ELSEVIER

Robotics and Autonomous Systems 965 (2002) 1–18

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Performance evaluation and optimization of human control strategy

Yangsheng Xu [a,c], Jingyan Song [b], Michael C. Nechyba [d,*], Yeung Yam [a]

[a] *Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong*
[b] *Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Shatin, NT, Hong Kong*
[c] *The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA*
[d] *Department of Electrical and Computer Engineering, University of Florida, 216 Larsen Hall, PO Box 116200, Gainesville, FL 32611-6200, USA*

Received 28 August 2001; received in revised form 10 December 2001; accepted 10 December 2001

Abstract

Modeling human control strategy (HCS) is becoming an increasingly popular paradigm in a number of different research areas, ranging from robotics and intelligent vehicle highway systems to expert training and virtual reality computer games. Usually, HCS models are derived empirically, rather than analytically, from real-time human input–output data. While these empirical models offer an effective means of transferring intelligent behaviors from humans to robots and other machines, there is a great need to develop adequate performance criteria for these models. It is our goal in this paper to develop several such criteria for the task of human driving. We first collect driving data from different individuals through a real-time graphic driving simulator that we have developed, and identify each individual's control strategy model through the flexible cascade neural network learning architecture. We then define performance measures for evaluating two aspects of the resultant HCS models. The first is based on event analysis, while the second is based on inherent analysis. Using the proposed performance criteria, we demonstrate the procedure for evaluating the relative skill of different HCS models. Finally, we propose an iterative algorithm for optimizing an initially stable HCS model with respect to independent, user-specified performance criteria, by applying the simultaneously perturbed stochastic approximation (SPSA) algorithm. The methods proposed herein offer a means for modeling and transferring HCS in response to real-time inputs, and improving the intelligent behaviors of artificial machines. © 2002 Published by Elsevier Science B.V.

Keywords: Human control strategy; Simultaneously perturbed stochastic approximation; Node-decoupled extended Kalman filtering

1. Introduction

HCS models, which accurately emulate dynamic human behavior, find application in a number of research areas ranging from robotics to the intelligent vehicle highway system. Because human control strategy (HCS) is a dynamic, nonlinear, stochastic process, developing good analytic models of human actions, however, tends to be quite difficult, if not impossible. Therefore, recent work in modeling HCS has focused on learning empirical models, through, e.g., fuzzy logic [7,16], and neural network techniques [1,8,11]. See [2,4,6,8], for detailed surveys of the human modeling literature.

Since most HCS models are empirical, few if any guarantees exist about their theoretical performance. In previous work, a stochastic similarity measure,

* Corresponding author.
E-mail address: nechyba@mil.ufl.edu (M.C. Nechyba).

0921-8890/02/$ – see front matter © 2002 Published by Elsevier Science B.V.
PII: S0921-8890(02)00169-0

which compares model-generated control trajectories to the original human training data, has been proposed for validating HCS models [9]. While this similarity measure can ensure that a given HCS model adequately captures the driving characteristics of the human operator, it does not measure a particular model's skill or performance. In other words, it does not (nor can it) tell us which model is better or worse. Thus, performance evaluation forms an integral part of HCS modeling research, without which it becomes impossible to rank or prefer one HCS controller over another. Moreover, only when we have developed adequate performance criteria, can we hope to optimize the HCS models with respect to those performance criteria.

In general, skill or performance can be defined through a number of task-dependent as well as task-independent criteria. Some of these criteria may conflict with one another, and which is most appropriate for a given task depends in part on the specific goals of the task. Overall, there are two approaches for defining performance criteria: (1) event analysis and (2) inherent analysis.

In event analysis, we examine performance within the context of a some event. Consider the task of human driving, e.g., For this task we can define any number of performance criteria tied to specific events. In preliminary work [14], e.g., two such event-based criteria were defined, one based on the HCS model's ability to avoid sudden obstacles, and the second based on the HCS model's ability to negotiate tight turns in a safe and stable manner. Each of these performance measures tests the HCS model's performance outside the range of its training data.

In inherent analysis, we examine a given model's behavior on a more global scale. Once again, consider the task of human driving. For a given HCS model, we might be interested in such measures as average speed, passenger comfort, driving smoothness, and fuel efficiency. These measures are not based on any single event, but rather are aggregate measures of performance. In other words, they measure the inherent characteristics of a particular HCS model.

Performance evaluation is, however, only one part of the solution for effectively applying models of HCS. When performing a specified task, a human will often commit occasional errors and deviate randomly from some nominal trajectory. Any empirical learning algorithm will necessarily incorporate those problems in the learned model, and will consequently be less than optimal. Furthermore, control requirements may differ between humans and robots, where stringent power or force requirements often have to be met. A given individual's performance level, therefore, may or may not be sufficient for a particular application.

Hence, in this paper, we not only consider the problem of performance evaluation, but the additional problem of performance optimization. We propose an iterative optimization algorithm, based on simultaneously perturbed stochastic approximation (SPSA), for improving the performance of learned HCS models. This algorithm leaves the learned model's structure in tact, but tunes the parameters of the HCS model in order to improve performance. It requires no analytic formulation of performance, only two experimental measurements of a user-defined performance criterion per iteration. The initial HCS model serves as a good starting point for the algorithm, since it already generates stable control commands.

In this paper, we first introduce the dynamic graphic driving simulator from which we collect human control data and with which we investigate the modeling and evaluation of human control strategies. We then show how we model a given individual's driving control strategies using the flexible cascade neural network learning architecture. Next, we develop and test performance criteria specifically related to the task of human driving, where we apply both event-based, as well as inherent analysis. We then propose the iterative optimization algorithm for improving performance in the HCS models. Finally, we describe and discuss some experimental results of the optimization algorithm.

2. Experimental setup

HCS, as we define the term, encompasses a large set of human-controlled tasks. It is neither practical nor possible to investigate all of these tasks comprehensively. In this paper, we therefore look towards a prototypical control application the task of human driving to collect, model and analyze control strategy data from different human subjects.

Within the driving domain, we have a choice between simulated driving (i.e. driving through a simulator) and real driving. For our purposes, the ideal

control task should embody several desirable qualities. First, during the execution of the control task, the human subject must not be injured or harmed in any way. Second, the human subject should have prior experiences that will help him complete the control task successfully. Third, the control task should pose a significant challenge to the human controller. Finally, the task should be complex enough that it allows for variations in strategy across different individuals.

Let us examine real driving in the context of these four criteria (safety, prior experience, control difficulty and control strategy variations). First, unless we ask individuals to drive very conservatively, it is difficult to guarantee the safety of our human subjects in real driving experiments. If we do ask them to drive conservatively, however, the control task will not be very challenging; moreover, variations between individuals will be somewhat muted. Finally, with respect to prior experience, real driving measures up to the qualities we seek in our control task.

Simulated driving, on the other hand, differs from real driving in a number of important respects. Most importantly, the human subject poses no threat to himself or others while driving in the simulator, no matter how recklessly he chooses to drive. Consequently, unlike in real driving, we can challenge individuals to drive near the edge of their abilities. This produces driving control strategies that are richer and more complex than their real counterparts. Because of this increased complexity, the demonstrated control strategies will potentially exhibit greater variations from one individual to the next. Finally, while human subjects may not be familiar with respect to a specific driving simulator prior to testing, they can, as experienced drivers, transition from real driving to simulated driving with relative ease and efficiency.

With respect to our goal of modeling and analyzing human control strategies, simulated driving embodies more of the qualities which we desire. Thus, we choose simulated driving as our primary control task. We emphasize that in choosing simulated driving, we do not suggest that simulation is in general better than reality for experimentation. We only suggest that since the focus of this paper is the human control strategies themselves, a simulated task can be appropriate if it bears substantial resemblance to a comparable real task. We believe that our driving simulation environment does meet that criterion.

Thus, for this work, we collect human driving data from a real-time graphic simulator, whose interface is shown in Fig. 1. In the simulator, the human operator has independent control of the vehicle's steering as well as the brake and gas pedals. The simulated vehicle's dynamics are given by the following second-order nonlinear model [5]:

$$\ddot{\theta} = \frac{l_\mathrm{f} P_\mathrm{f}\delta + l_\mathrm{f} F_{\xi\mathrm{f}} - l_\mathrm{r} F_{\xi\mathrm{r}}}{I}, \tag{1}$$

$$\dot{v}_\xi = \frac{P_\mathrm{f}\delta + F_{\xi\mathrm{f}} + F_{\xi\mathrm{r}}}{m} - v_\eta\dot{\theta} - (\operatorname{sgn} v_\xi)c_\mathrm{d}v_\xi^2, \tag{2}$$

$$\dot{v}_\eta = \frac{P_\mathrm{f} + P_\mathrm{r} - F_{\xi\mathrm{f}}\delta}{m} + v_\xi\dot{\theta} - (\operatorname{sgn} v_\eta)c_\mathrm{d}v_\eta^2, \tag{3}$$

