2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Modeling User's Driving-Characteristics in a Steering Task to Customize a Virtual Fixture Based on Task-Performance

Han U. Yoon, Ranxiao F. Wang, and Seth A. Hutchinson

Abstract— **This paper presents an approach for modeling user's driving-characteristics in a steering task, and determining the parameters of a virtual fixture to assist the user-control on the basis of his/her task-performances. First, we briefly introduce our assistive human-robot interaction (HRI) interface and a virtual fixture as backgrounds related to this research. The designed HRI interface provides assistance by actively constraining the user-control with a virtual fixture. Second, we discuss a way to model a user's driving-characteristics in a steering task. In modeling the driving-characteristics, we use techniques from inverse optimal control (IOC), where known basis functions (speed, steering, and proximities to inner/outer road boundary) are employed to design a cost function. Third, we describe the experimental setup and procedures to obtain user-demonstrated data from human subjects. Utilizing the obtained data sets, we infer the unknown parameter vector by solving inverse optimal control. Afterward, the user's driving-characteristics are expressed in terms of the balances of the inferred parameters, allowing us to find a relationship between the modeled driving-characteristics and task-completion time. Finally, we present a method to set a virtual fixture for a newly given task by predicting the user's task-performances.**

I. INTRODUCTION

Assistive control is currently prevalent in various applications due to its unique characteristics of constraining/regulating user-driven control input, rendering more freedom to users' control [1]. The assistive control (sometimes together with a haptic guidance) is known to improve user's task-performance, including telerobotic tasks [2], steering task [3], and robot-assisted manipulation [4], and so on. One way of implementing assistive control in human-robot collaborative tasks is through a virtual fixture (also known as active constraint). The virtual fixture is software-generated user-control constraining strategies that enhance a human operator's task performance, and is usually set along with a reference trajectory [4]. In assistive control, setting the proper level of constraint with virtual fixturing is important because an excessive level of constraint for a skilled user would slow down a task completion [5]. In contrast, a lack of constraint for a novice user may lead to a task failure.

For a decade, the inverse optimal control (IOC) has been applied to a broad range of fields, e.g., learning a user's driving style [6], operating an autopilot system that mimics special maneuvers demonstrated by a human pilot [7], finding an optimality principle in human walking [8], determining a user's driving style with continuous model [9], predicting a probabilistic pointing target [10], and so on. Through various applications, IOC has become a powerful tool that grants us an opportunity to scrutinize the following problems: finding the cost function for "flying well" or "driving hastily"? [6]. Namely, the IOC can be utilized in defining ambiguous characteristics, such as "driving well" or "steering well", by inferring the unknown parameter vector of known (predetermined) basis functions with a given user-demonstrated data [6][11].

In this paper, we consider the problems associated with modeling user's driving characteristics in a steering task to set a virtual fixture to assist a user-control on the basis of his/her *task-performances* (task-completion time and the maximum deviation from a reference trajectory). Specifically, we model the driving-characteristics by solving IOC to infer how s/he optimizes the balances of speed, steering, and proximities to inner/outer road boundary with observed user-demonstrations. Next, we find a relationship between the modeled driving-characteristics and task-completion time to determine virtual fixture parameters that will assist the user with a new task. Finally, we illustrate an example of virtual fixturing set by predicting the user's task-performances. In general, the outcomes from this study will provide an answer to a class of questions – "Can we provide the right level of assistance to a user whose driving styles is of certain type?"

The motivation for solving this problem is initiated from the larger problem of developing assistive interfaces for human-robot interaction (HRI) that guides a non-expert user with an expert's knowledge in a steering task. In particular, the modeled driving-characteristics can serve as design parameters to tune the assistive HRI interfaces (including visual feedback, joystick gains, etc). Our focus in this paper is on the methods for setting a virtual fixture to provide a proper level of constraint to the user.

We performed the experiment with human subjects to obtain the user-demonstrated data. The experimental setup consists of a display device and an input device just as the user (the subject) plays a video game. From the beginning to the finish line, the user is instructed to control a mobile robot under various road conditions. Afterward, we solve the IOC problem with the user-demonstrated data to infer the unknown parameter vector for each user. Then, we obtain the modeled user's driving-characteristics in terms of the balances of the inferred parameters, which in turn designates the level of assistance to the user by a statistical analysis.

H. U. Yoon is with Ph.D. Candidate of the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Urbana, IL 61801, USA hyoon24@illinois.edu

R. F. Wang is with Faculty of the Department of Psychology, University of Illinois at Urbana–Champaign, Champaign, IL 61820, USA wang18@illinois.edu

S. A. Hutchinson is with Faculty of the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Urbana, IL 61801, USA seth@illinois.edu

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

Finally, with a new task, the analysis allows us to set a virtual fixture based on the user's predicted task-performance.

This paper is organized in the following order: In Section II, we briefly introduce the background related to this research. In Section III, we discuss how we model user's driving-characteristics using techniques from inverse optimal control. Section IV describes the experimental setup and the task procedures of the human subject experiment to obtain user-demonstrated data. In Section V, we present the modeled driving-characteristics for all users, and a statistical approach to set a virtual fixture based on the predicted task-performances. Finally, we discuss the conclusions and future works in Section VI.

