Robotics and Autonomous Systems 91 (2017) 258–269

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Customizing haptic and visual feedback for assistive human–robot interface and the effects on performance improvement

Han U. Yoon [a], Ranxiao F. Wang [b], Seth A. Hutchinson [c], Pilwon Hur [a,*]

[a] Department of Mechanical Engineering, Texas A&M University, 3123 TAMU, College Station, TX 77843-3123, USA
[b] Department of Psychology, University of Illinois at Urbana–Champaign, 603 E Daniel St, Champaign, IL 61820, USA
[c] Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, 306 N Wright St, Urbana, IL 61801, USA

HIGHLIGHTS

- We propose a novel approach to customizing haptic and visual feedback.
- Geometric/behavioral characteristics of user control were extracted from user data.
- Using the virtual fixturing, we customized haptic and visual assistances.
- Human subject experiments were conducted to verify the proposed approach.
- The user's performance significantly enhanced with the customized haptic assistance.

ARTICLE INFO

Article history:
Available online 3 February 2017

Keywords:
Assistive human–robot interface
Assistance customization
Featuring user's control strategy
Inverse optimal control
Virtual fixturing

ABSTRACT

This paper presents an approach to developing an assistive interface for human–robot interaction that provides users with customized haptic and visual feedback. The developed interface targets to improve users' task performance by customizing assistance policy and level based on users' performance and control strategy. To achieve this, the users' control strategy was modeled based on inverse optimal control technique. Then, features describing the geometric and behavioral characteristics of user control are derived. Finally, an expert whose features most closely matched each user was identified. The identified expert was assigned to the user to define and provide customized assistance via a virtual fixturing. In human subject experiments, control strategies of twenty-three users were identified and featured; their performance was measured with four assistance types (no-assist, haptic assistance, visual zooming assistance, haptic assistance + visual zooming assistance) × two parameterization types (customized, non-customized). By analyzing the experimental data, we found an optimal combination of assistance type × parameterization type that results in the most improvement of performance. The results showed that the users' task completion time and mean required effort yielded the best improvements when haptic assistance (with no visual zooming assistance) × customized parameterization were provided for mobile robot driving tasks.

1. Introduction

Assistive interfaces are currently prevalent in various applications due to their unique characteristics of guiding user-driven control input and reducing the user's burden for fine control [1]. For about two decades, they have shown feasibility for user's task performance improvement including telerobotic tasks [2], steering tasks [3,4], robot-assisted manipulation [5,6], assistive medical devices and telesurgery [7–10], etc. Previous studies also reported that human task performance could be enhanced by blending two control inputs from a human user and an assistive system. These blending techniques are often referred to as various names such as shared control [11–13], control blending [2,14], assistive/cooperative control [6,15], and so on.

Accordingly, abundant applications of assistive interfaces exist to provide mobility related assistance [16]. To assist a user for a power wheelchair control, researchers conducted several studies. Marchal-Crespo et al. developed a wheelchair trainer system [17]. Urdiales et al. introduced a wheelchair collaborative control for indoor navigation [18]. Carson et al. proposed a collaborative smart power wheelchair [15]. Simpson et al. introduced a hephaestus smart wheelchair assistance which used a belief histogram of

* Corresponding author.
E-mail addresses: hanulyoon@tamu.edu (H.U. Yoon), wang18@illinois.edu (R.F. Wang), seth@illinois.edu (S.A. Hutchinson), pilwonhur@tamu.edu (P. Hur).

http://dx.doi.org/10.1016/j.robot.2017.01.015
0921-8890/

obstacle location for speed control [19].[1] For car-driving assistance, Storms et al. proposed a model predictive controller based approach [14]. Jensen et al. addressed a method to utilize a haptical sensation [21] whereas Griffiths and Gillespie proposed a shared control approach via a haptic steering wheel [11]. In all these applications, it was a critical issue to define an assistance policy (which means a way of blending a human user's intention and an assistive machine's intention and providing guidance control) and an assistance level (the intensity of guidance). For instance, a human user may experience severe difficulty controlling a smart power wheelchair in a preferred direction or at the desired speed when an assistance policy and an assistance level were set improperly.

Aforementioned aspects in assistive interface design claimed that "an excessive assistance for a skilled user would slow down task completion, whereas a lack of assistance for a novice user may cause a task failure [17,18]". These aspects initiated techniques to adapt the assistance policy and the assistance level based on user's skill level, such as "assistance-as-needed" [22] followed by "guidance-as-needed" [3,4], "virtual teacher" [12,23], "progressive guiding torque" [24], etc. Selected works in the current state of the art consider assistance adaptation techniques. For instance, Passenberg et al. reported a method that utilized a deviation from the desired trajectory as a metric to vary an assistance level in real time [25,26]. Abbink et al. introduced the smooth shifting of haptic control authority [13]. Smisek et al. proposed physiology-based approaches by considering user grip force [27] and neuromuscular impedance in [28]. Fisher et al. considered user's behavioral model [29]. Although there exist ample approaches, an assistance adaptation technique based on both the user's control strategy and performance has not yet been fully considered for assistive interfaces to provide improved mobility. Furthermore, most of the existing frameworks cannot compatibly/flexibly cope with selecting a reference trajectory or the intensity of assistance with respect to a wide variety of user's skill level and behavioral characteristics.

To mitigate these issues, we proposed an approach to customizing haptic and visual feedback based on the user's control strategy as well as user performance for the custom-designed assistive human–robot interaction (HRI) interface. In our approach, we first model user's control strategy as a cost function by using inverse optimal control (IOC) [30,31]. Then, we define characteristic features that describe the geometric and behavioral characteristics of user control. Finally, the features will serve as parameters to set an assistance policy and an assistance level to provide the customized haptic and visual feedback. Specifically, the parameters obtained from the characteristic features of the user control will be utilized to set a virtual fixture similar to constructing a virtual convex-shaped hull along the desired trajectory. The purpose of the virtual hull (i.e., wall) is to constrain the position of a controlled object in the vicinity of the desired trajectory [32]. Accordingly, the assistance via haptic and visual feedback will be modulated based on the slope of the virtual well.

To our best knowledge, no studies have explicitly addressed these issues. Fisher et al. considered user's behavioral pattern only within a constrained task space [29]. Kucukyilmaz and Demiris's approach is the closest example where a human expert simultaneously demonstrate his strategy while a novice is performing a given task [33]. However, direct assistance by an expert's demonstration can cause degraded user performance due to expert's fatigue or boredom.

To validate our proposed approach, we performed human subject experiments. In the experiments, users were given several tasks for which they control a mobile robot to a goal position while watching a monitor display in a virtual environment. These tasks were adopted for the potential applications where a rider drives an assistive power wheelchair while monitoring Google Maps or Bing Maps through a tablet. Statistical analysis was performed to explain the effects of assistance types and parameterization types (either positive or negative) on user performance improvement. While analyzing experimental data, the following research questions were addressed: "how do assistance types affect user's performance?", "how does the customized parameterization affect user's performance?", and "what is the optimal combination of the types of assistance and parameterization for the best user performance?"

The rest of the paper is organized as follows: Section 2 presents an assistive HRI interface platform, guidance-based virtual fixture, and mathematical descriptions for setting both haptic and visual assistance policies. Section 3 discusses an approach to customizing assistance policy and level. Section 4 introduces the design of the human subject experiment. The experimental results are discussed in Section 5, followed by discussion in Section 6. Section 7 addresses the conclusion of this study.

2. Assistive human–robot interface with haptic guidance and visual zooming

2.1. Interface architecture

The developed assistive HRI interface platform is illustrated in Fig. 1. The platform is equipped with a haptic joystick (input device), a display monitor, and a custom-designed simulator application with a virtual mobile robot. To control the mobile robot, a user interacts with the simulator application while receiving haptic and visual feedback from the haptic joystick and the display monitor, respectively. For human subject experiment, the platform provides several tasks where a user controls a mobile robot along a road with various scenarios.

A *haptic and visual feedback controller* performs functional key roles (see the inside of the programmed simulator application in Fig. 1). For the human user, the haptic and visual feedback controller modifies haptic force $\boldsymbol{f}_{\text{haptic}}$ and a zoom level α. It generates *guiding force* $\boldsymbol{f}_{\text{g}}$ which becomes $\boldsymbol{f}_{\text{haptic}}$ at the user's hand. To enhance the user's task performance, this guiding force is transferred to the mobile robot controller and blended with a control force $\boldsymbol{f}$ generated by a user input. All of these key roles depend on a virtual fixturing along the road (right-top corner in Fig. 1). The virtual fixturing is a set of three parameters: a *reference trajectory* (γ), a width of virtual fixturing (d), and *the maximum level of assistance* ($1 - \underline{\kappa}_{\delta^\perp}$) that indicates the maximum level of assistance that a user will receive from a haptic feedback system (i.e., the portion of control force $\boldsymbol{f}$ that will be blended with the haptic system). These three parameters are targeted to be customized for each user by the proposed assistance customization algorithm in Section 3.

2.2. Setting haptic and visual assistances via a guidance-based virtual fixturing

In our assistive HRI interface, an assistance policy is set by a virtual fixturing. Among various types of virtual fixturing, we adopt a guidance-based virtual fixturing (GVF) introduced in [32]. For the developed interface, the adopted GVF plays a key role in setting haptic and visual assistance.

2.2.1. Guidance-based virtual fixturing

GVF refers to defining a function along the desired trajectory to assist a user in moving a controlled object along the desired

[1] [20] provides a valuable historical survey of a smart wheelchair.

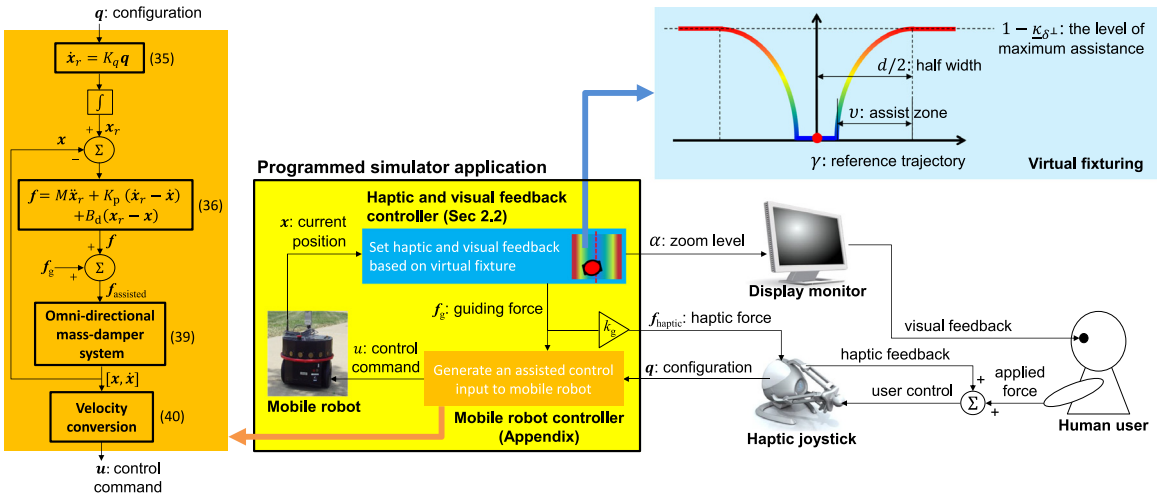


Fig. 1. Assistive HRI interface platform and the three virtual fixture parameters: the level of maximum assistance ($1-\underline{\kappa}_{\delta^\perp}$), width ($d$), and reference trajectory (γ). The proposed algorithm will explain a method to customize these three parameters for each user.

trajectory [5]. Typically, the function for GVF is monotonically non-decreasing as the controlled object deviates further from the desired trajectory (which will be referred to as a *reference trajectory* below), and its value is related to the intensity of applied assistance.