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}v_\xi\\ v_\eta\end{bmatrix}, \tag{4}$$

where

$$\dot{\theta} = \text{angular velocity of the car}, \tag{5}$$

$$v_\xi = \text{lateral velocity of the car}, \tag{6}$$

$$v_\eta = \text{longitudinal velocity of the car}, \tag{7}$$

$$F_{\xi k} = \mu F_{zk}\left(\tilde{\alpha}_k - \frac{(\operatorname{sgn}\delta)\tilde{\alpha}_k^2}{3} + \frac{\tilde{\alpha}_k^3}{27}\right) \times\sqrt{1 - \frac{P_k^2}{(\mu F_{zk})^2} + \frac{P_k^2}{c_k^2}}, \quad k \in \{f, r\}, \tag{8}$$

$$\tilde{\alpha}_k = \frac{c_k\alpha_k}{\mu F_{zk}}, \quad k \in \{f, r\}, \tag{9}$$

$$\alpha_\mathrm{f} = \text{front tire slip angle} = \delta - \frac{l_\mathrm{f}\dot{\theta} + v_\xi}{v_\eta}, \tag{10}$$

$$\alpha_\mathrm{r} = \text{rear tire slip angle} = \frac{l_\mathrm{r}\dot{\theta} - v_\xi}{v_\eta}, \tag{11}$$

$$F_{z\mathrm{f}} = \frac{\mathrm{mgl_r} - (P_\mathrm{f} + P_\mathrm{r})h}{l_\mathrm{f} + l_\mathrm{r}}, \qquad F_{z\mathrm{r}} = \frac{\mathrm{mgl_f} + (P_\mathrm{f} + P_\mathrm{r})h}{l_\mathrm{f} + l_\mathrm{r}}, \tag{12}$$

$$\xi, \eta = \text{body-relative lateral, longitudinal axis}, \tag{13}$$

$$c_\mathrm{f}, c_\mathrm{r} = 50\,000,\ 64\,000\ \mathrm{N/rad}, \tag{14}$$

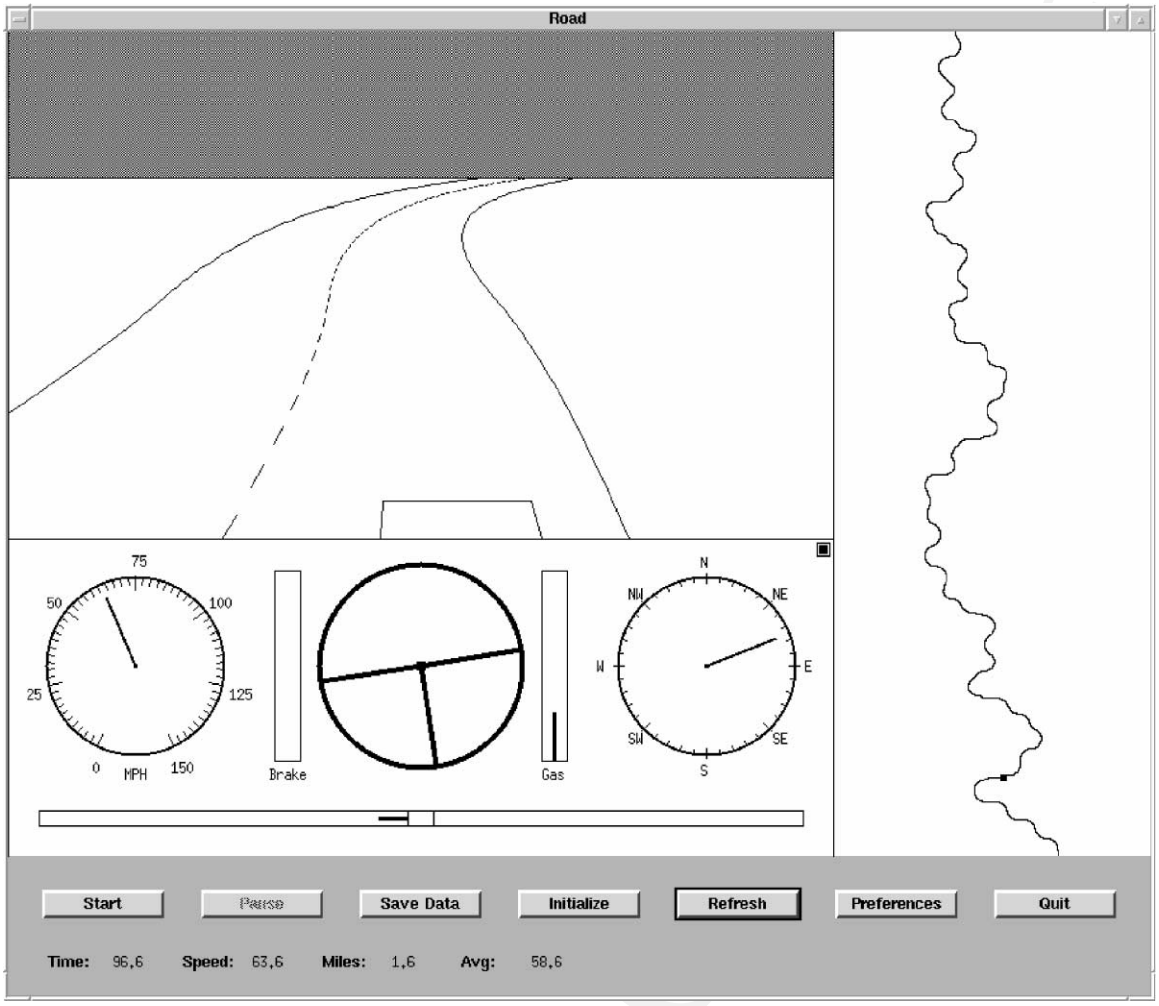

Fig. 1. The driving simulator gives the user a perspective preview of the road ahead. The user has independent control of the steering, break, and accelerator (gas).

$$c_d = \text{air resistance} = 0.0005\,\text{m}^{-1}, \tag{15}$$

where $\mu =$ coefficient of friction which is equal to 1.

$$F_{jk} = \text{frictional forces}, \quad j \in \{\xi, \eta\}, \quad k \in \{f, r\}, \tag{16}$$

$$P_r = \begin{cases} 0, & P_f \geq 0, \\ k_b P_f, & P_f < 0 \end{cases}, \quad k_b = 0.34, \tag{17}$$

$$m = 1500\,\text{kg}, \qquad I = 2500\,\text{kg}\,\text{m}^{-2},$$
$$l_f = 1.25\,\text{m}, \qquad l_r = 1.5\,\text{m}, \qquad h = 0.5\,\text{m}, \tag{18}$$

and the controls are given by

$$-8000\,\text{N} \leq P_f \leq 4000\,\text{N}, \tag{19}$$

$$-0.2\,\text{rad} \leq \delta \leq 0.2\,\text{rad}, \tag{20}$$

where P_f is the longitudinal force on the front tires, and δ the steering angle.

We ask each individual to navigate across several randomly generated roads, which consist of a sequence of: (1) straight-line segments, (2) left turns, and (3) right turns. The map in Fig. 1, e.g., illustrates one randomly generated 20 km road for which human driving

data was recorded. Each straight-line segment as well as the radius of curvature for each turn range in length from 100 to 200 m. Nominally, the road is divided into two lanes, each of which has width $w = 5$ m. The human operator's view of the road ahead is limited to 100 m. Finally, the entire simulator is run at 50 Hz.

3. HCS modeling

In this paper, we choose the flexible cascade neural network architecture with node-decoupled extended Kalman filtering (NDEKF) [10] for modeling the human driving data. We prefer this learning architecture over others for a number of reasons. First, no a priori model structure is assumed; the neural network automatically adds hidden units to an initially minimal network as the training requires. Second, hidden unit activation functions are not constrained to be a particular type. Rather, for each new hidden unit, the incremental learning algorithm can select that functional form which maximally reduces the residual error over the training data. Typical alternatives to the standard sigmoidal function are sine, cosine, and the Gaussian function. Finally, it has been shown that NDEKF, a quadratically convergent alternative to slower gradient descent training algorithms (such as backpropagation) fits well within the cascade learning framework and converges to good local minima in less time [10].

The flexible functional form which cascade learning allows is ideal for abstracting human control strategies, since we know very little about the underlying structure of each individual's internal controller. By making as few a priori assumptions as possible in modeling the human driving data, we improve the likelihood that the learning algorithm will converge to a good model of the human control data.

In order for the learning algorithm to properly model each individual's HCS, the model must be presented with those state and environmental variables upon which the human operator relies. Thus, the inputs to the cascade neural network should include: (1) current and previous state information $\{v_\xi, v_\eta, \dot{\theta}\}$, (2) previous output (command) information $\{\delta, P_f\}$, and (3) a description of the road visible from the current car position. More precisely, the network inputs are

$$\{v_\xi(k-n_s), \ldots, v_\xi(k-1), v_\xi(k), v_\eta(k-n_s), \ldots, v_\eta(k-1), v_\eta(k), \dot{\theta}(k-n_s), \ldots, \dot{\theta}(k-1), \dot{\theta}(k)\}, \quad (21)$$

$$\{\delta(k-n_c), \ldots, \delta(k-1), \delta(k), P_f(k-n_c), \ldots, P_f(k-1), P_f(k)\}, \quad (22)$$

$$\{x(1), x(2), \ldots, x(n_r), y(1), y(2), \ldots, y(n_r)\}, \quad (23)$$

where n_s is the length of the state histories and n_c the length of the previous command histories presented to the network as input. For the road description, we partition the visible view of the road ahead into n_r equivalently spaced, body-relative (x, y) coordinates of the road median, and provide that sequence of coordinates as input to the network. Thus, the total number of inputs to the network n_i is

$$n_i = 3n_s + 2n_c + 2n_r. \quad (24)$$

The two outputs of the cascade network are $\{\delta(k+1), P_f(k+1)\}$. For the system as a whole, the cascade neural network can be viewed as a feedback controller, whose two outputs control the vehicle.

