Learning Haptic Feedback for Guiding Driver Behavior*

Michael A. Goodrich and Morgan Quigley, mike@cs.byu.edu
Computer Science Department
Brigham Young University
Provo, UT, USA

Abstract – *Information about the driving state can be conveyed to automobile drivers through force feedback signals sent via the pedals and steering wheel. Because the set of possible haptic signals and driver responses is huge, it is desirable to automatically learn which signals are most useful to drivers. Thus, it is instructive to explore how machine learning techniques can be used as a step in the design of a haptic interface system. In this paper, we present a learning algorithm that learns useful haptic feedback and apply the algorithm to learning feedback for automobile drivers. We present evidence to show that the algorithm is sensitive enough to learn useful feedback under some circumstances, but that its scope may be limited by people's ability to act as admittance controllers.*

Keywords: Haptic interfaces, automobile driving, driver support systems.

1 Introduction

In this paper, we present a machine learning approach to discovering haptic signals that inform automobile drivers about driving risks. Because of the enormous complexity of human perception and motor control, it is important to explore how to create a good mapping from haptic stimuli to driver response selection without constructing a precise model of each possible perceptual controller used by a driver.

The two fundamental activities and obligations in highway driving are lateral (steering) and longitudinal (speed) control. It is desirable to design haptic feedback systems that support drivers in fulfilling these obligations. Such active feedback systems require people to expand the set of roles that they naturally assume as they interact with a vehicle. These interaction roles can be described in various ways.

- Position control (regulation). In this role, people suppress disturbances in forces that naturally occur by keeping the pedal or wheel in the desired position. For example, when drivers experience the vibration caused by an antilock brake system, they may choose to keep the pedal in the same position by damping the vibrations and thereby controlling the pedal's position.
- Position control (tracking). In this role, people move the pedal or wheel to a desired position despite forces in the environment. For example, the gas pedal has a spring that returns the pedal to the neutral position. To speed up, people overcome this force to move the pedal to a desired position. Stiffness forces imposed by the pedal or wheel can be used to inform the estimate of the position.
- Admittance control. In this role, people govern how pedal or wheel forces affect position. For example, mechanical forces on the front wheels of a vehicle induce a "return-to-center" force when the steering wheel is deflected from the straight-ahead position. When completing a turn, drivers may admit this force to straighten the steering wheel in a controlled way.
- Force control. In this role, people do not care about the position of the pedal or wheel, but just care about generating maximal movement. For example, in an emergency, people sometimes generate dramatic forces to slam on brakes or swerve sharply.

Both the gas pedal and steering wheel are primarily position devices, meaning that people control acceleration and velocity by changing the positions of these of these devices.

Two key questions arise in using haptic signals to support drivers: what type of haptic signal should be used (i.e., admittance, position, force), and how should this signal be shaped. We speculate that adding forces sometimes requires people to expand the range of conditions under which they will adopt an admittance control strategy. Such adaptation complicates the identification of useful systems and can increase design times. We propose that machine learning can be used to interactively learn acceptable haptic cues.

2 Using Machine Learning

Machine learning is a broad field that tries to develop algorithms which adapt their behavior to some specifi-

*0-7803-8566-7/04/$20.00 2004 IEEE

cation. Reinforcement learning is a subfield of machine learning that tries to generate control laws/policies that lead to desired states. The idea of a reinforcement learning algorithm is to identify a sequence of actions that lead to a desired reward.

In this sense, both reinforcement learning and optimal control theory have much in common; both try to find a sequence of actions that maximize expected reward. Optimal control theory requires a detailed model of the system to be controlled and then uses the calculus of variations to find a control law that maximizes a performance metric subject to the model. By contrast, some reinforcement learning algorithms do not require a model of the system to be controlled, but rather learn through experience the sequence of behaviors that maximizes expected reward. The algorithm that we will use is based on a variation of Q-learning, a popular reinforcement learning algorithm [4].

2.1 Q-Learning: A Brief Review

The Q-learning algorithm assumes the decision process consists of states, actions, rewards, and utilities. Formally, let Θ denote the set of possible states, $\mathcal{A}$ denote the set of possible actions available to an agent, and $R(\theta, a)$ denote a reward that the agent receives when action $a \in \mathcal{A}$ is taken from state $\theta \in \Theta$. Q-learning is a technique for learning a mapping $\pi : \Theta \rightarrow \mathcal{A}$ that maximizes the expected discounted reward experienced by the agent. This mapping is called a *policy*, and the objective is to find an optimal policy.

Formally, the Q-value for a state and action, denoted $Q(\theta, a)$, is defined as the expected discounted reward that occurs from state θ if the agent chooses action a at time t, and then chooses optimally thereafter. This means that we can write the definition of the Q-value as[4]

$$Q(\theta, a) = R(\theta, a) + \gamma \sum_{\theta'} P(\theta'|\theta, a) \max_{a' \in A} Q(s', a'),$$

where P is a probability describing transitions from one state, θ, to another state, $\theta\prime$, given a particular action, a; and $\gamma \in (0, 1)$ is a discount factor.