To explain the adopted GVF in detail, we start by introducing *preferred direction* and *non-preferred direction* vectors. Let $\boldsymbol{x}$ be the position vector of the controlled object, $[x_1, x_2]^T$, and $\mathbf{s}$ be the closest point on a reference trajectory from $\boldsymbol{x}$ (see Fig. 2). Following [32], we first define the error $\mathbf{e}$ to be

$$\mathbf{e} = \mathbf{s} - \boldsymbol{x}. \tag{1}$$

Next, let $\mathbf{h}$ and $\boldsymbol{f}$ be a tangential vector on $\mathbf{s}$ and the control force generated by a user input, respectively. We define a vector toward the preferred direction, denoted by $\boldsymbol{\delta}$, as

$$\boldsymbol{\delta} = \text{signum}\left(\boldsymbol{f}^T\mathbf{h}\right)\frac{\mathbf{h}}{\|\mathbf{h}\|} + k_{\mathbf{e}}\mathbf{e} \tag{2}$$

where $k_{\mathbf{e}}$ is a positive constant. Correspondingly, the non-preferred direction, denoted by $\boldsymbol{\delta}^\perp$, is defined as an orthogonal direction to the preferred direction.

Now, we define a *preservation level*, denoted by $\kappa_{\delta^\perp}$, as a monotonically non-increasing function from 1 to $\underline{\kappa}_{\delta^\perp}$ with respect to $\|\mathbf{e}\|$

$$\kappa_{\delta^\perp}(\|\mathbf{e}\|) = \begin{cases} \underline{\kappa}_{\delta^\perp}, & \text{if } \|\mathbf{e}\| > \dfrac{d}{2} \\ \underline{\kappa}_{\delta^\perp} + \left[\dfrac{d/2 - \|\mathbf{e}\|}{\nu}\right]^2 & \\ \quad \times (1 - \underline{\kappa}_{\delta^\perp}), & \text{if } \dfrac{d}{2} - \nu < \|\mathbf{e}\| \le \dfrac{d}{2} \\ 1.0, & \text{if } \|\mathbf{e}\| \le \dfrac{d}{2} - \nu \end{cases} \tag{3}$$

where $\frac{d}{2}$ is half the width of a virtual fixture, and ν is the width of an assist zone in which $\kappa_{\delta^\perp}$ changes its value [32].[2] ν is typically defined as $\nu = 0.9 \times \frac{d}{2}$, meaning that if the deviation of the controlled object is less than the 10% of road half width, no assistance will be provided. From Eq. (3), the lower preservation level is, the stronger assistance will be applied to the user. Since the preservation level $\kappa_{\delta^\perp}$ will play an important role for deriving guiding force $\boldsymbol{f}_g$, it is more convenient to consider *the level of haptic assistance* $1 - \kappa_{\delta^\perp}$,

[2] $\kappa_{\delta^\perp}$ is called *attenuating admittance* in [32]; however, we use a different name (i.e., *preservation level*) to present our idea more clearly.

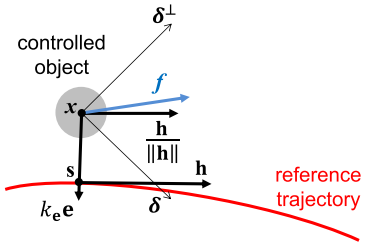


Fig. 2. Preferred direction $\boldsymbol{\delta}$ and non-preferred direction $\boldsymbol{\delta}^\perp$.

instead. Fig. 3 illustrates the two examples of haptic assistance policy, soft assistance and firm assistance policies, according to the two different GVFs with the maximum level of haptic assistance.

Similarly, we also define a *zoom level*, denoted by α, a monotonically non-decreasing function from 1 to $\bar{\alpha}$ with respect to $\|\mathbf{e}\|$

$$\alpha(\|\mathbf{e}\|) = \begin{cases} \bar{\alpha}, & \text{if } \|\mathbf{e}\| > \dfrac{d}{2} \\ \bar{\alpha} + \left[\dfrac{d/2 - \|\mathbf{e}\|}{\nu}\right]^2 & \\ \quad \times (1 - \bar{\alpha}), & \text{if } \dfrac{d}{2} - \nu < \|\mathbf{e}\| \le \dfrac{d}{2} \\ 1.0, & \text{if } \|\mathbf{e}\| \le \dfrac{d}{2} - \nu \end{cases} \tag{4}$$

where $\bar{\alpha} \ge 1.0$ is the upper bound of zoom level. For various zoom level settings, we regard $\bar{\alpha}$ as *the level of visual assistance*. From Eq. (4), we know that zoom level α has a similar shape to $1 - \kappa_{\delta^\perp}$ along the reference trajectory except that the range of the zoom level is from 1.0 to $\bar{\alpha}$; compared to that of the preservation level is from $\kappa_{\delta^\perp}$ to 1.0.

Finally, from Eqs. (3) and (4), we can summarize the property of $\kappa_{\delta^\perp}$ and α as follows:

- If the controlled object position $\boldsymbol{x}$ lies outside the virtual fixture, then $\kappa_{\delta^\perp}$ and α are set to $\underline{\kappa}_{\delta^\perp}$ and $\bar{\alpha}$, respectively.
- If the position lies in an assist zone ν, then $\kappa_{\delta^\perp}$ and α are adjusted based on the error $\|\mathbf{e}\|$.
- If the position is near to the reference trajectory, then $\kappa_{\delta^\perp}$ and α are set to 1, meaning that the user input is fully preserved, and the user will perceive no visual zooming effect.

2.2.2. Haptic assistance by a guiding force

Let $\boldsymbol{f}$ be a control force generated by a user input via haptic joystick to control an object as illustrated in Fig. 4. To decompose $\boldsymbol{f}$ into preferred direction and non-preferred direction components, we first define a projection operator P_δ that performs a vector projection of $\boldsymbol{f}$ onto $\boldsymbol{\delta}$

$$P_\delta = \frac{1}{\boldsymbol{\delta}^T\boldsymbol{\delta}}\boldsymbol{\delta}\boldsymbol{\delta}^T. \tag{5}$$

Now, we can express the control force $\boldsymbol{f}$ in terms of the preferred and the non-preferred direction component by using P_δ

$$\begin{aligned}\boldsymbol{f} &= P_\delta\boldsymbol{f} + (I - P_\delta)\boldsymbol{f}\\ &= \boldsymbol{f}_\delta + \boldsymbol{f}_{\delta^\perp}.\end{aligned} \tag{6}$$

We define the assisted control force $\boldsymbol{f}_{\text{assisted}}$ as

$$\boldsymbol{f}_{\text{assisted}} = \boldsymbol{f}_\delta + \kappa_{\delta^\perp}\boldsymbol{f}_{\delta^\perp} \tag{7}$$

where the preservation level $\kappa_{\delta^\perp}$ determines how much portion of the non-preferred direction component in $\boldsymbol{f}$ is preserved. The role of $\kappa_{\delta^\perp}$ for $\boldsymbol{f}$ can be interpreted as a suppression against $\boldsymbol{f}$. To show this, we perform simple algebraic manipulation on Eq. (7)

$$\begin{aligned}\boldsymbol{f}_{\text{assisted}} &= \boldsymbol{f}_\delta + \kappa_{\delta^\perp}\boldsymbol{f}_{\delta^\perp}\\ &= \boldsymbol{f}_\delta + \boldsymbol{f}_{\delta^\perp} - (1-\kappa_{\delta^\perp})\boldsymbol{f}_{\delta^\perp}\\ &= \underbrace{\boldsymbol{f}}_{\text{original term}} + \underbrace{[-(1-\kappa_{\delta^\perp})\boldsymbol{f}_{\delta^\perp}]}_{\text{suppressive term}}\\ &=: \boldsymbol{f} + \boldsymbol{f}_{\text{g}}.\end{aligned} \tag{8}$$

Thus, the guiding force $\boldsymbol{f}_{\text{g}}$ in Eq. (8) is defined as the suppressive term $-(1-\kappa_{\delta^\perp})\boldsymbol{f}_{\delta^\perp}$ above, which serves as an assistance to attract the controlled object to the reference trajectory. Hence, the guiding force can be transferred as haptic guidance $\boldsymbol{f}_{\text{haptic}}$ to the user's hand (see Fig. 1)

$$\boldsymbol{f}_{\text{haptic}} = k_g\boldsymbol{f}_{\text{g}} \tag{9}$$

where k_g is a positive scaling constant which can be determined by the specification of the employed haptic joystick. Note that enabling a haptic feedback functionality will be referred to as HAPTIC condition for assistance type in Section 4.