4. Performance criteria based on event analysis

Once we have abstracted models of driving control strategies from the human control data, we would like to evaluate the skill or performance exhibited by these models. The first set of performance measures that we develop is based on the observation that, in real driving, obstacles such as rocks and debris can unexpectedly obstruct a vehicle's path and force drivers to react rapidly. In order to gauge how well our learned models would deal with these types of events, we define two related performance criteria. The first measures a model's ability to avoid obstacles, while the second measures a model's capacity for negotiating tight turns.

4.1. Obstacle avoidance

Obstacle avoidance is one important measuring stick for gauging a model's performance. Since our

HCS models receive only a description of the road ahead as input from the environment, we reformulate the task of obstacle avoidance as *virtual path following*. Assume that an obstacle appears a distance τ ahead of the driver's current position. Furthermore, assume that this obstacle obstructs the width of the road ($2w$) and extends for a distance d along the road. Then, rather than follow the path of the actual road, we wish the HCS model to follow the virtual path illustrated in Fig. 2. This virtual path consists of: (1) two arcs with radius of curvature γ, which offset the road median laterally by $2w$, followed by (2) a straight-line segment of length d, and (3) another two arcs with radius of curvature γ which return the road median to the original path.

By analyzing the geometry of the virtual path, we can calculate the required radius of curvature γ of the virtual path segments as [14],

$$\gamma = \frac{\tau^2}{8w} + \frac{w}{2}, \tag{25}$$

and the corresponding sweep angle ρ as,

$$\rho = \sin^{-1}\left(\frac{\tau/2}{\gamma}\right) = \sin^{-1}\left(\frac{\tau}{\tau^2/4w + w}\right). \tag{26}$$

As an example, consider an obstacle located $\tau = 60\,\text{m}$ ahead of the driver's current position. For this obstacle distance and $w = 5\,\text{m}$, γ evaluates to 92.5 m. This is less than the minimum radius of curvature (100 m) that we allow for the roads over which we collect our human control data. Hence, a particular HCS model may deviate significantly from the center of the road during the obstacle avoidance maneuver.

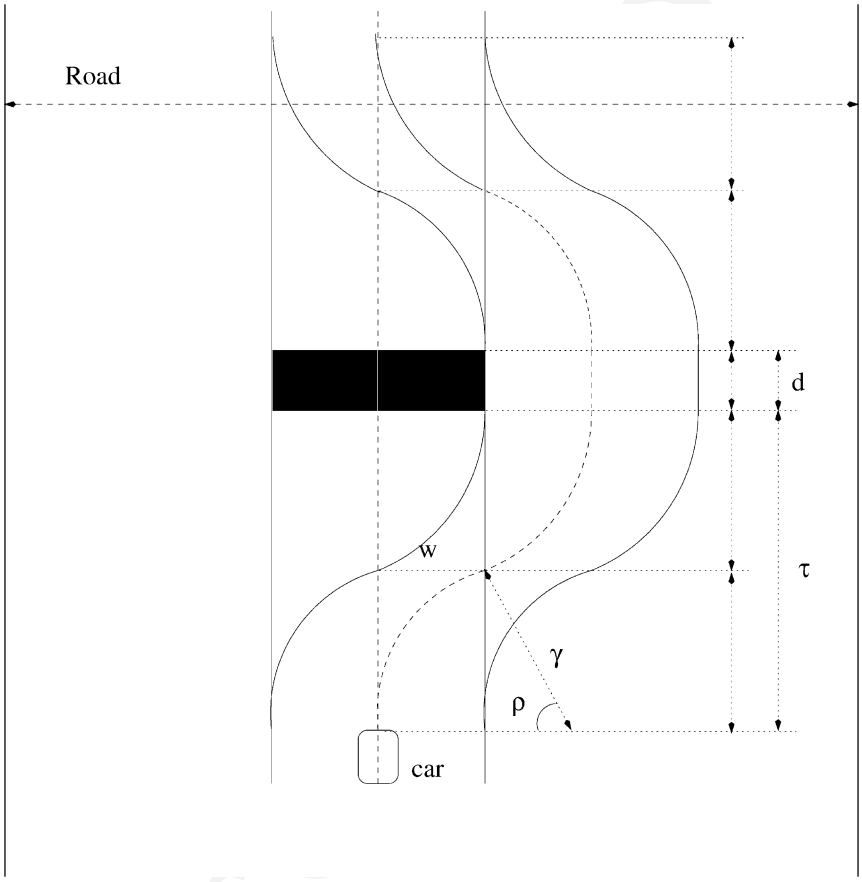

Fig. 2. Virtual path for obstacle avoidance.

Below, we derive the important relationship between the obstacle detection distance τ and a model's corresponding maximum lateral deviation ψ. First, we take N measurements of ψ for different values of τ, where we denote the ith measurement as (τ_i, ψ_i). Next, we assume a polynomial relationship of the form

$$\begin{aligned}\psi_i &= \alpha_p \tau_i^p + \alpha_{p-1}\tau_i^{p-1} + \cdots + \alpha_1\tau_i + \alpha_0 + e_i \\ &= \Gamma_i^{\mathrm{T}}\alpha + e_i,\end{aligned} \tag{27}$$

where e_i is the additive measurement error. We can then write

$$\begin{aligned}&\psi_1 = \Gamma_1^{\mathrm{T}}\alpha + e_1,\\ &\psi_2 = \Gamma_2^{\mathrm{T}}\alpha + e_2,\\ &\vdots\\ &\psi_N = \Gamma_N^{\mathrm{T}}\alpha + e_N,\end{aligned} \tag{28}$$

or, in matrix notation,

$$\Psi = \Gamma\alpha + e, \tag{29}$$

where $\Psi = [\psi_1, \psi_2, \ldots, \psi_N]^{\mathrm{T}}$ is the observation vector, $\Gamma = [\Gamma_1, \Gamma_2, \ldots, \Gamma_N]^{\mathrm{T}}$ the regression matrix, and $e = [e_1, e_2, \ldots, e_N]^{\mathrm{T}}$ the error vector.

Assuming white noise properties for e ($E\{e_i\} = 0$ and $E\{e_i e_j\} = \sigma_e^2\delta_{ij}$ for all i, j), we can minimize the least-squares error criterion,

$$V(\hat{\alpha}) = \frac{1}{2}\varepsilon^{\mathrm{T}}\varepsilon = \frac{1}{2}\sum_{k=1}^{N}\varepsilon_k^2 = \frac{1}{2}(\Psi - \Gamma\hat{\alpha})^{\mathrm{T}}(\Psi - \Gamma\hat{\alpha}) \tag{30}$$

with the optimal, unbiased estimate $\bar{\alpha}$,

$$\bar{\alpha} = (\Gamma^{\mathrm{T}}\Gamma)^{-1}\Gamma^{\mathrm{T}}\Psi \tag{31}$$

assuming that $(\Gamma^{\mathrm{T}}\Gamma)$ is invertible.

In this relationship, as the obstacle detection distance τ decreases, the maximum lateral offset increases [14]. Consequently, for a given model and initial velocity v_{initial}, there exists a value $\tau_{\min}$ below which the maximum offset error will exceed the lane width w. We define the driving control for obstacle distances above $\tau_{\min}$ to be stable; likewise, we define the driving control to be unstable for obstacle distances below $\tau_{\min}$.

Now, we define the following obstacle avoidance performance criterion J_1:

$$J_1 = \frac{\tau_{\min}}{v_{\text{initial}}}, \tag{32}$$

where v_{initial} is the velocity of the vehicle when the obstacle is first detected. The J_1 criterion measures to what extent a given HCS model can avoid an obstacle while still controlling the vehicle in a stable manner. The normalization by v_{initial} is required, because slower speeds increase the amount of time a driver has to react and therefore avoiding obstacles becomes that much easier.

4.2. Tight turning

Here we analyze performance by how well a particular HCS model is able to navigate tight turns. First, we define a special road connection consisting of two straight-line segments connected directly (without a transition arc segment) at an angle ζ. For small values of ζ, each HCS model will be able to successfully drive through the tight turn; for larger values of ζ, however, some models will fail to execute the turn properly by temporarily running off the road or losing complete sight of the road.

Fig. 3 illustrates, e.g., how one HCS model transitions through a tight turn for $\zeta = 5\pi/36$ rad. Fig. 3(a) plots the two straight-line segments connected at an angle ζ. The solid line describes the road median, while the dashed line describes the actual trajectory executed by Harry's HCS model. The length of the initial straight-line segment is chosen to be long enough (150 m) to eliminate transients by allowing the model to settle into a stable state. This is equivalent to allowing the vehicle to drive on a straight road for a long period of time before the tight turn appears in the road. Fig. 3(b) plots the lateral offset from the road median during the tight-turn maneuver. Here, Harry's model maximally deviates about 8 m from the road center. Both before and after the turn, the lateral offset converges to zero. Fig. 3(c) plots the commanded steering angle for Harry's HCS model, and Fig. 3(d) plots the corresponding change in velocity. Models for other drivers yield similar results.