II. BACKGROUNDS RELATED TO THIS RESEARCH

Our assistive HRI interface has been developed to assist a user control by setting active constraint (virtual fixturing) based on a user's task-performance together with providing haptic and visual feedbacks as shown in Fig. 1. Throughout this paper, we will mainly focus on a method for utilizing the modeled driving-characteristics to set a virtual fixture. In this section, therefore, we briefly introduce the backgrounds of virtual fixturing related to this research.

A. Virtual Fixturing and Assisted Control

A virtual fixturing is software-generated collaborative control strategies in order to improve the safety, accuracy, and speed of a human operator in robot-assisted manipulation tasks [4]. In other words, the virtual fixturing is defining *attenuating admittance*, denoted by $c_{\boldsymbol{\delta}^\perp}$, as a function of the deviation from a reference trajectory. From now on, we will refer to the reference trajectory as the *spine* of a virtual fixture.

Fig. 2 illustrates the effect of virtual fixturing along a spine and the resulted assisted control input. Let $\mathbf{x}(t)$ and $\mathbf{x}_r^\dagger$ be current mobile robot position and the closest point on the spine from $\mathbf{x}(t)$ respectively. We define $\mathbf{e}$ as

$$\mathbf{e}(\mathbf{x}(t)) = \mathbf{x}_r^\dagger - \mathbf{x}(t) \tag{1}$$

$\boldsymbol{\delta}$ represents the *preferred direction*, and points toward the direction where $\mathbf{e}$ decreases (For various ways to define $\boldsymbol{\delta}$, see [12]). $\boldsymbol{\delta}^\perp$ denotes the *non-preferred direction*, and is orthogonal to $\boldsymbol{\delta}$.

Following [12][13], we can express an original control input, $\mathbf{f}$, as the summation of the preferred direction component and the non-preferred direction component, thus

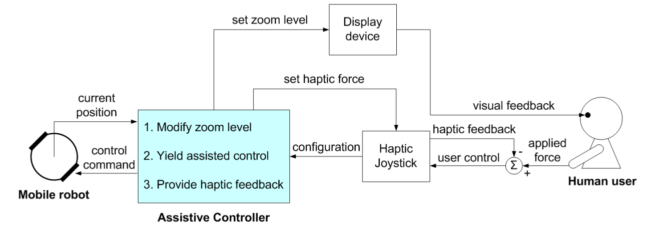


Fig. 1. The developed system architecture of the assistive HRI interface.

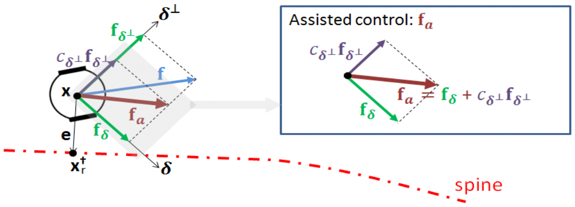


Fig. 2. The effect of virtual fixturing along the spine and the resulted assisted control input.

$$\mathbf{f} = \mathbf{f}_{\boldsymbol{\delta}} + \mathbf{f}_{\boldsymbol{\delta}^\perp} \tag{2}$$

We define an *assisted control* input, $\mathbf{f}_a$, as the summation of the preferred direction component and the attenuated non-preferred direction component

$$\mathbf{f}_a = \mathbf{f}_{\boldsymbol{\delta}} + c_{\boldsymbol{\delta}^\perp}\mathbf{f}_{\boldsymbol{\delta}^\perp} \tag{3}$$

$c_{\boldsymbol{\delta}^\perp}$ is defined in [12] as

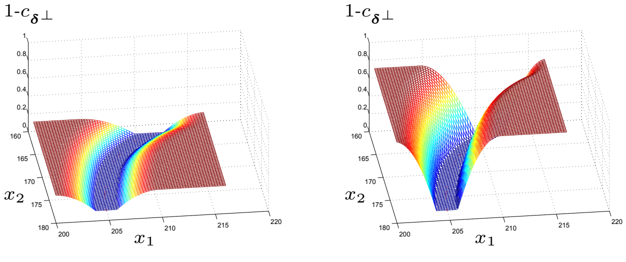
$$c_{\boldsymbol{\delta}^\perp}(\|\mathbf{e}\|) = \begin{cases} \underline{c}_{\boldsymbol{\delta}^\perp}, & \text{if } \|\mathbf{e}\| \geq \frac{d}{2} \\ \underline{c}_{\boldsymbol{\delta}^\perp} + \left[\frac{d/2 - \|\mathbf{e}\|}{\nu}\right]^n (1 - \underline{c}_{\boldsymbol{\delta}^\perp}), & \text{if } \frac{d}{2} - \nu < \|\mathbf{e}\| \leq \frac{d}{2} \\ 1, & \text{if } \|\mathbf{e}\| \leq \frac{d}{2} - \nu \end{cases} \tag{4}$$

where $\frac{d}{2}$ is the half width of a virtual fixture, and ν determines the width of an interval in which $c_{\boldsymbol{\delta}^\perp}$ changes its value from 1 to the lower limit $\underline{c}_{\boldsymbol{\delta}^\perp} < 1$. Note that $\underline{c}_{\boldsymbol{\delta}^\perp}$ and $\frac{d}{2}$ are two parameters that we want to tune based on a user's driving-characteristics and corresponding task-performance.