This equation is the basis for the Q-learning algorithm. Before stating the algorithm, note that because the world is stochastic, choosing action a in state θ does not always lead to a reward. We use $R(\theta, a)$ to denote the average value of the reward that occurs for choosing a from θ, and $r(\theta, a)$ as the reward that occurs on a particular trial. The Q-learning algorithm keeps a guess of the true Q-values, and updates this guess through experience according to the following equation:

$$\hat{Q}(\theta, a) \leftarrow (1 - \alpha_i)\hat{Q}(\theta, a) + \alpha_i \big[r(\theta, a) + \gamma \max_{a' \in A} \hat{Q}(\theta', a')\big],$$

where $\hat{Q}$ denotes the estimate of the Q-value, $\alpha_i \in (0, 1)$ denotes the learning rate at time i, and θ' is the state that actually occurs when action a is chosen from state θ. In words, this equation creates a new estimate of the Q-value that is a convex blend of the old estimate with the outcome of an experiment; this experiment produces a reward $r(\theta, a)$ and a new state θ', so this reward and the Q-value of the new state are used to update the estimate.

As the agent experiences the world, information about which actions lead to a reward or penalty slowly propagate to other Q-values. Provided that (a) the agent visits every state infinitely often, (b) the agent tries every action from every state infinitely often, and (c) the learning rate α_i decreases fast (but not too fast) [4], then it can be shown that $\hat{Q}(\theta, a) \rightarrow Q(\theta, a)$ for all states and actions. Condition (c) simply means that over time, new experiment outcomes are gradually suppressed until eventually just the average value is considered.

2.2 Dichotomous Attributes

In practice, Q-learning works well without satisfying the technical convergence conditions because a correct policy can be learned even if the estimate of the Q-values is not perfect. We can exploit this practical side of Q-learning to create an algorithm that learns a satisficing [6] policy very quickly in worlds that have dichotomous reward attributes. In practice, reward functions frequently have two dichotomous attributes: one attribute which encourages goal seeking, and one attribute which discourages risky actions. For some worlds, it is easier to discover which actions lead to collisions than it is to discover which actions lead to the goal. Why? Because it is often easier to find a path that hits a wall than a path that reaches the goal. In this case, penalties influence the Q-values faster than the rewards because information from nearby obstacles propagates back faster than information from far-away goals. The satisficing Q-learning algorithm exploits this frequently encountered characteristic to dramatically speed learning in both penalty-rich and reward-rich worlds.

2.3 Satisficing Q-Learning

Rather than keep a single Q-value that represents both goal-achieving and risk-avoiding values, satisficing Q-learning keeps two values: $G(\theta, a)$ and $L(\theta, a)$. The first function represents the goal-achieving rewards, and the second function represents the risk of incurring losses. We update these functions separately via the following variants of the Q-learning equation:

$$G(\theta, a) \leftarrow (1-\alpha)G(\theta, a) + \alpha(\max(0, r(\theta, a)) + \gamma G(\theta', a'))$$

$$L(\theta, a) \leftarrow (1-\alpha)L(\theta, a) + \alpha(\max(0, -r(\theta, a)) + \gamma L(\theta', a')),$$

where a' is an action chosen at the next decision point. These equations simply assign all reward information (e.g., $r(\theta, a) > 0$) to the G-function, and all penalty information (e.g., $r(\theta, a) < 0$) to the L-function. If the

1. Initialize $G = L = 1$ for all a and s.
2. Repeat the following until force is stable.
3. Calculate μ_A and μ_R from G and L.
4. Select $a \in S(\theta)$ in two ways:
   (a) with a uniform probability for a while
   (b) according to some selection rule for a while
5. Execute a for at least 1/4 second until θ changes ($\geq$ 2/3 sec during training to reduce consequence overlap).
6. Update G and L, tremble L, and decay tremble.

Figure 1: The application of the satisficing learning algorithm to learning force feedback.

L-function can quickly learn which actions lead to problems, then we should be able to exploit this to avoid risky choices while learning to choose actions that produce rewards.

Avoiding risky choices can be accomplished by using a satisficing decision rule: actions which are more likely to lead to a goal than to expolse the agent to risk are "good enough." Since L and G are determined by different rewards and depend on different discount values, it is necessary to make them comparable. We do this by normalizing the utilities as follows: $\mu_A(\theta, a) = \frac{G(\theta,a)}{\sum_a G(\theta,a)}$ and $\mu_R(\theta, a) = \frac{L(\theta,a)}{\sum_a L(\theta,a)}$. The subscript A denotes reasons to **A**ccept an action, and the subscript R denotes reasons to **R**eject an action. An action is satisficing if the reasons to accept it outweigh the reasons to reject it. This decision rule can be characterized by listing the set of all actions that are satisficing, $S(\theta) = \{a : \mu_A(\theta, a) \geq \mu_R(\theta, a)\}$.

3 Satisficing Q-Learning in the Driving Context

This section describes parameter settings and results from applying machine learning to the design of driver support systems. The goal is to learn how to support force, position, and admittance control. A key element of accomplishing this goal is to "first do no harm." As such, we invoke the following:

The burden of proof principle. *There should be a compelling reason to give information to the human or to take action. If the driver is using a behavior that is satisficing, do not intervene or inform.* The formal statement of the algorithm that attempts to do this is shown in Figure 1. We now present the specifics of this algorithm for both steering and speed control.