Now, define the maximum lateral offset error corresponding to a tight turn with angle ζ to be ψ. We can determine a functional relationship between ψ and ζ for a given HCS model. First, we take N measurements of ρ for different values of ζ where we denote the ith measurement as (ζ_i, ψ_i). Then, we assume a polynomial relationship between ψ and ζ such

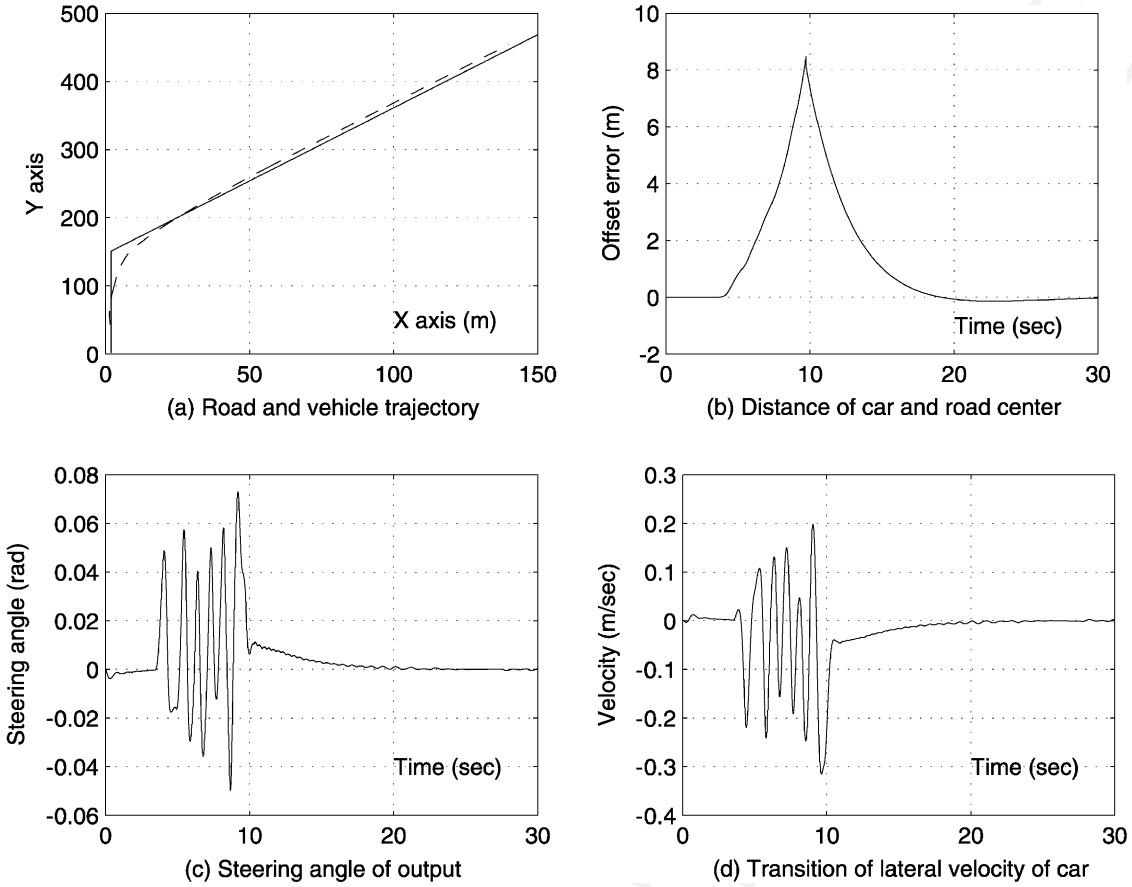

Fig. 3. Example model driving behavior through a tight turn.

that,

$$\psi_i = \alpha_p \zeta_i^p + \alpha_{p-1}\zeta_i^{p-1} + \cdots + \alpha_1\zeta_i + \alpha_0 + e_i, \tag{33}$$

The least-squares estimate of the model ($\hat{\alpha}$) is given by

$$\hat{\alpha} = (\hat{\zeta}^{\mathrm{T}}\hat{\zeta})^{-1}\hat{\zeta}^{\mathrm{T}}\hat{\psi}, \tag{34}$$

where

$$\hat{\psi} = [\psi_1, \psi_2, \ldots, \psi_N]^{\mathrm{T}}, \tag{35}$$

$$\hat{\zeta} = \begin{bmatrix} \zeta_1^p & \zeta_1^{p-1} & \cdots & \zeta_1 & 1 \\ \zeta_2^p & \zeta_2^{p-1} & \cdots & \zeta_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \zeta_N^p & \zeta_N^{p-1} & \cdots & \zeta_N & 1 \end{bmatrix}, \tag{36}$$

$$\hat{\alpha} = [\alpha_p, \alpha_{p-1}, \ldots, \alpha_0]^{\mathrm{T}}. \tag{37}$$

Previously, we have observed that the linear coefficient α_1 dominates the polynomial relationship in Eq. (33) [14]. Hence, as a first-order approximation, we define the following tight-turning performance criterion J_2:

$$J_2 = \alpha_1. \tag{38}$$

5. Performance criteria based on inherent analysis

In the previous section, we introduced performance criteria based on specific events. In this section, we now will investigate other performance criteria which evaluate the inherent characteristics of HCS models through analysis of the whole driving process.

5.1. Passenger comfort

Passenger comfort is one important criterion for evaluating driving control strategies. Suppose a person were sitting in a car driven by a learned HCS model. His/her comfort, while a combination of many factors, would be primarily influenced by the forces that that passenger experiences while in the car. Everytime the HCS model would change the applied force P_f on the car, the passenger would feel a longitudinal force. Similarly, everytime the HCS model would change the steering δ, the passenger would experience a lateral force. Below we quantify passenger comfort as a function of the applied forces on the vehicle under HCS model control.

Consider the vehicle shown in Fig. 4. Let the configuration of the system be described by the mass center of the vehicle (x, y), the angle θ between the positive Y-axis and the axis of symmetry of the car, and the location of the passenger S (x_s, y_s). Furthermore, define the distance from S to the axis of symmetry as s_2 and define the distance from S to the center of mass along the axis of symmetry as s_1.

The velocity of the point S as a function of the coordinate velocities is given by

$$\begin{aligned} v_s^2 &= \dot{x}_s^2 + \dot{y}_s^2 = (\dot{x} + s_1\dot{\theta}\cos\theta + s_2\dot{\theta}\sin\theta)^2 \\ &\quad +(\dot{y} - s_1\dot{\theta}\sin\theta + s_2\dot{\theta}\cos\theta)^2 \\ &= \dot{x}^2 + s_1^2\dot{\theta}^2\cos^2\theta + s_2^2\dot{\theta}^2\sin^2\theta + 2\dot{x}s_1\dot{\theta}\cos\theta \\ &\quad +2s_2\dot{x}\dot{\theta}\sin\theta + 2s_1s_2\dot{\theta}^2\cos\theta\sin\theta + \dot{y}^2 \\ &\quad +s_1^2\dot{\theta}^2\sin^2\theta + s_2^2\dot{\theta}^2\cos^2\theta - 2s_1\dot{y}\dot{\theta}\sin\theta \\ &\quad +2s_2\dot{y}\dot{\theta}\cos\theta - 2s_1s_2\dot{\theta}^2\sin\theta\cos\theta \\ &= \dot{x}^2 + \dot{y}^2 + s_1^2\dot{\theta}^2 + s_2^2\dot{\theta}^2 + 2s_1\dot{\theta}(\dot{x}\cos\theta \\ &\quad -\dot{y}\sin\theta) + 2s_2\dot{\theta}(\dot{x}\sin\theta + \dot{y}\cos\theta). \end{aligned} \tag{39}$$

As we described in Section 2, the longitudinal acceleration of the vehicle is given by

$$\dot{v}_\eta = \frac{P_f + P_r - F_{\xi f}\delta}{m} + v_\xi\dot{\theta} - (\mathrm{sgn}\, v_\eta)c_d v_\eta^2, \tag{40}$$

and the lateral acceleration of the vehicle is given by

$$\dot{v}_\xi = \frac{P_f\delta + F_{\xi f} + F_{\xi r})}{m} - v_\eta\dot{\theta} - (\mathrm{sgn} v_\xi)c_d v_\xi^2. \tag{41}$$

Now, the accelerations experienced by the passenger include not only the vehicle's acceleration, but also the centrifugal force, given by

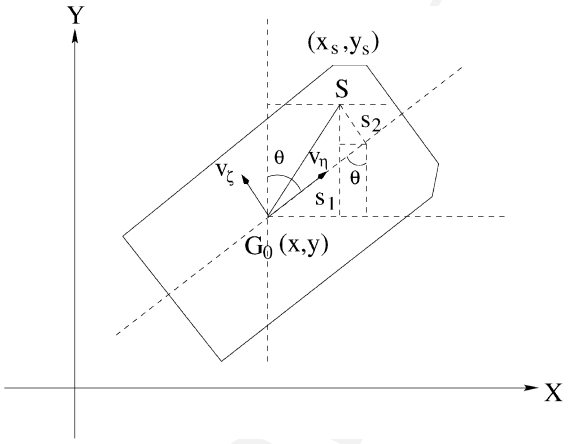

Fig. 4. The coordinate configuration of the vehicle and passenger.