B. The Level of Assistance

From (4), we know that $\underline{c}_{\boldsymbol{\delta}^\perp}$ is a parameter to determine the maximum level of attenuation. Throughout the paper, the two cases of $\underline{c}_{\boldsymbol{\delta}^\perp} = 0.2$ and $\underline{c}_{\boldsymbol{\delta}^\perp} = 0.8$ will be referred to as *firm assistance* and *soft assistance* respectively [3]. Accordingly, the more a user is skilled the softer assistance is applied to his control input. Oppositely, the less the user is skilled, the firmer assistance is applied.

The effect of virtual fixturing can be clearly seen by plotting "the amount of attenuation" along the spine. Fig. 3 illustrates virtual fixturings for the cases of soft assistance and firm assistance in terms of the amount of attenuation, 1-$c_{\boldsymbol{\delta}^\perp}$, with respect to $\|\mathbf{e}\|$.



(a) soft assistance $\underline{c}_{\boldsymbol{\delta}^\perp} = 0.8$ (b) firm assistance $\underline{c}_{\boldsymbol{\delta}^\perp} = 0.2$

Fig. 3. Soft and firm virtual fixturings in terms of the amount of attenuation, $1 - c_{\boldsymbol{\delta}^\perp}$, with respect to $\|\mathbf{e}\|$.

III. MODELING USER'S DRIVING-CHARACTERISTICS USING THE TECHNIQUES FROM INVERSE OPTIMAL CONTROL

In this section, we introduce a mobile robot kinematics, the way we adopt techniques from inverse optimal control (IOC) to model user's driving-characteristics, and applied numerical method. Throughout the section, we take into account for the specific application in which a user controls a mobile robot along various road shapes with a visual display.

A. Mobile Robot Kinematics in Discrete Time

Fig. 4 illustrates the mobile robot kinematics and the road boundaries at both sides. Let (x_1, x_2) and x_3 are the x-y position and orientation of a mobile robot of unicycle model, respectively. With two control inputs by a user, u_1 (linear velocity) and u_2 (angular velocity), the kinematic equation of the mobile robot in discrete time is [14]

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + t_s \cos x_{3,k} u_{1,k} \\ x_{2,k+1} &= x_{2,k} + t_s \sin x_{3,k} u_{1,k} \\ x_{3,k+1} &= x_{3,k} + t_s u_{2,k} \end{aligned} \tag{5}$$

where t_s and k be a sampling time and corresponding time step respectively.

We assume that the mobile robot is being driven on a road whose inner and outer boundaries correspond to the curves γ_{ib} and γ_{ob} respectively. In our application, these boundaries are identified by an image processing, and represented as a set of discrete points.

B. Discrete Time IOC Formulation: Problem Statement

To model user characteristics in a steering task, we consider a cost function that represents how the user regulates/balances a speed, a steering, and distances to road boundaries. In below, let $p_{\text{ib}} \in \gamma_{\text{ib}}$ and $p_{\text{ob}} \in \gamma_{\text{ob}}$ be the closest points from current mobile robot position $x_{1:2}$ (inspired by [15], we use MATLAB-like expression $x_{1:2}$:=$[x_1, x_2]^T$) to left and right side boundaries respectively.

Consider the following minimization problem [16]

$$\min_{x_k, u_k} t_s \Bigg[\sum_{k=0}^{N-1} u_k^T R u_k + (x_{1:2,k} - p_{\text{ib},k})^T Q_{\text{ib}} (x_{1:2,k} - p_{\text{ib},k}) + (x_{1:2,k} - p_{\text{ob},k})^T Q_{\text{ob}} (x_{1:2,k} - p_{\text{ob},k}) \Bigg]$$

$$\text{subject to} \quad \left.\begin{aligned} x_{1,k+1} &= x_{1,k} + t_s \cos x_{3,k} u_{1,k} \\ x_{2,k+1} &= x_{2,k} + t_s \sin x_{3,k} u_{1,k} \\ x_{3,k+1} &= x_{3,k} + t_s u_{2,k} \end{aligned}\right\} (\blacktriangle)$$
$$x_0 = x_{\text{start}}, \quad x_N = x_{\text{free}} \tag{6}$$

where

$$R = \begin{bmatrix} c_v & 0 \\ 0 & c_\omega \end{bmatrix}, \quad Q_{\text{ib}} = \begin{bmatrix} c_{\text{ib}} & 0 \\ 0 & c_{\text{ib}} \end{bmatrix} = c_{\text{ib}} I, \quad \text{and} \quad Q_{\text{ob}} = \begin{bmatrix} c_{\text{ob}} & 0 \\ 0 & c_{\text{ob}} \end{bmatrix} = c_{\text{ob}} I \tag{7}$$

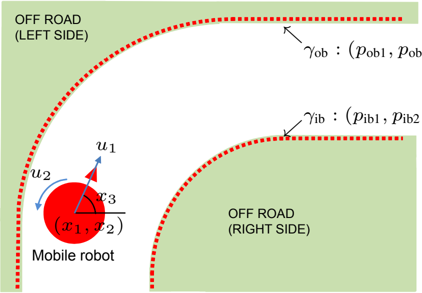

OFF ROAD
(LEFT SIDE)
$\gamma_{\text{ob}} : (p_{\text{ob1}}, p_{\text{ob2}})$
$\gamma_{\text{ib}} : (p_{\text{ib1}}, p_{\text{ib2}})$
u_1
u_2
x_3
(x_1, x_2)
Mobile robot
OFF ROAD
(RIGHT SIDE)

Fig. 4. The mobile robot kinematics and the road boundaries at both sides.