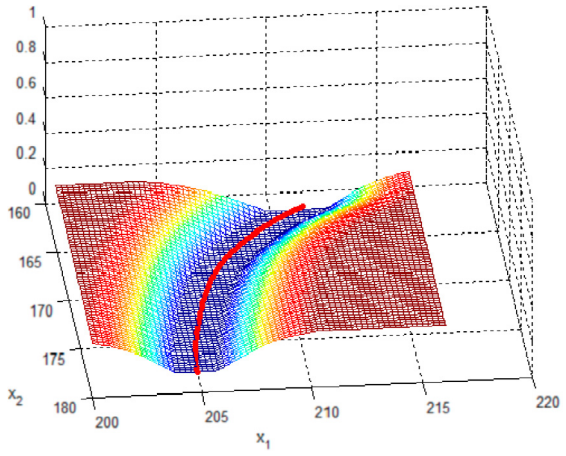
2.2.3. Visual assistance by adapting a zoom level

The developed assistive interface provides a global view and a local working field of view (WFoV) as shown in Fig. 5. By these two views, a user receives environmental information around a controlled object as a visual feedback. In [34], Accot and Zhai reported that the index of difficulty (ID) for a steering task is proportional to the integral of the inverse of the path width along the trajectory. If $\mathcal{C}$ is a curved path, then the ID can be defined

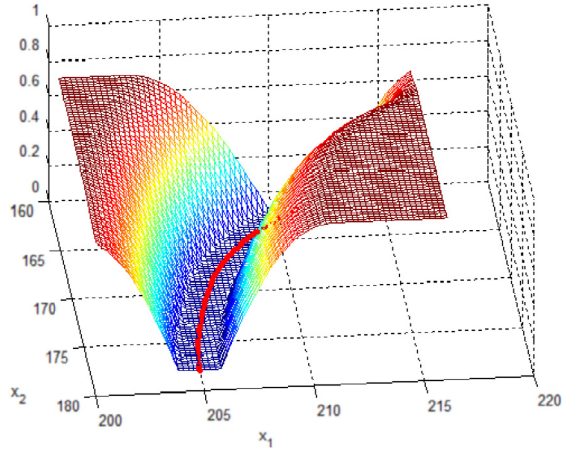
$$\text{ID} = \int_{\mathcal{C}} \frac{ds}{W(s)} \tag{10}$$

where s and $W(s)$ are a continuous curves along the center of $\mathcal{C}$ and the width of path, respectively.

Inspired by their findings, we design our assistive HRI interface to perform the zooming-in and out of the local WFoV so that it adapts to the difficulty of a given task by dynamically changing a zoom level α. To reduce the task difficulty, α increases (accordingly, the local WFoV is zoomed-in) as the controlled object further deviates from the reference trajectory by a user control. However, while the controlled object is moving at a velocity $\dot{\boldsymbol{x}}$, if α increases then its velocity being seen by a user through the local



(a) Soft assistance.



(b) Firm assistance.

Fig. 3. Two examples of haptic assistance policy according to the two different GVFs with the maximum level of haptic assistance $1-\underline{\kappa}_{\delta^\perp}$ (z-axis value in the figures). (a) soft assistance: if $1-\underline{\kappa}_{\delta^\perp} = 0.1$ and (b) firm assistance: $1-\underline{\kappa}_{\delta^\perp} = 0.9$. The solid-red line in the middle represents a reference trajectory.

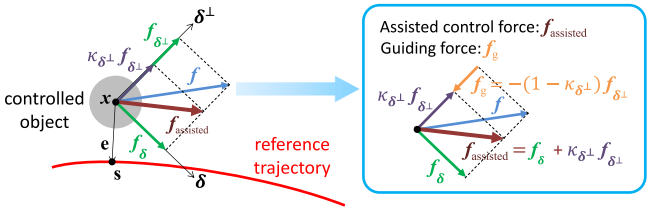


Fig. 4. Assisted control force $\boldsymbol{f}_{\text{assisted}}$ and guiding force $\boldsymbol{f}_{\text{g}}$.

WFoV window, denoted by $\dot{\boldsymbol{x}}_{\text{seen}}$, increases accordingly.[3] Namely, a relationship between $\dot{\boldsymbol{x}}_{\text{seen}}$ and $\dot{\boldsymbol{x}}$ is

$$\dot{\boldsymbol{x}}_{\text{seen}} = \alpha\dot{\boldsymbol{x}} \tag{11}$$

as illustrated in Fig. 6. Since varying v_{seen} may cause unexpected side effects, the actual velocity of the controlled object should be compensated while α is being adjusted. This compensation can be

[3] The closer the observer's distance is to an object, the faster the velocity is perceived.

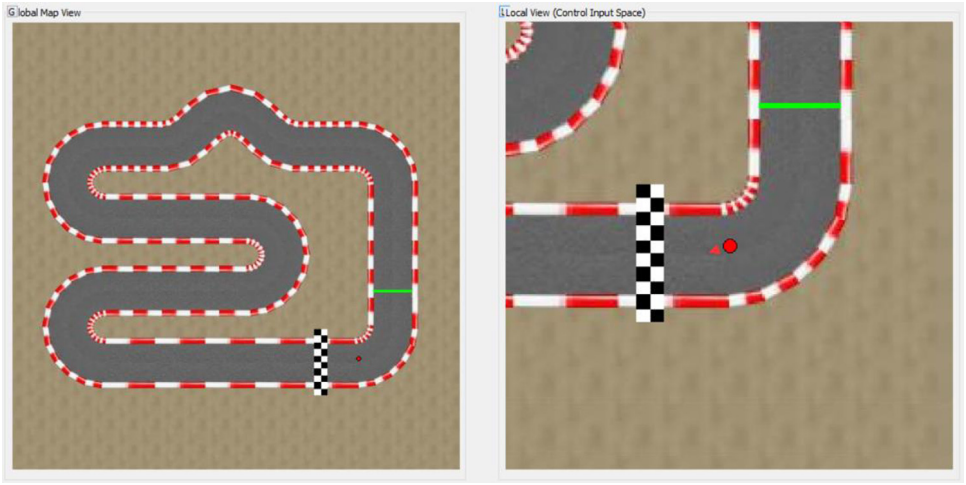
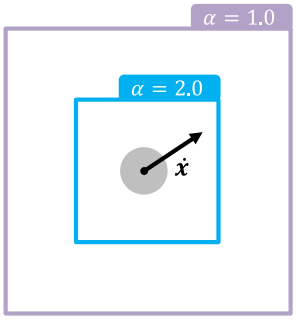
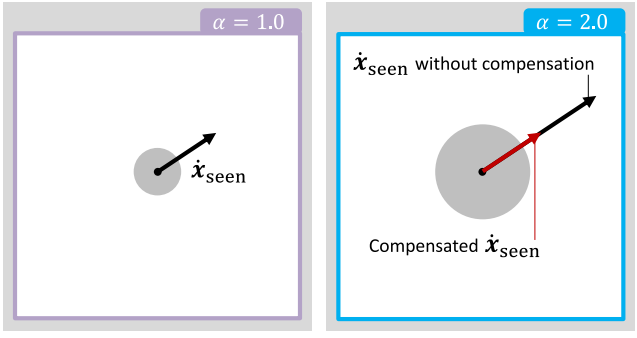


Fig. 5. Two windows in the developed assistive HRI interface providing visual assistance. A global view (left) and a local working field of view (right). The local working field of view will be zoomed-in and out based on Eq. (4).



(a) Two different WFoVs.



(b) The user will perceive $\dot{\boldsymbol{x}}_{\text{seen}}$ via the local screen.

Fig. 6. (a) Two different working field of view examples with respect to $\alpha = 1.0$ and $\alpha = 2.0$. (b) Although $\dot{\boldsymbol{x}}$ remains unchanged, the user will perceive two different $\dot{\boldsymbol{x}}_{\text{seen}}$s with respect to α. Therefore, the velocity of the controlled object needs to be compensated by multiplying $\frac{1}{\alpha}$ to $\dot{\boldsymbol{x}}$ when α is being adjusted.

easily done by multiplying $\frac{1}{\alpha}$ to $\dot{\boldsymbol{x}}$ so that $\dot{\boldsymbol{x}}_{\text{seen}}$ becomes

$$\dot{\boldsymbol{x}}_{\text{seen}} = \alpha \left(\frac{\dot{\boldsymbol{x}}}{\alpha} \right) = \dot{\boldsymbol{x}} \tag{12}$$

which represents the velocity of the controlled objects is indeed decreased proportionally as α increases (and vice versa). Consequently, the user perceives as if the controlled object is moving at the consistent velocity in the local WFoV while α is being adjusted. We note that enabling a zooming functionality of the assistive HRI interface will be referred to as VISUAL condition for assistance type in Section 4.

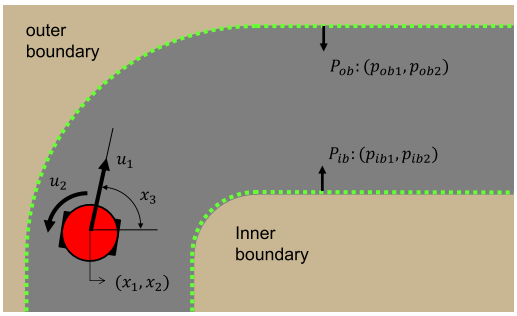


Fig. 7. The mobile robot kinematics and the inner and the outer road boundaries.

2.3. Joystick configuration, a control force, and a mobile robot control command

Throughout Section 2.2, we assumed that a controlled object could freely move toward any directions on 2D Cartesian plane (x_1-x_2 plane) when a control force $\boldsymbol{f} = [f_{x_1}, f_{x_2}]^T$ was applied. Namely, the control force $\boldsymbol{f}$ acting on the controlled objects produced $\dot{\boldsymbol{x}} = [\dot{x}_1, \dot{x}_2]^T$ whose elements represents two translational velocities. However, in practice our controlled object is a mobile robot with two differentially driving wheels of which kinematic equation is

$$\begin{aligned} \dot{x}_1 &= \cos(x_3)u_1 \\ \dot{x}_2 &= \sin(x_3)u_1 \\ \dot{x}_3 &= u_2. \end{aligned} \tag{13}$$

Therefore, it cannot take a translational maneuver toward a perpendicular direction with respect to a current heading direction; in other words, there exist nonholonomic constraints.

For this reason, we need to define a mapping from $\dot{\boldsymbol{x}}$, the velocity of the controlled object, to u, the linear and angular velocity of the mobile robot, as well as how to produce $\boldsymbol{f}$ via haptic joystick configuration $\boldsymbol{q} = [q_{x_1}, q_{x_2}]^T$. Note that we introduce the production of $\boldsymbol{f}$ via $\boldsymbol{q}$ as well as the mapping from $\dot{\boldsymbol{x}}$ to u in Appendix.

3. Approach to customize assistance policy and assistance level

3.1. The kinematic model of a mobile robot in a discrete-time

Let $[x_1, x_2, x_3]^T$ be the position and the orientation of a mobile robot, and $[u_1, u_2]^T$ be a control input as illustrated in Fig. 7. The

discrete-time kinematic equations of the mobile robot are given as follows [35,36]:

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + t_s \cos(x_{3,k}) u_{1,k} \\ x_{2,k+1} &= x_{2,k} + t_s \sin(x_{3,k}) u_{1,k} \\ x_{3,k+1} &= x_{3,k} + t_s u_{2,k} \end{aligned} \tag{14}$$

where t_s and k are the sampling time and corresponding time step, respectively. The points $p_{\text{ib1:2},k} = [p_{\text{ib1}}, p_{\text{ib2}}]^T \in P_{\text{ib}}$ and $p_{\text{ob1:2},k} = [p_{\text{ob1}}, p_{\text{ob2}}]^T \in P_{\text{ob}}$ are the closest points from current mobile robot position $x_{1:2} = [x_1, x_2]^T$ (for convenience, we use MATLAB-like expression interchangeably) to inner and outer side boundaries respectively.

