H. H. Bülthoff, and L. Pollini, "Haptic assistance for helicopter control based on pilot intent estimation," *Journal of Aerospace Information Systems*, vol. 17, no. 4, pp. 193–203, 2020. [Online]. Available: https://doi.org/10.2514/1.I010773

[3] S. Islam, R. Ashour, and A. Sunda-Meya, "Haptic and virtual reality based shared control for mav," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 5, pp. 2337–2346, 2019.

[4] M. Flad, J. Otten, S. Schwab, and S. Hohmann, "Steering driver assistance system: A systematic cooperative shared control design approach," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2014, pp. 3585–3592.

[5] P. R. Culmer, A. E. Jackson, S. Makower, R. Richardson, J. A. Cozens, M. C. Levesley, and B. B. Bhakta, "A control strategy for upper limb robotic rehabilitation with a dual robot system," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 4, pp. 575–585, 2009.

[6] G. Aguirre-Ollinger, J. E. Colgate, M. A. Peshkin, and A. Goswami, "Active-impedance control of a lower-limb assistive exoskeleton," in *2007 IEEE 10th international conference on rehabilitation robotics*. IEEE, 2007, pp. 188–195.

[7] B. Khademian and K. Hashtrudi-Zaad, "Dual-user teleoperation systems: New multilateral shared control architecture and kinesthetic performance measures," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 5, pp. 895–906, 2011.

[8] J. Mainprice, R. Hayne, and D. Berenson, "Goal set inverse optimal control and iterative replanning for predicting human reaching motions in shared workspaces," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 897–908, 2016.

[9] Y. Li and S. S. Ge, "Human–robot collaboration based on motion intention estimation," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 1007–1014, 2013.

[10] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," in *Advances in neural information processing systems*, 2016, pp. 3909–3917.

[11] Y. Li, K. P. Tee, W. L. Chan, R. Yan, Y. Chua, and D. K. Limbu, "Continuous role adaptation for human–robot shared control," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 672–681, 2015.

[12] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning robot objectives from physical human interaction," *Proceedings of Machine Learning Research*, vol. 78, pp. 217–226, 2017.

[13] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions." in *Robotics: Science and Systems*, vol. 2. Ann Arbor, MI, USA, 2016.

[14] A. Bajcsy, D. P. Losey, M. K. O'Malley, and A. D. Dragan, "Learning from physical human corrections, one feature at a time," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 141–149.

[15] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2013, pp. 301–308.

[16] J. Mainprice, R. Hayne, and D. Berenson, "Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 885–892.

[17] M. S. Erden and B. Marić, "Assisting manual welding with robot," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 4, pp. 818–828, 2011.

[18] K. P. Tee, R. Yan, and H. Li, "Adaptive admittance control of a robot manipulator under task space constraint," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 5181–5186.

[19] M. Kimmel, M. Lawitzky, and S. Hirche, "6d workspace constraints for physical human-robot interaction using invariance control with chattering reduction," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 3377–3383.

[20] J. Jiang and A. Astolfi, "Shared-control for fully actuated linear mechanical systems," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 4699–4704.

[21] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 2, pp. 286–298, 2007.

[22] R. Luo and D. Berenson, "A framework for unsupervised online human reaching motion recognition and early prediction," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2426–2433.

[23] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 299–306.

[24] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering," in *Proceedings of fifth IEEE international conference on automatic face gesture recognition*. IEEE, 2002, pp. 423–428.

[25] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura, "Incremental learning of full body motion primitives and their sequencing through human motion observation," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 330–345, 2012.

[26] E. Todorov, "Optimality principles in sensorimotor control," *Nature neuroscience*, vol. 7, no. 9, pp. 907–915, 2004.

[27] C. Chow and D. Jacobson, "Studies of human locomotion via optimal programming," *Mathematical Biosciences*, vol. 10, no. 3-4, pp. 239–306, 1971.

[28] R. M. Alexander, "The gaits of bipedal and quadrupedal animals," *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 49–59, 1984.

[29] K. Mombaur, A. Truong, and J.-P. Laumond, "From human to humanoid locomotion—an inverse optimal control approach," *Autonomous robots*, vol. 28, no. 3, pp. 369–383, 2010.

[30] A. Keshavarz, Y. Wang, and S. Boyd, "Imputing a convex objective function," in *2011 IEEE International Symposium on Intelligent Control*. IEEE, 2011, pp. 613–619.

[31] W. Jin, D. Kulić, J. F.-S. Lin, S. Mou, and S. Hirche, "Inverse optimal control for multiphase cost functions," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1387–1398, 2019.

[32] A. Y. Ng and S. Russell, "Algorithms for inverse reinforcement learning," in *Icml*, vol. 1, 2000, pp. 663–670.

[33] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in *Aaai*, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[34] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.

[35] B. Berret, E. Chiovetto, F. Nori, and T. Pozzo, "Evidence for composite cost functions in arm movement planning: an inverse optimal control approach," *PLoS computational biology*, vol. 7, no. 10, 2011.

[36] W. Jin, D. Kulić, S. Mou, and S. Hirche, "Inverse optimal control with incomplete observations," *arXiv preprint arXiv:1803.07696*, 2018.

[37] W. Jin, Z. Wang, Z. Yang, and S. Mou, "Pontryagin differentiable programming: An end-to-end learning and control framework," *arXiv preprint arXiv:1912.12970*, 2019.