The subscripted parameters c_v, c_ω, c_{ib}, and c_{ob} represent the parameters related to speed, steering, proximity to the inner side, and proximity to the outer side, respectively.

Now, suppose that we already have a user-demonstrated data, which is a set of tuple (x_k, u_k). Furthermore, assume that the user-demonstrated data is a (locally) optimal solution of (6); thus (x_k^*, u_k^*) is what we have. Then, the IOC problem is to infer the unknown parameters, c_v, c_ω, c_{ib}, and c_{ob}, in matrices R, Q_{ib}, and Q_{ob} with the given (x_k^*, u_k^*) [9][17].

C. Applied Numerical Method

Now, to solve IOC problem of (6), we apply the method developed in [17]. Here, we recapitulate that method, as it applied to our specific cost function. First, we rewrite the above kinematic equation (▲) for clarity as

$$x_{k+1} = x_k + t_s f(x_k, u_k)$$

then a discrete time Hamiltonian H_k can be defined [18][19]

$$H_k(x_k, u_k, \lambda_{k+1}) = L(x_k, u_k) + \lambda_{k+1}^T f(x_k, u_k) \tag{8}$$

where L and $\lambda \in \mathbb{R}^3$ represent Lagrangian and a costate vector respectively. From (6), we have

$$H_k = \Big[u_k^T R u_k + (x_{1:2,k} - p_{\text{ib},k})^T Q_{\text{ib}} (x_{1:2,k} - p_{\text{ib},k}) + (x_{1:2,k} - p_{\text{ob},k})^T Q_{\text{ob}} (x_{1:2,k} - p_{\text{ob},k}) \Big] + \lambda_{k+1}^T f(x_k, u_k) \tag{9}$$

By applying the maximum principle [20][21], we have

$$\frac{\partial H_k}{\partial x_k} = \begin{bmatrix} 2(x_{1,k} - p_{\text{ib1},k}) c_{\text{ib}} + 2(x_{1,k} - p_{\text{ob1},k}) c_{\text{ob}} \\ 2(x_{2,k} - p_{\text{ib2},k}) c_{\text{ib}} + 2(x_{2,k} - p_{\text{ob2},k}) c_{\text{ob}} \\ -\lambda_{1,k+1} \sin x_{3,k} u_{1,k} + \lambda_{2,k+1} \cos x_{3,k} u_{1,k} \end{bmatrix} = -\frac{\lambda_{k+1} - \lambda_k}{t_s} \tag{10}$$

Rearranging (10) yields

$$\begin{bmatrix} 2t_s(x_{1,k} - p_{\text{ib1},k}) c_{\text{ib}} + 2t_s(x_{1,k} - p_{\text{ob1},k}) c_{\text{ob}} \\ 2t_s(x_{2,k} - p_{\text{ib2},k}) c_{\text{ib}} + 2t_s(x_{2,k} - p_{\text{ob2},k}) c_{\text{ob}} \\ t_s(-\lambda_{1,k+1} \sin x_{3,k} u_{1,k} + \lambda_{2,k+1} \cos x_{3,k} u_{1,k}) \end{bmatrix} + \begin{bmatrix} \lambda_{1,k+1} - \lambda_{1,k} \\ \lambda_{2,k+1} - \lambda_{2,k} \\ \lambda_{3,k+1} - \lambda_{3,k} \end{bmatrix} = \mathbf{0}^T \tag{11}$$

We also know

$$\frac{\partial H_k}{\partial u_k} = \begin{bmatrix} 2u_{1,k}c_v + (\lambda_{1,k+1}\cos x_{3,k} + \lambda_{2,k+1}\sin x_{3,k}) \\ 2u_{2,k}c_\omega + \lambda_{3,k+1} \end{bmatrix} = \mathbf{0}^T \tag{12}$$

Finally, if we define a vector z_k as

$$z_k = [c_v\ c_\omega\ c_{\text{ib}}\ c_{\text{ib}}\ c_{\text{ob}}\ c_{\text{ob}}\ \lambda_{1,k}\ \lambda_{2,k}\ \lambda_{3,k}\ \lambda_{1,k+1}\ \lambda_{2,k+1}\ \lambda_{3,k+1}]^T \tag{13}$$

then (11) and (12) can be combined and rewritten by the form

$$\begin{bmatrix} A_{11,k} & A_{12,k} & A_{13,k} & A_{14,k} \\ A_{21,k} & A_{22,k} & A_{23,k} & A_{24,k} \end{bmatrix} z_k := A_k z_k \tag{14}$$