3.2. Modeling user's control strategy as a cost function

To model a user's control strategy as a cost function, we consider the specific form of a cost function for quadratic regulator (QR) that represents how the user regulates speed, steering, and proximities to road boundaries. Let the state and control input of mobile robot be $x = \{x_0, \ldots, x_{N-1}\}$ and $u = \{u_0, \ldots, u_{N-1}\}$, for $k = 0, \ldots, N-1$, respectively. Then the cost function, $L(x, u)$, is defined as

$$\begin{aligned} L(x,u) = \sum_{k=0}^{N-1} \Big[& u_k^T R u_k + (x_{1:2,k} - p_{\text{ib1:2},k})^T Q_{\text{ib}} (x_{1:2,k} - p_{\text{ib1:2},k}) \\ & + (x_{1:2,k} - p_{\text{ob1:2},k})^T Q_{\text{ob}} (x_{1:2,k} - p_{\text{ob1:2},k}) \Big] \end{aligned} \tag{15}$$

where

$$R = \begin{bmatrix} c_v & 0 \\ 0 & c_\omega \end{bmatrix}, \quad Q_{\text{ib}} = \begin{bmatrix} c_{\text{ib}} & 0 \\ 0 & c_{\text{ib}} \end{bmatrix}, \text{ and } \quad Q_{\text{ob}} = \begin{bmatrix} c_{\text{ob}} & 0 \\ 0 & c_{\text{ob}} \end{bmatrix}. \tag{16}$$

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

An optimal control problem is to find an optimal control u^* (and its corresponding state x^*) which minimizes a given cost function

$$L(x,u) \longrightarrow (u^*, x^*).$$

From now on, an optimal control problem will be referred to as a *forward optimal control* problem for disambiguation. The forward optimal control problem with the cost function in Eq. (15) can be formalized

$$\begin{aligned} \min_u \Big[& \sum_{k=0}^{N-1} u_k^T R u_k + (x_{1:2,k} - p_{\text{ib1:2},k})^T Q_{\text{ib}} (x_{1:2,k} - p_{\text{ib1:2},k}) \\ & + (x_{1:2,k} - p_{\text{ob1:2},k})^T Q_{\text{ob}} (x_{1:2,k} - p_{\text{ob1:2},k}) \Big] \\ \text{subject to} \quad & x_{1,k+1} = x_{1,k} + t_s \cos(x_{3,k}) u_{1,k} \\ & x_{2,k+1} = x_{2,k} + t_s \sin(x_{3,k}) u_{1,k} \\ & x_{3,k+1} = x_{3,k} + t_s u_{2,k} \\ & x_0 = x_{\text{start}} \\ & x_N = x_{\text{goal}}. \end{aligned} \tag{17}$$

In contrast, an *inverse optimal control* (IOC) problem is to infer a cost function with unknown parameters from a given set of tuples (u^*, x^*) obtained from user demonstration, which is assumed to be a locally optimal solution; thus

$$(u^*, x^*) \longrightarrow L(x,u).$$

Therefore, the IOC problem corresponding to Eq. (17) can be described as

Given: (u^*, x^*) data demonstrated by a user, which is assumed to be a locally optimal solution of Eq. (17),

Infer: the user's cost function by inferring the unknown parameters, c_v, c_ω, c_{ib}, and c_{ob}, in matrices R, Q_{ib}, and Q_{ob}.

3.2.1. Applied numerical method to solve ioc problem

We start by rewriting the kinematic equation of a mobile robot in Eq. (14) as

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

Following by [37,38], we define the discrete time Hamiltonian H_k

$$\begin{aligned} & H_k(x_k, u_k, \lambda_k) \\ & = L(x_k, u_k) + \lambda_k^T f(x_k, u_k) \\ & = \Big[u_k^T R u_k + (x_{1:2,k} - p_{\text{ib1:2},k})^T Q_{ib} (x_{1:2,k} - p_{\text{ib1:2},k}) \\ & \quad + (x_{1:2,k} - p_{\text{ob1:2},k})^T Q_{ob} (x_{1:2,k} - p_{\text{ob1:2},k}) \Big] \\ & \quad + \lambda_k^T \begin{bmatrix} \cos(x_{3,k}) u_{1,k} \\ \sin(x_{3,k}) u_{1,k} \\ u_{2,k} \end{bmatrix} \end{aligned} \tag{18}$$

where $\lambda_k \in \mathbb{R}^{3\times 1}$ is a costate vector at time step k. By applying Pontryagin's Maximum Principle, we obtain the costate (propagation) equation [39,40]

$$\begin{aligned} \frac{\partial H_k}{\partial x_k} &= \begin{bmatrix} 2(x_{1,k} - p_{\text{ib1},k}) c_{\text{ib}} + 2(x_{1,k} - p_{\text{ob1},k}) c_{\text{ob}} \\ 2(x_{2,k} - p_{\text{ib2},k}) c_{\text{ib}} + 2(x_{2,k} - p_{\text{ob2},k}) c_{\text{ob}} \\ -\lambda_{1,k} \sin x_{3,k} u_{1,k} + \lambda_{2,k} \cos x_{3,k} u_{1,k} \end{bmatrix} \\ &= -\frac{\lambda_{k+1} - \lambda_k}{t_s}. \end{aligned} \tag{19}$$

Rearranging Eq. (19) yields

$$\begin{aligned} & \begin{bmatrix} 2t_s(x_{1,k} - p_{\text{ib1},k}) c_{\text{ib}} + 2t_s(x_{1,k} - p_{\text{ob1},k}) c_{\text{ob}} \\ 2t_s(x_{2,k} - p_{\text{ib2},k}) c_{\text{ib}} + 2t_s(x_{2,k} - p_{\text{ob2},k}) c_{\text{ob}} \\ t_s(-\lambda_{1,k} \sin x_{3,k} u_{1,k} + \lambda_{2,k} \cos x_{3,k} u_{1,k}) \end{bmatrix} \\ & + \begin{bmatrix} \lambda_{1,k+1} - \lambda_{1,k} \\ \lambda_{2,k+1} - \lambda_{2,k} \\ \lambda_{3,k+1} - \lambda_{3,k} \end{bmatrix} = \mathbf{0}^T. \end{aligned} \tag{20}$$

From a necessary condition for optimality [40], we also have

$$\frac{\partial H_k}{\partial u_k} = \begin{bmatrix} 2u_{1,k} c_v + (\lambda_{1,k} \cos x_{3,k} + \lambda_{2,k} \sin x_{3,k}) \\ 2u_{2,k} c_\omega + \lambda_{3,k} \end{bmatrix} = \mathbf{0}^T. \tag{21}$$

By defining a vector z_k as

$$z_k = [c_v \; c_\omega \; c_{\text{ib}} \; c_{\text{ib}} \; c_{\text{ob}} \; c_{\text{ob}} \; \lambda_{1,k+1} \; \lambda_{2,k+1} \; \lambda_{3,k+1} \; \lambda_{1,k} \; \lambda_{2,k} \; \lambda_{3,k}]^T \tag{22}$$

then two equations Eq. (20) and Eq. (21) can be combined and rewritten in the form, which gives us a system of equation r_k

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

The submatrices in the first row are (Note: $A_{12,k}$ takes a transposed representation by the limited column width)

$$\begin{aligned} A_{11,k} &= O^{3\times 2} \\ A_{12,k} &= \begin{bmatrix} 2t_s(x_{1,k} - p_{\text{ib1},k}) & 0 & 0 \\ 0 & 2t_s(x_{2,k} - p_{\text{ib2},k}) & 0 \\ 2t_s(x_{1,k} - p_{\text{ob1},k}) & 0 & 0 \\ 0 & 2t_s(x_{2,k} - p_{\text{ob2},k}) & 0 \end{bmatrix}^T \\ A_{13,k} &= I^{3\times 3} \\ A_{14,k} &= -I^{3\times 3} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -t_s u_{1,k} \sin x_{3,k} & t_s u_{1,k} \cos x_{3,k} & 0 \end{bmatrix} \end{aligned} \tag{24}$$

and in the second row

$$A_{21,k} = \begin{bmatrix} 2u_{1,k} & 0 \\ 0 & 2u_{2,k} \end{bmatrix}$$
$$A_{22,k} = O^{2\times 4}$$
$$A_{23,k} = O^{2\times 3} \tag{25}$$
$$A_{24,k} = \begin{bmatrix} \cos x_{3,k} & \sin x_{3,k} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where $O^{m\times n}$ and $I^{m\times n}$ represent m-by-n zero matrix and identity matrix, respectively.

Finally, the given discrete time IOC problem becomes identical to solving the following least squares problem

$$\min_{c,\lambda} \sum_{k=0}^{N-1} \|r_k^*\|^2 \tag{26}$$

or, equivalently

$$\min_{c,\lambda} \sum_{k=0}^{N-1} \|A_k^* z_k\|^2 \tag{27}$$

where A_k^* represents A_k evaluated at k with (x_k^*, u_k^*).

3.3. Featuring modeled control strategy and classifying into the featured group

3.3.1. Three features describing user's control strategy

After modeling each user's strategy by using inverse optimal control (IOC), a user's control strategy can be characterized by features from the inferred parameters, c_v, c_ω, c_{ib}, and c_{ob}, that describe

- c_v/c_ω: the curvature of the generated path
- $c_{\text{ib}}/c_{\text{ob}}$: its proximity to inner/outer boundaries
- $\|RQ_{\text{ib}}^{-1}\|_F + \|RQ_{\text{ob}}^{-1}\|_F$: the ratio of control effort over boundary collision avoidance

By identifying features of user's control strategy, the user can be treated as a sample point q in feature space $\mathbb{R}^3$

$$q^\# = \left(\frac{c_v}{c_\omega}, \frac{c_{\text{ib}}}{c_{\text{ob}}}, \|RQ_{\text{ib}}^{-1}\|_F + \|RQ_{\text{ob}}^{-1}\|_F \right) =: \left(q_1^\#, q_2^\#, q_3^\#\right) \tag{28}$$

where # indicates a user index. In the third feature, $\|\cdot\|_F$ (in the last feature) represents the Frobenius norm, hence

$$\|RQ_{\text{ib}}^{-1}\|_F + \|RQ_{\text{ob}}^{-1}\|_F = \sqrt{\left(\frac{c_v}{c_{\text{ib}}}\right)^2 + \left(\frac{c_\omega}{c_{\text{ib}}}\right)^2} + \sqrt{\left(\frac{c_v}{c_{\text{ob}}}\right)^2 + \left(\frac{c_\omega}{c_{\text{ob}}}\right)^2}. \tag{29}$$

3.4. Assistance customization by setting a customized virtual fixture parameters

3.4.1. Classifying users into expert groups

Recall that the purpose of assistance customization is to guide novice users with experts' control strategy. Suppose that four experts were selected, and we want to assign each expert to a group of users whose three control strategic features are most similar to the expert. To do so, we use the preselected experts as four-centroid for k-nearest neighbor (KNN) classification algorithm. Let q^{A}, q^{B}, q^{C}, and q^{D} be the four feature vectors of expert A, B, C, and D, respectively. For a user whose feature is $q^\#$, we can assign an expert q^* to the user and store them as a tuple

$$\{q^\#, q^*\} \quad \text{where} \quad q^* \leftarrow \arg\min_{*\in\{\text{A,B,C,D}\}} \|q^\# - q^*\|_1 \tag{30}$$

where $\|\cdot\|_1$ represents L_1 norm.