$$\begin{aligned} \frac{v_s^2}{R} &= \frac{1}{R}[\dot{x}^2 + \dot{y}^2 + s_1^2\dot{\theta}^2 + s_2^2\dot{\theta}^2 + 2s_1\dot{\theta}(\dot{x}\cos\theta \\ &\quad -\dot{y}\sin\theta) + 2s_2\dot{\theta}(\dot{x}\sin\theta + \dot{y}\cos\theta)]. \end{aligned} \tag{42}$$

The centrifugal force generally points in the direction of the negative lateral acceleration of the vehicle. By combining the vehicle and centrifugal accelerations, we then arrive at the following expression for the total acceleration at point S:

$$a = \sqrt{\dot{v}_\eta^2 + \left(\dot{v}_\xi - \frac{v_0^2}{R}\right)^2}. \tag{43}$$

In defining a "comfort" performance criterion J_3, we will normalize this acceleration felt by the passenger, by the speed of the vehicle, since higher speeds generate higher accelerations through a given curve:

$$J_3 = \frac{a_{\mathrm{mean}}}{v_{\mathrm{mean}}}. \tag{44}$$

Thus, J_3 is defined as the ratio of average acceleration over average speed for a given road.

Let us now look at how different HCS models perform with respect to this performance criterion. First, we collect driving data from three human operators—Tom, Dick and Harry. After training an HCS model for each individual, we then run that person's model over three different roads (1, 2, and 3). Each run takes approximately 15 min over the 20 km roads. That means

Table 1
The statistic of the acceleration

Data name	$a > g$ (9.8 m/s^2)	$a > 2g$ (19.6 m/s^2)	$a > 3g$ (29.4 m/s^2)	$\rho = \frac{a_{\text{mean}}}{V_{\text{mean}}}$
Tom1-A	28.20%	3.85%	0.39%	0.2411
Tom1-B	20.99%	0.79%	0	0.2101
Tom2-A	19.11%	0.74%	0	0.2142
Tom2-B	18.00%	0.63%	0	0.2185
Tom3-A	22.15%	0.21%	0	0.2117
Tom3-B	25.76%	0.31%	0	0.2196
Dick1-A	36.43%	8.69%	1.63%	0.2938
Dick1-B	36.85%	6.79%	0	0.2625
Dick2-A	37.73%	8.87%	2.19%	0.2956
Dick2-B	37.25%	7.31%	0.43%	0.2721
Dick3-A	43.20%	15.17%	5.13%	0.3367
Dick3-B	43.35%	14.97%	4.61%	0.3307
Harry1-A	5.93%	0.48%	0	0.1662
Harry1-B	8.13%	1.30%	0.013%	0.1941
Harry2-A	1.16%	0	0	0.1302
Harry2-B	2.69%	0	0	0.1267
Harry3-A	4.93%	0	0	0.1552
Harry3-B	11.49%	1.28%	0	0.1757

that at a data collection rate of 50 Hz, each run consists of approximately 45 000 time-sampled data vectors. In other words, for each model run, we collect approximately 135 000 data vectors. After data collection, we split the three runs for each driver into two groups A and B, where group A represents the first half of each run, while group B represents the second half of each run. Thus, e.g., 'Tom1-A' represents the first half of Tom's HCS model's run over road 1.

Table 1 gives some aggregate statistics for each of these model-generated data sets. Specifically, the table lists the percentage of time that the acceleration in a particular data set is larger than one g, $2g$ and $3g$, respectively. These percentages give us a rough idea about the comfort level of each model driver. If we average the percentages for each HCS model, we find that Tom's model generates accelerations above one g 22.36% of the time, accelerations above $2g$ 1.09% of the time, and accelerations above $3g$ 0.065% of the time. The same statistics for Dick's model are 39.14, 10.30 and 2.33%, respectively. Similarly, for Harry's model the statistics are 5.72, 0.51 and 0.00%, respectively. From these results, we would expect that Harry's HCS model offers the smoothest ride of the three models, since it generates the smallest forces. Driving with Dick's model, on the other hand, would prove to be quite uncomfortable. Calculating the J_3 performance criterion for each model confirms these qualitative observations. For Tom's model, J_3 varies from 0.2101 to 0.2411, and the average is given by

$$J_{3\text{Tom}} = 0.2192. \quad (45)$$

For Dick's model, J_3 varies from 0.2625 to 0.3367, and the average is given by

$$J_{3\text{Dick}} = 0.2986. \quad (46)$$

Finally, for Harry's model, J_3 varies from 0.1303 to 0.1941, and the average is given by

$$J_{3\text{Harry}} = 0.1508. \quad (47)$$

We observe that J_3 is the smallest for Harry's model, and that that value is much smaller than J_3 for Dick's model.

5.2. Driving smoothness

Another way to evaluate the smoothness of a given driver's control strategy is through frequency analysis of the instantaneous curvature of the road and the corresponding instantaneous curvature of the vehicle's path. As an HCS model steers the car along the road, the vehicle's curvature will in general not be the same

as the that of the road. Below, we will use this difference between the two curvatures to evaluate the driving smoothness of a given model in the frequency domain. We will show that the resulting performance measure yields consistent results with the J_3 passenger comfort performance criterion defined in the previous section.

Let us define $u(k)$ as the instantaneous curvature of the road at time step k, and let $z(k)$ be the instantaneous curvature of the vehicle's path at time step k. We can view the road's curvature $u(k)$ as the input to the HCS model, and $z(k)$ as the output of the HCS model.

To calculate the frequency response from u to z, we first partition the complete data into N groups, where each group is of length L. Hence, the kth element of group i is given by

$$u_i(k) = u[k + (i-1)L],$$
$$z_i(k) = z[k + (i-1)L],$$
$$i = 1, 2, \ldots, N; 1 \le k \le L. \tag{48}$$

We also define the following convolutions for each group of data i:

$$I_{u_i,L}(w) = \frac{1}{L} U_i(jw) U_i^*(jw) = \frac{1}{L} ||U_i(jw)||^2,$$
$$I_{u_i z_i,L}(jw) = \frac{1}{L} U_i(jw) Z_i^*(jw), \quad i = 1, 2, \ldots, N, \tag{49}$$

where

$$U_i(jw) = \sum_{k=1}^{L} u_i(k) H_k \, \mathrm{e}^{-\mathrm{j}wk}, \tag{50}$$

$$Z_i(jw) = \sum_{k=1}^{L} z_i(k) H_k \, \mathrm{e}^{-\mathrm{j}wk} \tag{51}$$

define the discrete Fourier transform [13] and,

$$H_k = 0.54 - 0.46 \cos\left[\frac{2\pi(k-1)}{L-1}\right],$$
$$k \in \{1, 2, \ldots, L\} \tag{52}$$

defines the Hamming coefficients, which we include to minimize the spectral leakage effects of data windowing.

By summing up the terms in Eq. (50),

$$S_{u,L}(w) = \frac{1}{N} \sum_{i=1}^{N} I_{u_i,L}(w), \tag{53}$$

$$S_{uz,L}(jw) = \frac{1}{N} \sum_{i=1}^{N} I_{u_i,L}(jw), \tag{54}$$

we define the frequency response $G(jw)$ for a given HCS model as

$$G(jw) = \frac{S_{uz,L}(jw)}{S_{u,L}(w)}. \tag{55}$$

Fig. 5 plots $|G(jw)|$ for the HCS models corresponding to Tom, Dick and Harry, Each group of data corresponds to 40 s ($L = 2000$ at 50 Hz), and the data for each model was collected over road 1. In Fig. 5 the solid line corresponds to Tom, the dash-dotted line corresponds to Dick, and the dashed line corresponds to Harry.

Given the plots of $|G(jw)|$, we now define the following smoothness performance criterion:

$$J_4 = f_{\text{domain}}, \tag{56}$$

where f_{domain} corresponds to the domain frequency of each $|G(jw)|$ curve.

We get the following smoothness results for the three models:

$$J_{\text{Harry}} = 0.52\,\text{Hz}, \tag{57}$$

$$J_{\text{Tom}} = 0.66\,\text{Hz}, \tag{58}$$

$$J_{\text{Dick}} = 0.72\,\text{Hz}. \tag{59}$$

Note that these results agree with the J_3 passenger comfort criterion defined in the previous section. Harry's model was found to offer the best passenger comfort, and here, his model is found to offer the smoothest ride. Similarly, Dick's model was found to be the least comfortable and here, his model is found to be the least smooth.