The submatrices in the first row are

$$\begin{aligned} A_{11,k} &= O^{3\times 2} \\ A_{12,k} &= \begin{bmatrix} 2t_s(x_{1,k} - p_{\text{ib}1,k}) & 0 & 0 \\ 0 & 2t_s(x_{2,k} - p_{\text{ib}2,k}) & 0 \\ 2t_s(x_{1,k} - p_{\text{ob}1,k}) & 0 & 0 \\ 0 & 2t_s(x_{2,k} - p_{\text{ob}2,k}) & 0 \end{bmatrix}^T \\ A_{13,k} &= -I^{3\times 3} \\ A_{14,k} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -t_s u_{1,k}\sin x_{3,k} & t_s u_{1,k}\cos x_{3,k} & 1 \end{bmatrix} \end{aligned} \tag{15}$$

and in the second row

$$\begin{aligned} A_{21,k} &= \begin{bmatrix} 2u_{1,k} & 0 \\ 0 & 2u_{2,k} \end{bmatrix} \\ A_{22,k} &= O^{2\times 4} \\ A_{23,k} &= O^{2\times 3} \\ A_{24,k} &= \begin{bmatrix} \cos x_{3,k} & \sin x_{3,k} & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{16}$$

where $O^{m\times n}$ and $I^{m\times n}$ represent m-by-n zero matrix and identity matrix respectively. Hence, with the given (x_k^*,u_k^*), the problem of inferring the unknown parameters, $c_v, c_\omega, c_{\text{ib}}, c_{\text{ob}}$ (which are involved in z_k), becomes identical to solve the following least square problem

$$\min_{z_k} \quad \|A_k^* z_k\|^2 \tag{17}$$

where A_k^* represents the matrix A_k being evaluated at (x_k^*,u_k^*).

IV. EXPERIMENT WITH HUMAN SUBJECTS TO OBTAIN USER-DEMONSTRATED DATA

In this section, we describe the experimental setup and procedure of the human subject experiment. The objective of the experiment is to obtain the user-demonstrated data that will be utilized as (x_k^*,u_k^*) for the IOC problem.

A. Experimental Setup

A user (a human subject) was provided with an input device and a display just as he played a video game. We used a game controller with two analog-sticks, one for speed and the other for steering, as the input device. Typical 17-inch monitor was used as a display device. Fig. 5 shows our simulator interface provided to a user. The interface window size was 8.89mm×17.78mm (height×width) approximately in 1280×1024 display resolution, and provided a global view and a local view of the environment for the steering task. The other area of the monitor was filled with uniform gray color to prevent distractions caused by background contents.

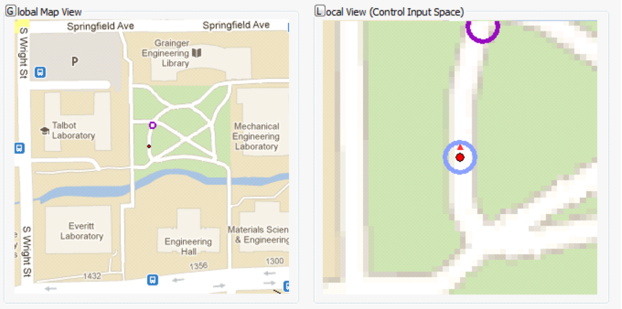


Fig. 5. Simulator interface: a global view (left window) and a local working field of view (right window). Two circles in the middle and at the top of the right window represent the start and the goal respectively.

B. Task Procedure

The experiment was divided into a practice stage and a testing stage. During the practice stage, the user was asked to familiarize him/herself with the user interface by driving a mobile robot around either in- or out-side of the road boundaries. However, we instructed the user that it would be regarded as a task failure to drive the mobile robot out-side of the road boundary during the real task. The real task started when the user verbally expressed that he/she felt familiar with the apparatus and confident with the task.

During the real task, the user was supposed to drive the mobile robot from the starting location to the goal. The simulator provided 4-different task-locations wherein roads had different curvatures and turning angles. Each task was repeated 5-times (trials), then the user proceeded to the next task. Among the 5-trials, the user data that showed the best performance (thus, the fastest one in task-completion time) was used as (x^*, u^*) for a given task. There was a 3-second interval between every trial. While the user was performing the given task, a program read the position and the orientation of the mobile robot every 20-millisecond, and stored the position, orientation, two control inputs, task number, trial number, and task result (success or failure).

V. RESULTS

A. Inferred Unknown Parameter Vector by Solving IOC

Let $D_j : \{(x_k^*, u_k^*)\}_{k=1}^{N_j}$ be a set of user-demonstrated data for task #j, where N_j is task-completion and j=1, $\cdots$, 4 (for 4-different task-locations). We solve the IOC problem of multiple demonstrations by applying the method introduced in Section III-C (Also, see [17] for IOC with multiple user-demonstrated trajectories).