3.4.2. Setting a customized virtual fixture for each user

Based on the feature-based classification result for each user, a customized virtual fixture can be set by following steps:

1. **Assign a reference trajectory** γ^*: If a new task is given to a user, a reference trajectory for virtual fixturing can be generated by solving a forward optimal control problem since the assigned expert's control strategy was priorly modeled. If a known task (e.g., the task that was used to select experts) is given to the user, we can either directly employ a trajectory demonstrated by the assigned expert as the reference trajectory or generate one by solving a forward optimal control problem.
2. **Determine a width** d: The second feature representing the ratio of inner/outer boundary preference is a good cue to determine the width d of the virtual fixture. If expert$*$ is assigned to a user#, then we define d as
$$d = W\left[1 + \beta\left(q_2^\# - q_2^*\right)^2\right]^{-1} \tag{31}$$
where W is a road width, and β is a positive constant determined by the developer's choice. For instance, we set $W = 50$ (pixels) and $\beta = 1.0$.
3. **Set** $\underline{\kappa}_{\delta^\perp}$ **and** $\bar{\alpha}$: From Eq. (3), we know that $1 - \kappa_{\delta^\perp}$ monotonically increases with respect to the deviation from the reference trajectory up to $1 - \underline{\kappa}_{\delta^\perp}$. For our application, we want $\underline{\kappa}_{\delta^\perp}$ to be a function of user's task completion time $T^\#$ so that it linearly decreases within an interval $[\underline{\kappa}_{\delta^\perp}^{\min}, \underline{\kappa}_{\delta^\perp}^{\max}]$ with respect to $T^\#$
$$\underline{\kappa}_{\delta^\perp}(T^\#) = \frac{(\underline{\kappa}_{\delta^\perp}^{\min} - \underline{\kappa}_{\delta^\perp}^{\max})}{\eta} |T^\# - T^*| + \underline{\kappa}_{\delta^\perp}^{\max} \tag{32}$$
where $\underline{\kappa}_{\delta^\perp}^{\min}$ and $\underline{\kappa}_{\delta^\perp}^{\max}$ are the minimum and the maximum bound for $\underline{\kappa}_{\delta^\perp}$, respectively. T^* is the assigned expert's task completion time, and η is a constant can be determined by examining a difference between the fastest expert's and the slowest subject's task-completion time. If $|T^\# - T^*| \geq \eta$, meaning that a new user takes a longer time to complete a task than the slowest user, then $\underline{\kappa}_{\delta^\perp} = \underline{\kappa}_{\delta^\perp}^{\min}$. We used $\underline{\kappa}_{\delta^\perp}^{\min} = 0.1$ and $\underline{\kappa}_{\delta^\perp}^{\max} = 0.9$. Similarly, $\bar{\alpha}$ can be determined as an increasing function with respect to $|T^\# - T^*|$
$$\bar{\alpha}(T^\#) = \frac{(\bar{\alpha}^{\max} - \bar{\alpha}^{\min})}{\eta} |T^\# - T^*| + \bar{\alpha}^{\min}. \tag{33}$$
If $|T^\# - T^*| \geq \eta$ then $\bar{\alpha} = \bar{\alpha}^{\max}$, and $\bar{\alpha}^{\max} = 2.0$ and $\bar{\alpha}^{\min} = 1.1$ were used.

The entire assistance customization procedure is summarized in Algorithm 1.

4. Human subject experiment

4.1. Subjects and experimental setup

Twenty-three healthy young adults (11 male and 12 female, age = 18–25 yrs) participated in this study. None of the subjects had previous experiences with haptic driving tasks. Subjects were instructed to sit comfortably on an office chair with wheels at 0.8 m from a display monitor and were provided with a haptic input device. All subjects gave informed consent before their participation and were given the instructions about the whole experimental procedure. This study was approved by the Texas A&M University Institutional Review Board.

The experimental setup (see Fig. 8) consists of a Novint Falcon as an input device, a computer that runs the simulation, and a 21-inch monitor showing the simulated environment. The top right

Algorithm 1: ASSISTANCE CUSTOMIZATION.

Input: $\{x^*, u^*\}^{\#}$ and $T^{\#}$ of N-users. q^{A}, q^{B}, q^{C}, q^{D} of experts.
Output: User# : $\{\gamma, d, \underline{\kappa}_{\delta^\perp}, \bar{\alpha}\}$ for N-users.

1. Inferring a cost function from the user-demonstrated data
for # $\leftarrow 1$ **to** N **do**
  $[c_v, c_\omega, c_{\mathrm{ib}}, c_{\mathrm{ob}}] \leftarrow SolveInverseOptimalControl(\{x^*, u^*\}^{\#})$
end

2. Featuring user's control strategy
for # $\leftarrow 1$ **to** N **do**
  $q_1^{\#} = c_v/c_\omega$
  $q_2^{\#} = c_{\mathrm{ib}}/c_{\mathrm{ob}}$
  $q_3^{\#} = \|RQ_{\mathrm{ib}}^{-1}\|_F + \|RQ_{\mathrm{ob}}^{-1}\|_F$
end

3. Assign an expert
for # $\leftarrow 1$ **to** N **do**
  $q^* = \arg\min_{*\in\{\mathrm{A,B,C,D}\}} \|q^{\#} - q^*\|_1$
end

4. Set the customized virtual fixture parameters: $\gamma, d, \underline{\kappa}_{\delta^\perp}, \bar{\alpha}$
for # $\leftarrow 1$ **to** N **do**
  $\gamma \leftarrow \gamma^*$(expert$*$'s trajectory for user#)
  $d = W\left[1 + \beta\left(q_2^{\#} - q_2^*\right)^2\right]^{-1}$
  $\underline{\kappa}_{\delta^\perp}(T^{\#}) = \frac{(\underline{\kappa}_{\delta^\perp}^{\min} - \underline{\kappa}_{\delta^\perp}^{\max})}{\eta} |T^{\#} - T^*| + \underline{\kappa}_{\delta^\perp}^{\max}$
  $\bar{\alpha}(T^{\#}) = \frac{(\bar{\alpha}^{\max} - \bar{\alpha}^{\min})}{\eta} |T^{\#} - T^*| + \bar{\alpha}^{\min}$
end

return User#:$\{\gamma, d, \underline{\kappa}_{\delta^\perp}, \bar{\alpha}\}$

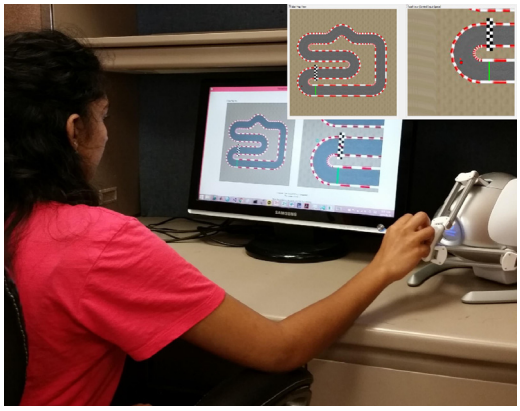


Fig. 8. The apparatus of the experimental setup. Right top corner shows simulator interface while the subject is performing a given task (the photo is used by courtesy of the participant).

corner in Fig. 8 shows our simulator interface displayed to the subject. The interface window size was approximately 19.63 cm $\times$ 35.55 cm (height $\times$ width) in 1920 $\times$ 1080 display resolution and provided a global view and a local working field of view of the environment. The other area of the monitor was filled with uniform gray color to prevent distractions caused by background content.

4.2. Procedure

4.2.1. Practice session

During the practice session, a user (subject) was asked to familiarize him/herself with the developed assistive interface by driving a mobile robot. The user was allowed to drive the mobile robot around either inside or outside of the road boundaries. However, the user was instructed that it would be regarded as a task failure to drive the mobile robot outside of the road boundary after the practice session. We provided the user a five-minute practice session to make the user to be familiarized with the experimental settings and the virtual navigation system. Also, by limiting the practice session to five minutes, we tried to avoid the user learning about the task. Note that the road path presented in the practice session was not used in the rest of the sessions.

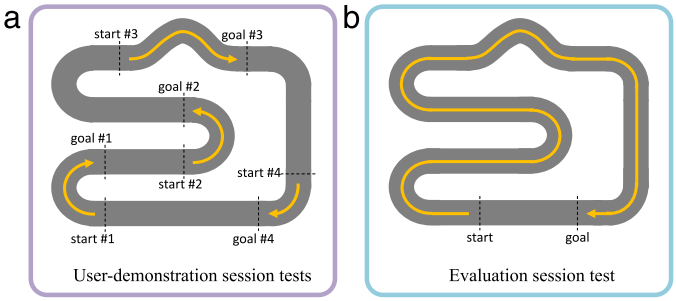


Fig. 9. Track and given tests for (a) the user demonstration session and (b) the evaluation session. The arrows and dotted-lines indicate start and goal positions for each test.

4.2.2. User demonstration session

During the user demonstration session, users were instructed to drive the mobile robot from the starting position to the goal position as fast and safely as possible for each of four locations (see Fig. 9(a) for the starting and goal positions for four locations). The simulator provided four tasks (or locations) each of which has different road curvature and turning angles as illustrated in Fig. 9. These four locations were chosen due to the following reasons: (1) these four locations are more difficult regions to complete the task compared with straight path, (2) variability of task performance (i.e., task completion time) is more pronounced in more difficult tasks, and (3) accordingly, the users' control strategy through these four locations are directly related to their task performance in the evaluation session where the entire lap along a given track includes the four curved locations and straight paths (see Fig. 9(b)).