6. Performance optimization

In Sections 4 and 5, we introduced performance measures for evaluating the performance of our driving models. Below, we develop an algorithm for

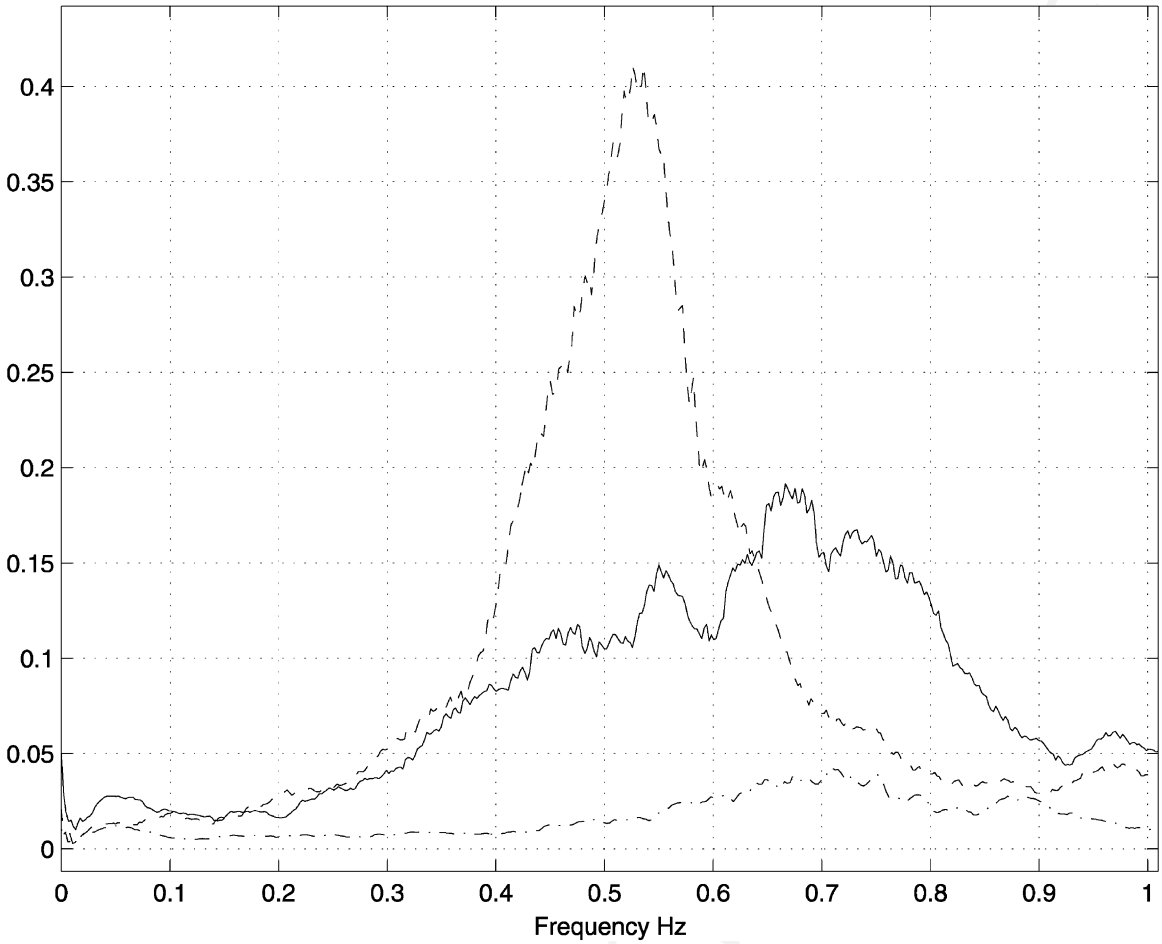

Fig. 5. PSD analysis with Harry (dashed), Dick (dash-dot), and Tom (solid).

optimizing a learned control strategy model with respect to one of those (or for that matter, any other) performance criterion. There are two primary reasons why this may be necessary in order to successfully transfer control strategies from humans to robots.

First, while humans are in general very capable of demonstrating intelligent behaviors, they are far less capable of demonstrating those behaviors without occasional errors and random (noise) deviations from some nominal trajectory. The cascade learning algorithm will necessarily incorporate those in the learned HCS model, and will consequently be less than optimal. Second, control requirements may differ between humans and robots, where, e.g., stringent power or force requirements often have to be met. Thus, a given individual's performance level may or may not be sufficient for a particular application.

Since an HCS model does offer an initially stable model, however, it represents a good starting point from which to further optimize performance. Let

$$\omega = [\, w_1 \quad w_2 \quad \cdots \quad w_n \,] \tag{60}$$

denote a vector consisting of all the weights in the trained HCS model $\Gamma(\omega)$. Also let $J(\omega)$ denote any one performance criterion (e.g., J_1 or J_2 in the previous sections). We would now like to determine the weight vector ω^* which optimizes the performance criterion $J(\omega)$. This optimization is difficult

in principle because: (1) we have no explicit gradient information

$$G(\omega) = \frac{\partial}{\partial \omega} J(\omega), \tag{61}$$

and (2) each experimental measurement of $J(\omega)$ requires a significant amount of computation. We lack explicit gradient information, since we can only compute our performance measures empirically. Hence, gradient-based optimization techniques, such as steepest descent and Newton–Raphson [13] are not suitable. And because each performance measure evaluation is potentially computationally expensive, genetic optimization [3], which can requires many iterations to converge, also does not offer a good alternative. Therefore, we turn to SPSA to carry out the performance optimization.

Stochastic approximation (SA) is a well-known iterative algorithm for finding roots of equations in the presence of noisy measurements. SPSA [15] is a particular multivariate SA technique which requires as few as two measurements per iteration and shows fast convergence in practice. Hence, it is well suited for our application. Denote ω_k as our estimate of ω^* at the kth iteration of the SA algorithm, and let ω_k be defined by the following recursive relationship:

$$\omega_{k+1} = \omega_k - \alpha_k \bar{G}_k, \tag{62}$$

where $\bar{G}_k$ is the simultaneously perturbed gradient approximation at the kth iteration,

$$\bar{G}_k = \frac{1}{p} \sum_{i=1}^{p} G_k^i \approx \frac{\partial}{\partial \omega} J(\omega), \tag{63}$$

$$G_k^i = \frac{J_k^{(+)} - J_k^{(-)}}{2c_k} \begin{bmatrix} \frac{1}{\Delta_{kw_1}} \\ \frac{1}{\Delta_{kw_2}} \\ \cdots \\ \frac{1}{\Delta_{kw_n}} \end{bmatrix}. \tag{64}$$

Eq. (63) averages p stochastic two-point measurements G_k^i for a better overall gradient approximation, where

$$J_k^{(+)} = J(\omega_k + c_k \Delta_k), \tag{65}$$

$$J_k^{(-)} = J(\omega_k - c_k \Delta_k), \tag{66}$$

$$\Delta_k = [\Delta_{kw_1} \Delta_{kw_2} \cdots \Delta_{kw_n}]^{\mathrm{T}}, \tag{67}$$

and where Δ_k is a vector of mutually independent, mean-zero random variables (e.g., symmetric Bernoulli distributed), the sequence $\{\Delta_k\}$ is independent and identically distributed, and the $\{\alpha_k\}$, $\{c_k\}$ are positive scalar sequences satisfying the following properties:

$$\alpha_k \to 0, \qquad c_k \to 0 \text{ as } k \to \infty, \tag{68}$$

$$\sum_{k=0}^{\infty} \alpha_k = \infty, \qquad \sum_{k=0}^{\infty} \left(\frac{\alpha_k}{c_k} \right)^2 < \infty. \tag{69}$$

The weight vector ω_0 is of course the weight representation in the initially stable learned cascade model. Larger values of p in Eq. (63) will give more accurate approximations of the gradient. Fig. 6 illustrates the overall performance optimization algorithm.

7. Experiment

7.1. Results

Here, we test the performance optimization algorithm on control data collected from two individuals, Harry and Dick. In order to simplify the problem somewhat, we keep the applied force constant at $P_{\mathrm{f}} = 300\,\mathrm{N}$. Hence, the user is asked to control only the steering δ.

For each person, we train a two-hidden-unit HCS model with $n_{\mathrm{s}} = n_{\mathrm{c}} = 3$, and $n_{\mathrm{r}} = 15$; because we are keeping P_{f} constant, the total number of inputs for the neural network models is therefore $n_i = 42$.

Now, we would like to improve the tight-turning performance criterion J_2 defined in Eq. (38) for each of the trained models. In the SPSA algorithm, we empirically determine the following values for the scaling sequences $\{\alpha_k\}$, $\{c_k\}$:

$$\alpha_k = \frac{0.000001}{k}, \quad k > 0, \tag{70}$$

$$c_k = \frac{0.001}{k^{0.25}}, \quad k > 0. \tag{71}$$

We also set the number of measurements per gradient approximation in Eq. (63) to $p = 1$. Finally, denote J_2^k as the criterion J_2 after iteration k of the optimization

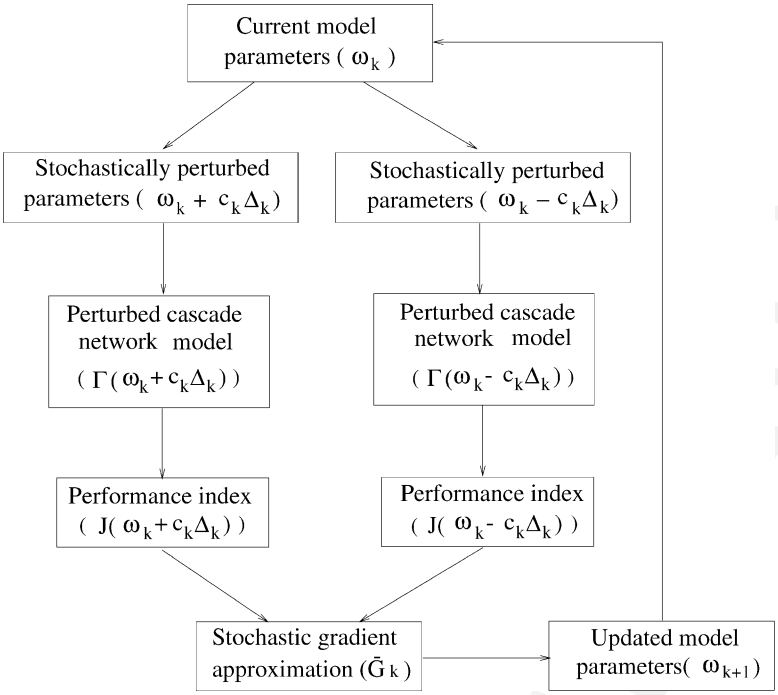

Fig. 6. Stochastic optimization algorithm.

algorithm; hence, J_2^0 denotes the performance measure prior to any optimization.