$[c_v, c_\omega, c_{\text{ib}}, c_{\text{ob}}]$ in Table I shows the inferred parameters by solving the IOC. Recalling the cost function in (6), we know that the balance of these parameters is indeed

TABLE I
RESULT TABLE FOR SECTION V

GROUP	USER#	$[\ c_v, c_\omega, c_{\text{ib}}, c_{\text{ob}}\ ]$	$\sum N_j$	$\boldsymbol{z}_i = [\ c_\omega/c_v,\ c_{\text{ib}}/c_{\text{ob}}\ ]$	r	r_{avg}	$\underline{c}_{\boldsymbol{\delta}^\perp}(r_{\text{avg}})$
$\mathcal{G}_1$	USER11	[0.2329, 0.6693, 1.1646, 1.000]	458	$\boldsymbol{z}_{11}$ = [2.8738, 1.1646]	1.8813	1.38	0.8
	USER5	[0.2198, 0.5901, 1.0989, 1.000]	461	$\boldsymbol{z}_5$ = [2.6844, 1.0989]	1.0717		
	USER3	[0.2117, 0.4366, 1.0585, 1.000]	467	$\boldsymbol{z}_3$ = [2.0622, 1.0585]	1.2148		
	USER6	[0.2091, 0.4322, 1.0454, 1.000]	488	$\boldsymbol{z}_6$ = [2.0670, 1.0454]	1.2140		
	USER12	[0.2314, 0.7383, 1.1569, 1.000]	495	$\boldsymbol{z}_{12}$ = [3.1908, 1.1569]	1.5370		
$\mathcal{G}_2$	USER2	[0.2000, 0.4456, 0.8075, 1.000]	516	$\boldsymbol{z}_2$ = [2.2279, 0.8075]	15.8932	24.33	0.5
	USER7	[0.5497, 0.8420, 1.1868, 1.000]	517	$\boldsymbol{z}_7$ = [1.5316, 1.1868]	11.7397		
	USER10	[0.2033, 1.6280, 1.0166, 1.000]	551	$\boldsymbol{z}_{10}$ = [8.0068, 1.0166]	41.1482		
	USER8	[0.2000, 1.1228, 0.9609, 1.000]	554	$\boldsymbol{z}_8$ = [5.6140, 0.9609]	28.5570		
$\mathcal{G}_3$	USER9	[0.2009, 0.2775, 0.8903, 1.000]	575	$\boldsymbol{z}_9$ = [1.3812, 0.8903]	6.1638	4.67	0.2
	USER1	[0.2000, 0.2011, 0.9767, 1.000]	594	$\boldsymbol{z}_1$ = [1.0055, 0.9767]	4.0564		
	USER4	[0.2000, 0.2533, 0.9297, 1.000]	657	$\boldsymbol{z}_4$ = [1.2663, 0.9297]	3.8113		

$[c_v, c_\omega, c_{\text{ib}}, c_{\text{ob}}]$: inferred parameters, $\sum N_j$: task-completion time, $\boldsymbol{z}_i$: sample point, r: Mahalanobis distance, r_{avg} : average r in $\mathcal{G}$, and $\underline{c}_{\boldsymbol{\delta}^\perp}(r_{\text{avg}})$: the desired $\underline{c}_{\boldsymbol{\delta}^\perp}$ value at r_{avg}.
$*\, c_{\text{ob}}$ is normalized to 1. As we solve (17), the IOC solver returns trivial solution $[0, 0, 0, 0]$ unless we normalize one of $[c_v, c_\omega, c_{\text{ib}}, c_{\text{ob}}]$.

related to a curvature and a route of the user-demonstrated trajectory, which in turn determines the task-completion time and the maximum deviation from a spine. Therefore, if we define the driving-characteristics of a user by the inferred parameters, then it allows us to estimate his/her predicted task-performance for newly given tasks. We will further discuss it in the following sections.

B. Driving-Characteristics and a Metric for Skill Level

Due to the limited number of users, i.e., we have 12 user-demonstrations, here we present an example of our methodology to find a relationship between the driving-characteristics and the task-completion time. Note that the presented statistical approach is developed for this limited case.

1) Represent a user as a sample point: First of all, we express the user #i as a sample point in $\mathbb{R}^2$, denoted by $\boldsymbol{z}_i$, with the driving-characteristics that is defined as follows:

c_ω/c_v : how the user controls a steering to a speed
$c_{\text{ib}}/c_{\text{ob}}$: proximity to inner boundary over outer boundary

$\boldsymbol{z}_i = [\ c_\omega/c_v, c_{\text{ib}}/c_{\text{ob}}\ ]$ in Table I shows the expressed sample points.