The width of road displayed on the screen was fixed at all times. Each test set was presented to the user five times; therefore there were 4 (locations) $\times$ 5 (repetition per one location) $=$ 20 trials. During the 20 trials, the test sets were presented in randomized order. Among the five repetitions for each test set, a trial of which completion time is median was selected as the user-demonstrated data (x^*, u^*) to exclude any coincidental best/worst demonstration.

4.2.3. Assistance customization session

After obtaining user-demonstration data, the procedure was followed by solving an inverse optimal control problem to infer a cost function with the user-demonstration data. The cost function featured users' control strategy, and classified the users into their corresponding expert groups based on the three features. Then, $\{\gamma, d, \underline{\kappa}_{\delta^\perp}, \bar{\alpha}\}$ for the users were customized as presented in Algorithm 1. During this session, users were asked to take a 10-minute break while this procedure was being processed.

4.2.4. Evaluation session

Finally, after 10 min break of the assistance customization session, users were instructed to complete an entire lap of a given track during the evaluation session. When the mobile robot collided with the road boundaries, the task was restarted from the

Table 1
Two metrics of user's performance improvement. The acronyms stand for NO-ASSIST(N), HAPTIC(H), VISUAL(V), HAPTIC-VISUAL(HV), CUSTOMIZED(CS), and NON-CUSTOMIZED(NCS). Value represents mean(standard deviation) for four assistance type conditions and two parameterization type conditions, and p-values for interaction effect (assistance type × parameterization type). Superscripts represent significant differences from the other main effects conditions resulting from Bonferroni adjusted comparison.

Performance metrics	Assistance type				Parameterization type		Interaction p-value
	N	H	V	HV	CS	NCS	
Time improvement (%)	0.00 (0.00)^{H}	9.03 (2.89)^{N,V,HV}	−10.05 (3.48)^{H}	−6.91 (3.18)^{H}	−0.64 (2.51)^{NCS}	−3.33 (2.20)^{CS}	**<0.001**
Mean. $\boldsymbol{f}_g$ improvement (%)	0.00 (0.00)^{H,HV}	46.75 (3.20)^{N,V}	13.39 (6.76)^{H,HV}	44.56 (4.00)^{N,V}	42.142 (4.31)^{NCS}	10.21 (3.44)^{CS}	**<0.001**

center of the road at the point of failure, and users received a 3-second penalty. The maximum number of collisions was limited to ten. If the user collided the mobile robot more than ten times, the user's data was discarded. There were eight evaluation sets as a combination of two dependent factors: different *assistance type* {NO-ASSIST(NA), HAPTIC(H), VISUAL(V), HAPTIC-VISUAL(HV)} and *parameterization type* {CUSTOMIZED(CS), NON-CUSTOMIZED(NCS)}. When the condition of the parameterization type was NON-CUSTOMIZED, the virtual fixturing was set with a road center as a reference trajectory γ with $\underline{\kappa}_{\delta^\perp} = 0.1$ and $\bar{\alpha} = 2.0$. Each evaluation trial was presented three times. Hence, there were 8 (evaluation sets) × 3 (repetition) = 24 trials. The evaluation sets were presented in a randomized order during the 24 trials to prevent learning effects.

4.3. Data storing and analysis

During the evaluation session, the developed application recorded the position (x_1, x_2) and the orientation x_3 of the mobile robot and guiding force $\boldsymbol{f}_g$ every 20 ms. It also stored task completion time, task number, trial number, position, orientation, linear/angular velocities, the number of collisions, and task result (i.e., success or failure). The improvement in completion time and average guiding force were used as two metrics for task performance. One subject data (subject#17) was excluded due to unexpected distraction during the experiment; hence, 22-subjects data were used for data analysis.

To identify the effect of two dependent factors, assistance type and parameterization type, a repeated measures analysis of variance (rANOVA) was performed with the significance level of $p < 0.05$ (SPSS, v21, Chicago, IL). When the assumption of sphericity was violated for the main effects, the degree of freedom was corrected using Greenhouse-Geisser estimates of sphericity. For a pairwise comparison, the Bonferroni-adjusted pairwise comparison was used, and the result is presented in the form of "mean difference (standard error)". The significance level was set at $p < 0.05$. For better readability, we present the "mean (standard error)" values of the two user performance metrics across four assistance type conditions and two parameterization type conditions in Table 1.

5. Results

We present the effects of two independent factors, assistance type (N, H, V, HV) and parameterization type (CS, NCS), on two user performance metrics, i.e., completion time improvement and mean guiding force improvement. Please note that when the assistance type was set to N or V, our assistive HRI interface did not transfer the actual guiding force to users' hand via haptic joystick because N and V do not involve haptic feedback. However, here, we include all four assistance type conditions, N, H, V, HV for the analyses of the improvement in completion time as well as in mean guiding force over time. This allows us to identify the possibly desired efforts to guide a mobile robot to a reference trajectory by examining the calculated mean guiding force under the four assistance type conditions. The positive improvements were defined as "less completion time" and "less mean guiding force", respectively.

5.1. Effects of assistance type

Effect on completion time improvement. The rANOVA result on the improvement in completion time indicated that there was a significant main effect of the assistance type, $F(1.70, 35.65) = 22.17$, $p < 0.05$. The pairwise comparison revealed that N vs. H, −9.03 (2.89), $p = 0.031$, N vs. V, 10.05 (3.48), $p = 0.053$, and N vs. HV, 6.91 (3.18), $p = 0.249$, indicating a significance difference between N and H but only a tendency between N and V. No significant difference was found between N and HV. In contrast, the comparisons of H vs. other conditions showed significant differences as H vs. N, 9.03 (2.89), $p = 0.031$, H vs. V, 19.08 (1.83), $p < 0.001$, and H vs. HV, 15.94 (1.59), $p < 0.001$, implying that H led to the greatest improvement over other conditions in completion time. No other significant differences were found between the rest of assistance type conditions.

Effect on mean guiding force improvement. Analysis on mean guiding force improvement yielded a significant main effect for assistance type, $F(1.56, 32.77) = 51.03$, $p < 0.05$. Pairwise comparisons revealed N vs. H, −46.75 (3.20), $p < 0.001$, N vs. V, −13.39 (6.76), $p = 0.364$, and N vs. HV, −44.56 (4.00), $p < 0.001$, suggesting that the improvement in mean guiding force increased significantly under assistance type conditions H and HV compared to N. Since the mean guiding force mostly improved (i.e., yielded the least mean difference) under H with significance, we performed pairwise comparison with H vs. other assistance type conditions, and it revealed H vs. N, 46.75 (3.20), $p < 0.001$, H vs. V, 33.36 (5.44), $p < 0.001$, and H vs. HV, 2.19 (1.76), $p = 1.000$. Hence, H significantly lowered the average guiding force than N and V but no significant difference could be found for HV.

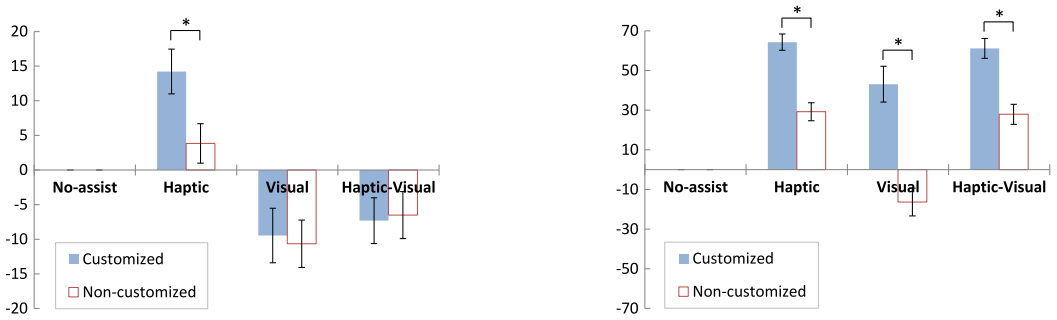
5.2. Effects of parameterization type

Effect on completion time improvement. Parameterization type showed the significant main effect on the improvement in completion time, $F(1, 21) = 5.28$, $p < 0.05$. The pairwise comparison revealed a significant difference CS vs. NCS, 2.69 (1.17), $p = 0.032$, which implies the users' improvement in completion time is higher when parameters are customized.

Effect on mean guiding force improvement. The significant main effect of the parameterization type on the average guiding force was also found $F(1, 21) = 49.92$, $p < 0.001$. Pairwise comparison yielded a significant difference CS vs. NCS, 31.93 (4.52), $p < 0.001$, which indicates the mean guiding force improvement is higher when parameters are customized.

5.3. Interaction effects of assistance type × parameterization type

Interaction effect on completion time improvement. The analysis revealed that there was a significant interaction effect between the assistance type and the parameterization type in completion time improvement, $F(3, 63) = 9.66$, $p < 0.001$, as shown in the rightmost column in Table 1. Fig. 10(a) depicts that users' completion time improvement tended to increase in H and V when assistance was customized whereas it tended to decrease in HV. This indicates that parameterization type had different effects on



(a) Completion Time Improvement (%).

(b) Mean Guiding Force Improvement (%).

Fig. 10. Mean values for (a) users' completion time improvement and (b) mean guiding force over time under four assistance type conditions: No-assist, Haptic, Visual, Haptic-Visual. Each assistance type condition showed two mean values when its parameters were Customized and Non-customized. Error bars indicate standard error. Significance is indicated for $p < 0.05$ ($*$) from the pairwise comparison between CUSTOMIZED and NON-CUSTOMIZED within each of the assistance type conditions.

completion time improvement depending on which condition of assistance type was provided. Pairwise comparisons revealed that CS significantly increased completion time improvement under H ($p < 0.001$).

Interaction effect on mean guiding force improvement. A significant interaction effect between the assistance type and the parameterization type was also found in mean guiding force improvement, $F(2.15, 45.07) = 25.69$, $p < 0.001$. From Fig. 10(b), we can see that mean guiding force improvement simply tended to decrease in all assistance type conditions. Hence, the significant of the interaction effect on mean guiding force improvement might be caused by using N as a baseline condition. The pairwise comparison indicated that CS significantly increased the mean force improvement under all assistance type conditions, H, V, and HV, with $p < 0.001$.

6. Discussion

The main idea of this study was to develop an assistive HRI interface providing customized haptic and visual feedback. We also wanted to identify the effect of assistance type and parameterization type on users' performance. The uniquenesses of our design approach are (1) to set the maximum level of assistance based on user's performance as well as control strategy and (2) to modify assistance interactively with respect to a user input and the current state of a mobile robot. Furthermore, we wanted to find some insights about an optimal combination of assistance type and parameterization type throughout data analysis.