Fig. 7 plots $100 \times J_2^k / J_2^0$, $0 \leq k \leq 60$, for the HCS models corresponding to Dick and Harry. We note that for Dick, the performance index J_2 improves from $J_2^0 = 25.5$ to $J_2^{60} = 12.5$. For Harry, the improvement is less dramatic; his model's performance index improve from $J_2^0 = 17.7$ to $J_2^{60} = 16.1$. Thus, the performance optimization algorithm is able to improve the performance of Dick's model by about 55% and Harry's model by about 9% over their respective initial models. In other words, the optimized models negotiate tight turns better without running off the road. From Fig. 7, we observe that most of the improvement in the optimization algorithm occurs in the first few iterations. Then, as $k \to \infty$, J_2^k converges to a stable value since $\alpha_k, c_k \to 0$. Clearly, the extent to which we can improve the performance in the trained HCS models depends on the characteristics of the original models. Dick's initial performance index of $J_2^0 = 25.5$ is much worse than Harry's initial performance index of $J_2^0 = 17.7$. Therefore, we would expect that Dick's initial model lies further away from the nearest local minimum, while Harry's model lies closer to that local minimum. As a result, Harry's model can be improved only a little, while Dick's model has much larger room for improvement.

7.2. Discussion

Below we discuss some further issues related to performance optimization including: (1) the effect of performance optimization on other performance criteria, and (2) the similarity of control strategies before and after performance optimization.

First, we show how performance improvement with respect to one criterion can potentially affect performance improvement with respect to a different criterion. Consider Dick's HCS model once again. As we have already observed, his tight turning performance criterion improves from $J_2^0 = 25.5$ to $J_2^{60} = 12.5$. Now, let J_1^0 denote the obstacle avoidance performance criterion for Dick's initial HCS model, and let J_1^{60} denote the obstacle avoidance performance criterion for Dick's HCS model, optimized with respect to J_2. Fig. 8 plots the maximum offset from the road

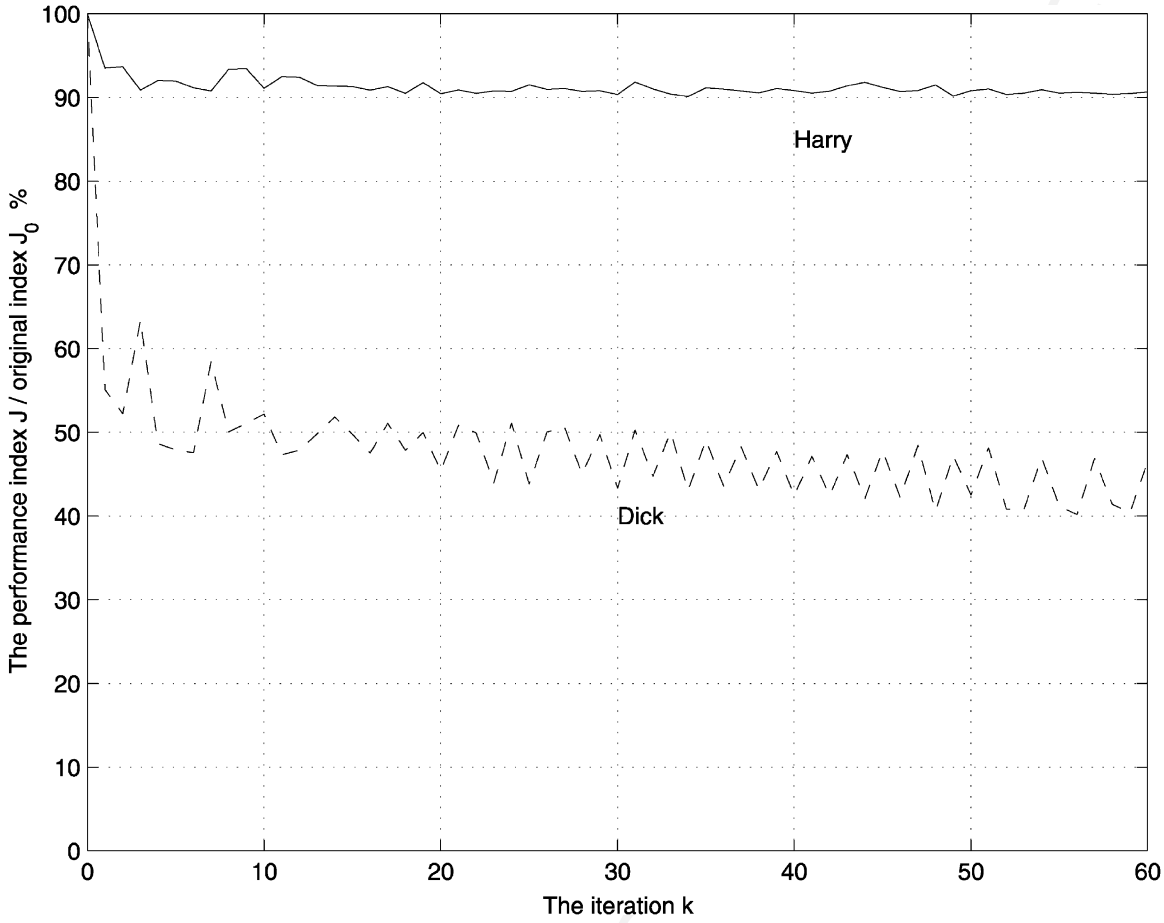

Fig. 7. Performance improvement in stochastic optimization algorithm.

median as a function of the obstacle detection distance τ for Dick's initial model (solid line) and Dick's optimized model (dashed line), where $v_{\text{initial}} = 35$.

From Fig. 8, we can calculate J_1^0 and J_1^{60}:

$$J_1^0 \approx \frac{42}{35} = 1.20, \tag{72}$$

$$J_1^{60} \approx \frac{36}{35} = 1.03. \tag{73}$$

Thus, Dick's optimized HCS model not only improves tight turning performance, but obstacle-avoidance performance as well. This should not be too surprising, since the tight-turning and obstacle-avoidance behaviors are in fact tightly related. During the obstacle avoidance maneuver, tight turns are precisely what is required for successful execution of the maneuver.

Second, we would like to see how much performance optimization changes the model's control strategy away from the original human control approach. To do this we turn to a hidden Markov model-based similarity measure [9] developed for comparing human-based control strategies. Let H_x denote the human control trajectory for individual x, let M_x denote control trajectories for the unoptimized model corresponding to individual x, and let O_x denote control trajectories for the optimized model (with respect to J_2) corresponding to individual x. Also let $0 \leq \sigma(A, B) \leq 1$ denote the similarity measure for two different control trajectories A and B, where larger values indicate greater similarity, while smaller values indicates greater dissimilarity between A and B.