2) Mahalanobis distance as a metric to represent difference from a skilled user group: Based on task-completion time, we classify the samples into three groups, i.e., a skilled group $\mathcal{G}_1$:$\{\boldsymbol{z}_{11}, \boldsymbol{z}_5, \boldsymbol{z}_3, \boldsymbol{z}_6, \boldsymbol{z}_{12}\}$, an intermediate group $\mathcal{G}_2$:$\{\boldsymbol{z}_2, \boldsymbol{z}_7, \boldsymbol{z}_{10}, \boldsymbol{z}_8\}$, and a novice group $\mathcal{G}_3$:$\{\boldsymbol{z}_9, \boldsymbol{z}_1, \boldsymbol{z}_4\}$. Then, we assign a weight, $(\sum N_j)^{-1}$, to the samples in $\mathcal{G}_1$, e.g., $\boldsymbol{w}_{11}=\frac{1}{458}$. From the five weighted samples, we calculate mean $\boldsymbol{\mu}_{\mathcal{G}_1}$ and covariance $\boldsymbol{\Sigma}_{\mathcal{G}_1}$. With $\boldsymbol{\mu}_{\mathcal{G}_1}$ and $\boldsymbol{\Sigma}_{\mathcal{G}_1}$, for all $\boldsymbol{z}_i$, we calculate the Mahalanobis distance r and the group's average r_{avg} as shown in Table I (the third and the second column from the right, respectively).

C. Setting a Virtual Fixture for a New Task

Recall that, in Section II-B, we have introduced $\underline{c}_{\boldsymbol{\delta}^\perp}$ (the lower limit of $c_{\boldsymbol{\delta}^\perp}$) and $\frac{d}{2}$ (a half width of virtual fixture) as two parameters that we want to adjust based on a user's driving-characteristics and corresponding task-performance.

1) Define $\underline{c}_{\boldsymbol{\delta}^\perp}$ as a function of r: From r and $\sum N_j$ in Table I, we can see that the task-completion time is neither monotonically increasing nor monotonically decreasing with respect to r. In this case, one way to define $\underline{c}_{\boldsymbol{\delta}^\perp}$ as a function of r is that we set the following three $(r_{\text{avg}}, \underline{c}_{\boldsymbol{\delta}^\perp}(r_{\text{avg}}))$-points for the groups

$$\mathcal{G}_1 : (1.38, 0.8),\ \mathcal{G}_3 : (4.67, 0.2),\ \text{and}\ \mathcal{G}_2 : (24.33, 0.5)$$

which consist of group's r_{avg} and desired $\underline{c}_{\boldsymbol{\delta}^\perp}$ value. Then, we separately perform two curve fittings to find continuous functions that connect $(1.38, 0.8)$ to $(4.67, 0.2)$ and $(4.67, 0.2)$ to $(24.33, 0.5)$, respectively; it yields

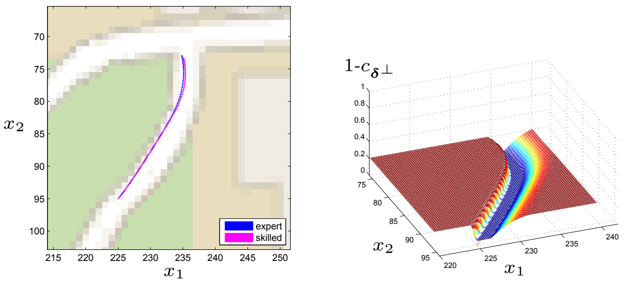
$$\underline{c}_{\boldsymbol{\delta}^\perp}(r) = \begin{cases} 0.8, & \text{if } 1.07 < r \leq 1.38 \\ 0.0554r^2 - 0.5177r + 1.4089, & \text{if } 1.38 < r < 4.67 \\ 0.2, & \text{if } r = 4.67 \\ 0.0008r^2 - 0.0072r + 0.2169, & \text{if } 4.67 < r \leq 24.33 \\ 0.5, & \text{if } 24.33 < r \leq 41.15 \end{cases} \tag{18}$$

We note that the smallest and the largest r values in Table I are $1.0717 \simeq 1.07$ and $41.1482 \simeq 41.15$, respectively.

2) Define $\frac{d}{2}$ to be a function of the maximum deviation: We begin by solving (forward) optimal control problem with the expert's and the user's $[c_v, c_\omega, c_{\text{ib}}, c_{\text{ob}}]$ to generate the spine and the user's predicted trajectories for a new task, respectively. After generating the spine and the predicted trajectories, we calculate the maximum deviation between them, $\mathbf{e}_{\max}$, and define $\frac{d}{2}$ to be

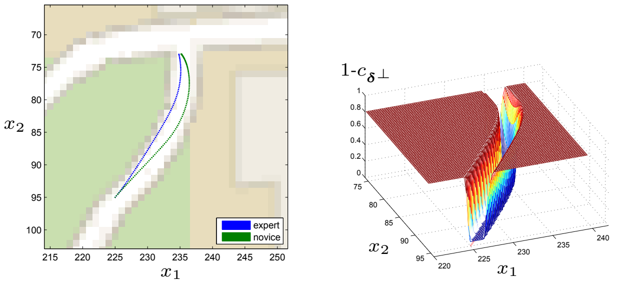
$$\frac{d}{2}(\mathbf{e}_{\max}) = \beta W \mathbf{e}_{\max}^{-1} \tag{19}$$

where β is a positive constant, and W is a road width assumed to be consistent during the discrete time interval $[1, N_j]$. If the resulted virtual fixture goes outside of either inner or outer boundary, we set $\frac{d}{2}$ to be the closest distance to the inner/outer boundary from the spine.



(a) USER5's predicted trajectories (b) virtual fixturing for USER5

Fig. 6. Predicted trajectory and virtual fixturings for USER5. Parameters are $\underline{c}_{\delta\perp} = 0.8$ (thus $1 - \underline{c}_{\delta\perp} = 0.2$) and $\frac{d}{2} \simeq 5$.