H outperformed the other assistance type conditions regarding both completion time improvement and mean guiding force improvement for our mobile robot control task (see Table 1). Although the benefit of haptic feedback still has been debatable [4,12,41], this could be interpreted as the positive effect of applying haptic feedback for our task. In this perspective, it was worthwhile to further investigate whether V can be improved when H was added. Pairwise comparison HV vs V revealed 3.14(1.51) with $p = 0.299$ in completion time improvement and 31.16(4.66) with $p < 0.001$ in mean guiding force improvement. Therefore, adding H on V would be beneficial to lessen the total desired effort to guide the mobile robot.

Next, the users' performance tended to degrade in completion time improvement when V was solely provided. This outcome is consistent to existing findings in [42] and [43] which presented the deteriorative tendency of user performance with zooming and visual feedback, respectively. When visual feedback is being zoomed in/out during the task, the performance degradation might be caused due to changes in a working field of view. Specifically, user's behavior and mobile robot dynamics became different in the enlarged working field. Consequently, providing guidance based on the expert's control strategy might not work properly since it was obtained under the absence of visual cue. In contrast, V encouraged mean guiding force improvement.

CS seemed to promote the improvement in both completion time and mean guiding force except the improvement in completion time under HV. We have already observed that V has the negative effect on user performance. Therefore, the exception under HV was possibly caused by combining the customized V with H. Additional studies are needed, however, to substantiate the varying negative effect of V.

Statistical analysis revealed that the customized haptic (H/CS) is the optimal combination to provide feedback for given tasks in this study. However, if we only consider the mean guiding force as a performance metric, then HV/CS is also a competitive candidate as an option to provide feedback (the estimated "mean(standard error)" values for H/CS and HV/CS are 64.29(4.11) and 61.18(5.01), respectively). Therefore, we can choose the option to provide feedback for a newly given task by various definition about performance index.

Lastly, there exists a wide range of real world applications for which the proposed assistance customization algorithm can be applied. For example, medical instruments such as the endoscope [8], laparoscope [44], and surgical robots [45] are promising candidates expected to encourage the user's performance improvement by applying the haptic and visual assistance customization.

7. Conclusion

In this study, we proposed an approach to design an assistive human–robot interface that provides haptic and visual feedback as well as an algorithm to customize two feedbacks based on user's control strategy. In our approach, the user's control strategy was modeled by using inverse optimal control. Then, it was featured to describe the geometric and behavioral characteristic of user control. For each user, features were utilized to assign an expert to the user so that both features were the closest to each other. Finally, assistance was customized based on the assigned expert's strategy to provide feedback via a virtual fixturing.

The result showed that the user performances were improved when the customized assistance was provided; however, contradicting tendency in completion time improvement was also observed when two assistance types were combined. Among four assistance type conditions, haptic outperformed the other assistance type conditions regarding the improvement in both completion time and mean guiding force. Consequently, the customized haptic feedback was revealed as an optimal assistance way for the given mobile robot control task.

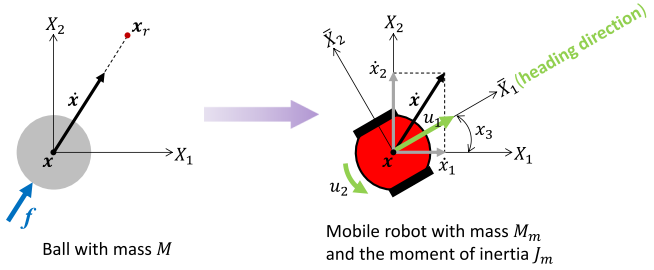


Fig. 11. Suppose that $\boldsymbol{x}_r$ is the desired position of a ball determined by the user input, then the control force $\boldsymbol{f} = M\ddot{\boldsymbol{x}}_r + B_d(\dot{\boldsymbol{x}}_r - \dot{\boldsymbol{x}}) + K_p(\boldsymbol{x}_r - \boldsymbol{x})$ is applied to the ball (left). Afterward, the generated velocity $\dot{\boldsymbol{x}}$ of the ball is aligned to the current heading direction of the mobile robot and scaled by c_1 and c_1 to yield the control input $[u_1, u_2]$ for the mobile robot.

Future studies should be followed in the directions of applying the proposed approach for various tasks, finding more features that can describe the geometric and behavioral characteristics of user control, and employing various sensory feedback types in assistive HRI interface design.

Appendix

Suppose that a controlled object is a ball of mass M as shown in Fig. 11(left). Let $\boldsymbol{x} = [x_1, x_2]^T$ and $\boldsymbol{f}$ be the current position of the ball (center) and a control force applied to the ball, respectively. The equation of motion of this system is

$$M\ddot{\boldsymbol{x}} = \boldsymbol{f}. \tag{34}$$

Also, let $\boldsymbol{x}_r = [x_{r1}, x_{r2}]^T$ be the desired position of the ball. In our assistive interface, this desired position is translated by a user input via the haptic joystick

$$\dot{\boldsymbol{x}}_r = k\boldsymbol{q} \tag{35}$$

where k is a positive constant which can be determined by a haptic joystick specification (e.g., the range of motion and the maximum retrieved value from the encoder).

Next, if the desired position $\boldsymbol{x}_r$ is set by the user input, the ball can be guided to $\boldsymbol{x}_r$ by applying the following control force $\boldsymbol{f}$

$$\boldsymbol{f} = M\ddot{\boldsymbol{x}}_r + B_d(\dot{\boldsymbol{x}}_r - \dot{\boldsymbol{x}}) + K_d(\boldsymbol{x}_r - \boldsymbol{x}) \tag{36}$$

where B_d and K_d are desired damping and stiffness parameters, respectively. Substituting Eq. (36) into Eq. (34) yields

$$M\ddot{\tilde{\boldsymbol{x}}} + B_d\dot{\tilde{\boldsymbol{x}}} + K_d\tilde{\boldsymbol{x}} = \boldsymbol{0}, \quad \text{where } \tilde{\boldsymbol{x}} = \boldsymbol{x}_r - \boldsymbol{x}, \tag{37}$$

which implies the ball reaches to the desired position. When an assisted control force is applied, Eq. (34) becomes

$$M\ddot{\boldsymbol{x}} = \boldsymbol{f}_{\text{assisted}} = \boldsymbol{f} + \boldsymbol{f}_g, \tag{38}$$

and substituting Eq. (36) into Eq. (38) yields

$$M\ddot{\tilde{\boldsymbol{x}}} + B_d\dot{\tilde{\boldsymbol{x}}} + K_d\tilde{\boldsymbol{x}} = \boldsymbol{f}_g \quad \text{where } \tilde{\boldsymbol{x}} = \boldsymbol{x}_r - \boldsymbol{x}. \tag{39}$$

As the ball is being guided to a reference trajectory, $\boldsymbol{f}_g$ will be eliminated ($\boldsymbol{f}_g \to \boldsymbol{0}$). Hence, Eq. (37) will eventually hold.

In Eq. (37) or Eq. (39), the velocity of the ball generated by the control force is $\dot{\boldsymbol{x}} = [\dot{x}_1, \dot{x}_2]^T$. Considering our controlled object in practice is a mobile robot with nonholonomic constraints, we need to define a mapping from $[\dot{x}_1, \dot{x}_2]^T$ to $[u_1, u_2]^T$. As illustrated in Fig. 11(right), this mapping can be defined by aligning x_1-axis to the current heading direction x_3 of the mobile robot, then scaling the aligned $\dot{x}_1$ and $\dot{x}_2$ to yield u_1 and u_2

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} c_1 & 0 \\ 0 & c_2 \end{bmatrix} \begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} c_1 & 0 \\ 0 & c_2 \end{bmatrix}}_{\text{scaling}} \underbrace{\begin{bmatrix} \cos(x_3) & \sin(x_3) \\ -\sin(x_3) & \cos(x_3) \end{bmatrix}}_{\text{aligning } x_1\text{-axis to } x_3} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} \tag{40}$$

where c_1 and c_2 are positive scaling constants. We can choose c_1 and c_2 to satisfy passivity condition: the kinetic energy of the mobile robot is not greater than that of the ball [46]. Let M_m and J_m be the mass and the moment of inertia of the mobile robot, then the passivity condition for this mapping can be expressed as

$$\frac{1}{2}M_m u_1^2 + \frac{1}{2}J_m u_2^2 \le \frac{1}{2}M(\dot{\tilde{x}}_1^2 + \dot{\tilde{x}}_2^2). \tag{41}$$

Hence, from Eq. (40) and Eq. (41), we have

$$c_1 \le \sqrt{\frac{M}{M_m}} \text{ and } c_2 \le \sqrt{\frac{M}{J_m}}. \tag{42}$$

References

[1] S.A. Bowyer, B.L. Davies, F.R.y. Baena, Active constraints/virtual fixtures: A survey, IEEE Trans. Robot. 30 (1) (2014) 138–157.
[2] A. Dragan, S. Srinivasa, Formalizing assistive teleoperation, Robot. Sci. Syst. (2012).
[3] L.M. Crespo, D.J. Reinkensmeyer, Haptic guidance can enhance motor learning of a steering task, J. Motor Behav. 40 (6) (2008) 545–557.
[4] L. Marchal-Crespo, S. McHughen, S.C. Cramer, D.J. Reinkensmeyer, The effect of haptic guidance, aging, and initial skill level on motor learning of a steering task, Exp. Brain Res. 201 (2) (2010) 209–220.
[5] J.J. Abbott, P. Marayong, A.M. Okamura, Haptic virtual fixtures for robot-assisted manipulation, in: Robotics Research, Springer, 2007, pp. 49–64.
[6] I. Emeagwali, P. Marayong, J.J. Abbott, A.M. Okamura, Performance analysis of steady-hand teleoperation versus cooperative manipulation, in: Proceedings of 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004, pp. 316–322.
[7] R. Reilink, S. Stramigioli, S. Misra, Image-based flexible endoscope steering, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 2339–2344.
[8] R. Reilink, S. Stramigioli, A.M. Kappers, S. Misra, Evaluation of flexible endoscope steering using haptic guidance, Int. J. Med. Robot. Comput. Assist. Surg. 7 (2) (2011) 178–186.
[9] I. Nisky, A. Pressman, C.M. Pugh, F.A. Mussa-Ivaldi, A. Karniel, Perception and action in simulated telesurgery, in: Haptics: Generating and Perceiving Tangible Sensations, Springer, 2010, pp. 213–218.
[10] N. Padoy, G. Hager, Human-machine collaborative surgery using learned models, in: Proceedings of IEEE International Conference on Robotics and Automation, 2011, pp. 5285–5292.
[11] P. Griffiths, R. Gillespie, Shared control between human and machine: haptic display of automation during manual control of vehicle heading, in: Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings. 12th International Symposium on, 2004, pp. 358–366.
[12] D. Powell, M.K. O'Malley, The task-dependent efficacy of shared-control haptic guidance paradigms, IEEE Trans. Haptics 5 (3) (2012) 208–219.
[13] D.A. Abbink, M. Mulder, E.R. Boer, Haptic shared control: smoothly shifting control authority?, Cogn. Techno. Work 14 (1) (2012) 19–28.
[14] J.G. Storms, D.M. Tilbury, Blending of human and obstacle avoidance control for a high speed mobile robot, in: American Control Conference, ACC, 2014, IEEE, 2014, pp. 3488–3493.
[15] T. Carlson, Y. Demiris, Collaborative control in human wheelchair interaction reduces the need for dexterity in precise manoeuvres, in: Proceedings of Workshop at ACM/IEEE HRI, Robotic Helpers: User Interaction, Interfaces and Companions in Assistive and Therapy Robotics, 2008, pp. 59–66.
[16] R.E. Cowan, B.J. Fregly, M.L. Boninger, L. Chan, M.M. Rodgers, D.J. Reinkensmeyer, et al., Recent trends in assistive technology for mobility, J. Neuroeng. Rehabil. 9 (1) (2012) 20.
[17] L. Marchal-Crespo, J. Furumasu, D.J. Reinkensmeyer, A robotic wheelchair trainer: design overview and a feasibility study, J. Neuroeng. Rehabil. 7 (1) (2010) 40.
[18] C. Urdiales, M. Fernández-Carmona, J.M. Peula, U. Cortés, R. Annichiaricco, C. Caltagirone, F. Sandoval, Wheelchair collaborative control for disabled users navigating indoors, Artif. Intell. Med. 52 (3) (2011) 177–191.