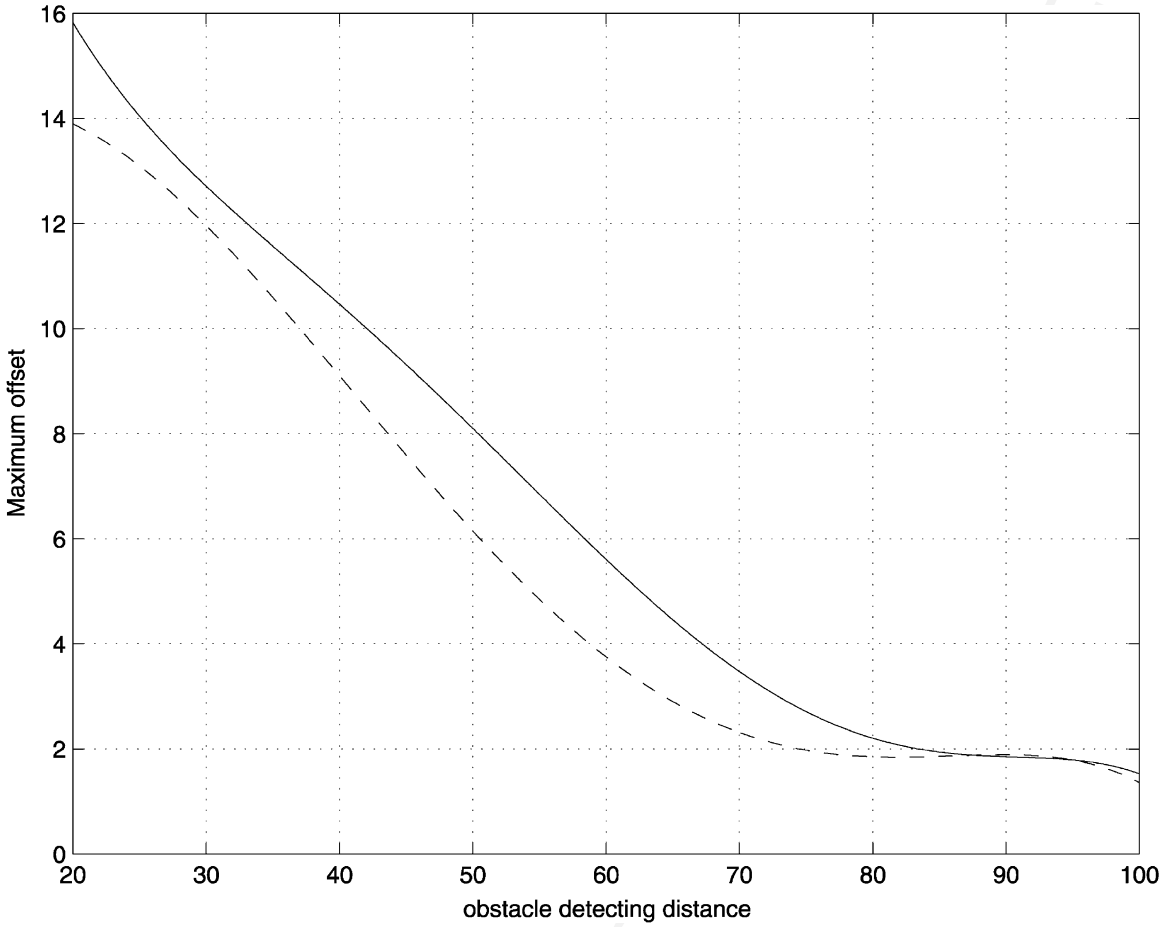

Fig. 8. Maximum lateral offset for original (solid) and final (dashed) HCS models.

For each individual, we can calculate the following three similarity measures:

$$\sigma(H_x, M_x), \tag{74}$$

$$\sigma(H_x, O_x), \tag{75}$$

$$\sigma(M_x, O_x). \tag{76}$$

Table 2 lists these similarities for Dick and Harry. From our experience with this similarity measure, we note that all the values in Table 2 indicate significant similarity. Specifically, the similarities for $\sigma(H_x, O_x)$ (0.434 and 0.469) suggest that even after performance optimization, a substantial part of the original HCS is preserved. Furthermore, the other similarity measures are consistent with the degree of performance improvement in each case. For Dick, where a substantial performance improvement of 55% was achieved, the similarity between the initial and optimized models is far less than Harry, where the performance improvement was more incremental.

We conclude with one final observation. Pomerleau's work on vision-guided autonomous driving

Table 2
Control strategy similarity

	x = Dick	x = Harry
$\sigma(H_x, M_x)$	0.762	0.573
$\sigma(H_x, O_x)$	0.434	0.469
$\sigma(M_x, O_x)$	0.544	0.823

[11], while impressive and ground-breaking, does not directly address the issues we have investigated here. Pomerleau learned to map the view of the road ahead to an appropriate steering direction, first through a neural network [11] and later with a statistical algorithm known as RALPH [12]. He does not model or analyze the dynamics inherent in human control strategies; rather, he very successfully solves the computer-vision problem of correctly estimating the position of the road in a video stream of data. Therefore, we view our work as complementary to Pomerleau's work, in that both research aspects are desirable in an eventual autonomous driving system.

8. Conclusion

Modeling HCS analytically is difficult at best. Therefore, an increasing number of researchers have resorted to empirical modeling of HCS as a viable alternative. This in turn requires that performance criteria be developed, since few if any theoretical guarantees exist for these models. In this paper, we develop several such criteria for the task of human driving, including criteria based on event analysis and criteria based on inherent analysis. We model human driving using the cascade neural network architecture, and evaluate the performance of driving models derived from different individuals using the developed performance criteria. Based on the criteria, we have proposed an iterative optimization algorithm for improving the performance of learned models of HCS. The algorithm keeps the overall structure of the learned models in tact, but tunes the parameters (i.e. weights) in the model to achieve better performance. It requires no analytic formulation of performance, only two experimental measurements of a defined performance criterion per iteration. We have demonstrated the viability of the approach for the task of human driving, where we model the HCS through cascade neural networks. While performance improvements vary between HCS models, the optimization algorithm always settles to stable, improved performance after only a few iterations. Furthermore, the optimized models retain important characteristics of the original HCS.

Acknowledgements

This work is supported in part by Hong Kong Research Council Grant No. CUHK519/95E, CUHK-4138/97E and Carnegie Mellon University.

References

[1] H. Asada, S. Liu, Transfer of human skills to neural net robot controllers, Proceedings of the IEEE International Conference on Robotics and Automation 3 (1991) 2442–2447.
[2] H. Friedrich, M. Kaiser, R. Dillman, What can robots learn from humans? Annual Reviews in Control 20 (1996) 167–172.
[3] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[4] M. Kaiser, Transfer of elementary skills via human–robot interaction, Adaptive Behavior 5 (4) (1997) 249–280.
[5] H. Hatwal, E.C. Mikulcik, Some inverse solutions to an automobile path-tracking problem with input control of steering and brakes, Vehicle System Dynamics 15 (1986) 61–71.
[6] R.A. Hess, Human-in-the-loop control, in: W.S. Levine (Ed.), The Control Handbook, CRC Press, Boca Raton, FL, 1996, pp. 1497–1505.
[7] U. Kramer, On the application of fuzzy sets to the analysis of the system driver-vehicle-environment, Automatica 21 (1) (1985) 101–107.
[8] M.C. Nechyba, Y. Xu, Human control strategy: abstraction, verification and replication, IEEE Control Systems Magazine 17 (5) (1997) 48–61.
[9] M.C. Nechyba, Y. Xu, Stochastic similarity for validating human control strategy models, Proceedings of the IEEE Conference on Robotics and Automation 1 (1997) 278–283.
[10] M.C. Nechyba, Y. Xu, Cascade neural networks with node-decoupled extended Kalman filtering, Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation 1 (1997) 214–219.
[11] D.A. Pomerleau, Neural Network Perception for Mobile Robot Guidance, Kluwer Academic Publishers, Boston, 1993.
[12] D.A. Pomerleau, T. Jochem, Rapidly adapting machine vision for automated vehicle steering, IEEE Expert 11 (2) (1996) 19–27.
[13] W.H. Press, et al., Numerical Recipes in C, 2nd Edition, Cambridge University Press, Cambridge, 1992.
[14] J. Song, Y. Xu, M.C. Nechyba, Y. Yam, Two measures for evaluating human control strategy, Proceedings of the IEEE Conference on Robotics and Automation 3 (1998) 2250–2255.
[15] J.C. Spall, Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, IEEE Transactions on Automation Control 37 (3) (1992) 332–341.
[16] M. Sugeno, T. Yasukawa, A fuzzy-logic-based approach to qualitative modeling, IEEE Transactions on Fuzzy Systems 1 (1) (1993).

Yangsheng Xu received his Ph.D. from the University of Pennsylvania in Robotics. He is currently Professor and Chairman of the Department of Automation and Computer-Aided Engineering at the Chinese University of Hong Kong (CUHK). Before joining CUHK, he was a faculty member at the Robotics Institute, Carnegie Mellon University, where he was working on space robotics, zero-gravity systems, robot design and control, and human skill learning. Recently, he has been working on microsystems, wearable robots and dynamically stable systems. He has published over 180 papers and has served on the editorial board of the IEEE Transactions on Robotics and Automation and the Journal of Automation. He has also served on various advisory boards to industry and government organizations in Hong Kong, US, Japan, Korea, mainland China and the United Nations. He is a Fellow of HKIE.

Jingyan Song is an Associate Professor in the Department of Automation at Tsinghua University, Beijing. He received his Ph.D. in systems engineering and engineering management at the Chinese University of Hong Kong (CUHK) in 1999. His current research interests include machine learning, robotics, intelligent control, neural networks and intelligent transportation systems.

Michael C. Nechyba received his B.S. in electrical engineering at the University of Florida in 1992, and his Ph.D. in robotics at Carnegie Mellon University in 1998. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Florida, where he serves as associate director of the Machine Intelligence Laboratory. His current research interests include real-time computer vision, machine learning, human-centered robotics, neural networks and hidden Markov models. Current research projects include vision-based activity tracking and vision-based flight autonomy for micro-air vehicles.

Yeung Yam received his B.S. and M.S. degrees in physics from the Chinese University of Hong Kong (CUHK) and the University of Akron, respectively, in 1975 and 1977, and his M.S. and Sc.D. degrees in aeronautics and astronautics from the Massachusetts Institute of Technology, Cambridge, in 1979 and 1983, respectively. He is currently Associate Professor in the Department of Automation and Computer-Aided Engineering at the CUHK. Before joining the university in 1992, he was with the Control Analysis Research Group of the Guidance and Control Section at Jet Propulsion Laboratory, Pasadena, CA. His research interests include dynamics modeling, control analysis and design, and system identification.