(a) USER1's predicted trajectories (b) virtual fixturing for USER1

Fig. 7. Predicted trajectory and virtual fixturings for USER1. Parameters are $\underline{c}_{\delta\perp} = 0.2205$ (thus $1 - \underline{c}_{\delta\perp} = 0.7795$) and $\frac{d}{2} \simeq 2$.

Finally, recalling that our assistive HRI system has been developed to guide the novice user with the expert's knowledge in a steering task, we pick USER11 as the expert to generate the spine for virtual fixturing. Fig. 6 and Fig. 7 illustrate the examples of virtual fixturing for USER5 and USER1 by (18) and (19).

VI. Conclusions and Future Works

In this paper, we consider the problem of modeling user's driving characteristics in a steering task to customize a virtual fixture to assist the user-control based on his/her task-performances. We performed the human subjects experiment, and inferred the unknown parameter vectors by solving inverse optimal control problem. Then, we modeled the user's driving-characteristics in terms of the balances of the inferred parameters, defined the driving-characteristics and the metric for skill level, and defined the parameters to set a virtual fixture by predicting the user's task performance. By identifying user's driving-characteristics by our approach, we could get an insight into how to tune our designed HRI interface to assist a user of certain characteristics. The resulting insight could be utilized to determine not only a way of assistance but also the level of assistance.

For future works, first, we should increase the size of human subject pool, and find more robust/general classification method to overcome the limitations of the presented method, i.e., it was not so straightforward to define the level of assistance as a function of the defined metric. Second, we should verify whether virtual fixturing actually improves the user's task-performance by human subjects experiment. Finally, we should compare our approach to other established techniques, e.g., potential field, and analyze the results.

References

[1] S. A. Bowyer, B. L. Davies, and F. R. y. Baena, "Active constraints/virtual fixtures: A survey," *to appear, IEEE Transactions on Robotics*, 2013.

[2] A. Dragan and S. Srinivasa, "Formalizing assistive teleoperation," *Robotics: Science and Systems*, 2012.

[3] L. Marchal-Crespo, S. McHughen, S. C. Cramer, and D. J. Reinkensmeyer, "The effect of haptic guidance, aging, and initial skill level on motor learning of a steering task," *Experimental Brain Research*, vol. 201, no. 2, pp. 209–220, 2010.

[4] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *Robotics research*. Springer, 2007, pp. 49–64.

[5] C. Passenberg, R. Groten, A. Peer, and M. Buss, "Towards real-time haptic assistance adaptation optimizing task performance and human effort," in *World Haptics Conference (WHC), 2011 IEEE*, June 2011, pp. 155–160.

[6] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the 21st International Conference on Machine Learning*. ACM Press, 2004.

[7] A. Coates, P. Abbeel, and A. Y. Ng, "Learning for control from multiple demonstrations," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 144–151.

[8] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz, "An optimality principle governing human walking," *Robotics, IEEE Transactions on*, vol. 24, no. 1, pp. 5–14, feb. 2008.

[9] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," in *ICML '12: Proceedings of the 29th International Conference on Machine Learning*, 2012.

[10] B. Ziebart, A. Dey, and J. A. Bagnell, "Probabilistic pointing target prediction via inverse optimal control," in *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, 2012, pp. 1–10.

[11] S. Levine, Z. Popovic, and V. Koltun, "Feature construction for inverse reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[12] A. Bettini, P. Marayong, S. Lang, A. M. Okamura, and G. D. Hager, "Vision-assisted control for manipulation using virtual fixtures," *Robotics, IEEE Transactions on*, vol. 20, no. 6, pp. 953–966, 2004.

[13] J. J. Abbott and A. M. Okamura, "Pseudo-admittance bilateral telemanipulation with guidance virtual fixtures," *The International Journal of Robotics Research*, vol. 26, no. 8, pp. 865–884, 2007.

[14] S. Mastellone, D. Stipanovic, and M. Spong, "Remote formation control and collision avoidance for multi-agent nonholonomic systems," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2007, pp. 1062–1067.

[15] P. Abbeel and A. Y. Ng, "Exploration and apprenticeship learning in reinforcement learning," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 1–8.

[16] A.-S. Puydupin-Jamin, M. Johnson, and T. Bretl, "A convex approach to inverse optimal control and its application to modeling human locomotion," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 531–536.

[17] M. Johnson, "Inverse optimal control for deterministic nonlinear systems," *Thesis*, University of Illinois at Urbana-Champaign, 2013.

[18] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I, 3rd Ed.* Belmont, MA: Athena Scientific, 2005.

[19] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *Circuits and Systems Magazine, IEEE*, vol. 9, no. 3, pp. 32–50, 2009.

[20] H. J. Sussmann and J. C. Willems, "300 years of optimal control: from the brachystochrone to the maximum principle," *Control Systems, IEEE*, vol. 17, no. 3, pp. 32–44, 1997.

[21] D. Liberzon, *Calculus of variations and optimal control theory: A concise introduction*. Princeton University Press, 2012.