[19] R. Simpson, D. Poirot, F. Baxter, The hephaestus smart wheelchair system, IEEE Trans. Neural Syst. Rehabil. Eng. 10 (2) (2002) 118–122.
[20] R.C. Simpson, Smart wheelchairs: A literature review, J. Rehabil. Res. Dev. 42 (4) (2005) 423.
[21] M.J. Jensen, A.M. Tolbert, J.R. Wagner, F.S. Switzer, J.W. Finn, A customizable automotive steering system with a haptic feedback control strategy for obstacle avoidance notification, IEEE Trans. Veh. Technol. 60 (9) (2011) 4208–4216.
[22] J.L. Emken, R. Benitez, D.J. Reinkensmeyer, Human-robot cooperative movement training: learning a novel sensory motor transformation during walking with robotic assistance-as-needed, J. NeuroEng. Rehabil. 4 (1) (2007) 1.
[23] R.B. Gillespie, M. OModhrain, P. Tang, D. Zaretzky, C. Pham, The virtual teacher, in: Proceedings of the ASME Dynamic Systems and Control Division, vol. 64, American Society of Mechanical Engineers, 1998, pp. 171–178.
[24] H. Lee, S. Choi, Combining haptic guidance and haptic disturbance: an initial study of hybrid haptic assistance for virtual steering task, in: Haptics Symposium, HAPTICS, 2014 IEEE, IEEE, 2014, pp. 159–165.
[25] C. Passenberg, R. Groten, A. Peer, M. Buss, Towards real-time haptic assistance adaptation optimizing task performance and human effort, in: World Haptics Conference, WHC, 2011 IEEE, 2011, pp. 155–160.
[26] C. Passenberg, A. Glaser, A. Peer, Exploring the design space of haptic assistants: the assistance policy module, IEEE Trans. Haptics 6 (4) (2013) 440–452.
[27] J. Smisek, W. Mugge, J.B. Smeets, M.M. van Paassen, A. Schiele, Adapting haptic guidance authority based on user grip, in: Systems, Man and Cybernetics, SMC, 2014 IEEE International Conference on, IEEE, 2014, pp. 1516–1521.
[28] J. Smisek, M.M. van Paassen, M. Mulder, D.A. Abbink, Neuromuscular analysis based tuning of haptic shared control assistance for UAV collision avoidance, in: World Haptics Conference, WHC, 2013, IEEE, 2013, pp. 389–394.
[29] M.E. Fisher, F.C. Huang, V. Klamroth-Marganska, R. Riener, J.L. Patton, Haptic error fields for robotic training, in: World Haptics Conference, WHC, 2015 IEEE, IEEE, 2015, pp. 434–439.
[30] P. Abbeel, A.Y. Ng, Apprenticeship learning via inverse reinforcement learning, in: Proceedings of the 21st International Conference on Machine Learning, ACM Press, 2004.
[31] S. Levine, V. Koltun, Continuous Inverse Optimal Control with Locally Optimal Examples, in: ICML'12: Proceedings of the 29th International Conference on Machine Learning, 2012.
[32] A. Bettini, P. Marayong, S. Lang, A.M. Okamura, G.D. Hager, Vision-assisted control for manipulation using virtual fixtures, IEEE Trans. Robot. 20 (6) (2004) 953–966.
[33] A. Kucukyilmaz, Y. Demiris, One-shot assistance estimation from expert demonstrations for a shared control wheelchair system, in: Robot and Human Interactive Communication, RO-MAN, 2015 24th IEEE International Symposium on, IEEE, 2015, pp. 438–443.
[34] J. Accot, S. Zhai, Beyond Fitts' law: models for trajectory-based HCI tasks, in: Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, ACM, 1997, pp. 295–302.
[35] C. Burns, J. Zearing, R.F. Wang, D. Stipanović, Autonomous and semiautonomous control simulator, in: Proceedings of the 2010 AAAI Spring Symposium on Embedded Reasoning, 2010, pp. 10–16.
[36] S. Mastellone, D. Stipanovic, 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.
[37] D.P. Bertsekas, Dynamic Programming and Optimal Control, Vol. I, third ed., Athena Scientific, Belmont, MA, 2005.
[38] F.L. Lewis, D. Vrabie, Reinforcement learning and adaptive dynamic programming for feedback control, IEEE Circuits Syst. Mag. 9 (3) (2009) 32–50.
[39] B. Anderson, J.B. Moore, Optimal Control: Linear Quadratic Methods, Prentice-Hall, Inc., 1990.
[40] D. Liberzon, Calculus of Variations and Optimal Control Theory: A Concise Introduction, Princeton University Press, 2012.
[41] D. Feygin, M. Keehner, R. Tendick, Haptic guidance: Experimental evaluation of a haptic training method for a perceptual motor skill, in: Proceedings of IEEE 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002, pp. 40–47.
[42] M.D. Plumlee, C. Ware, Zooming versus multiple window interfaces: Cognitive costs of visual comparisons, ACM Trans. Comput.-Hum. Interact. 13 (2006) 179–209.
[43] L. Marchal-Crespo, M. van Raai, G. Rauter, P. Wolf, R. Riener, The effect of haptic guidance and visual feedback on learning a complex tennis task, Exp. Brain Res. 231 (3) (2013) 277–291.
[44] T. Howard, J. Szewczyk, Improving precision in navigating laparoscopic surgery instruments toward a planar target using haptic and visual feedback, Front. Robot. AI 3 (2016) 37.
[45] R.D. Ellis, A. Cao, A. Pandya, A. Composto, M. Chacko, M. Klein, G. Auner, Optimizing the surgeon-robot interface: the effect of control-display gain and zoom level on movement time, in: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 48, SAGE Publications, 2004, pp. 1713–1717.
[46] A. Van der Schaft, L2-Gain and Passivity Techniques in Nonlinear Control, Springer Science & Business Media, 2012.

Han U. Yoon received his Ph.D. from University of Illinois in 2014. His Ph.D. research was done as a member of the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute for Advanced Science and Technology. Dr. Yoon also has professional work experiences in Korea Institute of Industrial Technology (Nat'l Lab) and LG Electronics as a research engineer. He is currently working as a Postdoctoral Researcher in the Human Rehabilitation (HUR) group, the Department of Mechanical Engineering at Texas A&M University. He is mainly interested in assistive human–robot interaction systems, rehabilitation robotics, sensory augmentation and wearable sensor modules with IoT, and equine assisted therapeutics for whom suffers from autism, cerebral palsy, and PTSD.

Ranxiao F. Wang received her Ph.D. from the Massachusetts Institute of Technology in 1999. In 1999 she joined the faculty at the University of Illinois, where she is currently a Professor in the Department of Psychology, and the Beckman Institute for Advanced Science and Technology. She is interested in the basic principles used by humans and other animals to solve various perceptual and spatial problems, as well as how to implement these principles in artificial systems. She has been studying the reference systems and processes underlying human navigation, space perception, object and scene recognition, and visual perception of self motion. She is also studying the nature of the representations and processes in "imagined" space, the relationship between real environments and imagined/virtual environments, and techniques to facilitate exploration in virtual environments.

Seth A. Hutchinson received his Ph.D. from Purdue University in 1988. In 1990 he joined the faculty at the University of Illinois, where he is currently a Professor in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute for Advanced Science and Technology. He currently serves on the editorial boards of the International Journal of Robotics Research and the Journal of Intelligent Service Robotics, and previously served as Editor-in-Chief of the IEEE Transactions on Robotics and the first Editor-in-Chief for the RAS Conference Editorial Board. In 1996 he was a guest editor for a special section of the Transactions devoted to the topic of visual servo control, and in 1994 he was cochair of an IEEE Workshop on Visual Servoing. He was co-chair of IEEE Robotics and Automation Society Technical Committee on Computer and Robot Vision from 1992 to 1996, and has served on the program committees for more than fifty conferences related to robotics and computer vision. He has published more than 150 papers on the topics of robotics and computer vision, and is coauthor of the books "Principles of Robot Motion: Theory, Algorithms, and Implementations", published by MIT Press, and "Robot Modeling and Control", published by Wiley. Hutchinson is a Fellow of the IEEE.

Pilwon Hur received the B.S. degree in mechanical engineering from Hanyang University, Seoul, Korea, in 2004, the M.S. degree in mechanical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2006, and the M.S. degree in mathematics and the Ph.D. degree in mechanical engineering from the University of Illinois at Urbana–Champaign, USA, in 2010. He also held a postdoctoral fellowship in the Center for Ergonomics at the University of Wisconsin-Milwaukee. He is an Assistant Professor in the Department of Mechanical Engineering at Texas A&M University, College Station, TX. He directs the Human Rehabilitation (HUR) Group and is interested in whole body movement, specifically, gait, balance, slip recovery, and hand movement. To understand how human plans and controls the movements, he focuses on research in motor control, neuromechanics, and biomechanics of these movements. Based on the findings, he develops rehabilitation programs for neurologically-impaired patients using technologies from rehabilitation robotics, virtual reality, and sensory augmentation.