3.1 Steering

States. The state representation must be expressive enough to represent deviation from ideal driving behavior. This was implemented by creating a simple lookahead PD controller that mapped vehicle orientation and velocity into steering wheel angle. This PD controller was tested in the simulator and produced reliable steering behavior.

Part of the state was then defined as the difference between the observed steering wheel position and the position specified by the PD controller. This latter position is referred to as the *ideal* position, even though the controller was imperfect[1]. This difference was then discretized into five sets, corresponding to negligible deviations from ideal, minor deviations left and right of the ideal, and major deviations left and right of the ideal. We refer to this state dimension as *error*.

The urgency of the situation was also included in state. Urgency was represented by time to lane crossing (TLC) and discretized into five sets, corresponding to negligible urgency, minor urgency to the left or right, and major urgency to the left or right. TLC was measured using the distance between the side of the vehicle and the lane boundary, denoted by δ, as follows $\text{TLC} = (\delta)/(\frac{d\delta}{dt})$.

Actions. The philosophy of selecting actions is to generate forces that inform drivers of the correct action to take. Five forces were considered, corresponding to a large force to the left or right, a small force to the left or right, and no extra force. This approach uses forces to nudge the steering wheel in directions that would suggest a correction in steering wheel position. When an action was selected, it was applied for at least 0.25 seconds and lasted until the state changed.

We submit that much of steering is position-based implemented by commands such as "Move the wheel position to a desired angle," albeit with admittance-based exceptions such as "Allow the return-to-center force of the wheel to move the wheel until the position reaches a desired angle." It is desirable to exploit such admittance-based exceptions to communicate the need to change behaviors by indicating an incorrect position through haptic signals. To do this, an error in wheel position can be indicated by generating a force in the direction of the correct position. This can be done by increasing or decreasing wheel stiffness, changing the "zero-point" of the wheel to communicate a missed position, or generating a "nudge" force that lasts for a brief interval of time and that pushes the wheel in the correct position. It is interesting to note that wheel forces generated by any of the above means produce the same initial behavior on the wheel; the differences in these forces can only be perceived as the wheel starts moving. This suggests that if forces are applied for only brief duration then any of the above approaches can be used.

Because of its compatibility with the learning al-

[1]Since Q-learning works in stochastic domains and since errors in the PD controller were approximately zero mean, the non-ideal nature of the controller was ignored.

gorithm and because of limitations of our force feedback steering wheel, we adopt the latter perspective. As mentioned previously, the set of actions was limited to five discrete actions. Thus, $c_{\text{alg}} \in \{c_{LL}, c_{SL}, c_Z, c_{SR}, \text{ and } c_{LR}\}$ corresponding to **L**arge nudges **L**eft and **R**ight, **S**mall nudges **L**eft and **R**ight, and **Z**ero nudge.

Rewards. The satisficing Q-learning algorithm assigns rewards and penalties for an action given a state. This requires us to specify what actions are rewarded and what actions are penalized. Intuitively, an action is good if it enhances comfort. We selected two first order estimates of comfort: workload and impedance. Workload was estimated using Boer's steering entropy metric [5] applied over a 2 second sliding window.

We used a heuristic estimate of impedance that minimized computations. Rather than explicitly estimating human impedance, we instead used the notion that impedance is a measure of human effort expended to oppose the forces of the wheel. This effort is approximately given by the difference between the observed wheel position and the position that the force would have generated if the human had not impeded it. When this difference is small it indicates that the human allowed the force to change the position of the wheel. This, in turn, indicates that the human is not expending effort to oppose the wheel, which suggests that the human is comfortable with the wheel's behavior. Heuristic thresholds were selected for levels of acceptable differences. Anecdotal evidence suggests that the behavior of the learned policy is not sensitive to the precise selection of the threshold.

Penalties. An action is bad if it exposes a driver to risk; in lateral control, a bad action tends to lead to a lane departure. Two factors determine this risk exposure: lane position, δ, and drift, $\frac{d\delta}{dt}$. Lane position is an objective assessment of *risk*, and drift is an assessment of the *urgency* of the situation. As illustrated in Figure 2, threshold lines were empirically chosen, and states and actions were penalized that lead the vehicle outside of these thresholds or did not significantly bring the vehicle closer to the safe region.

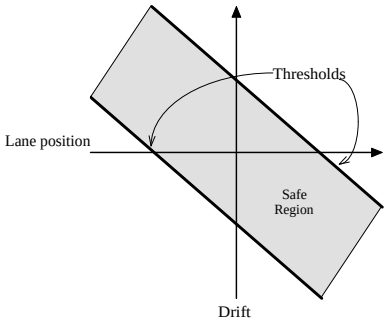

Figure 2: Thresholds for penalizing an action in lateral control.

3.2 Speed Control

The motivations of how rewards, penalties, and states are chosen under speed control are similar to those of steering. We therefore omit much of the discussion in the interest of space.

States. The state representation is obtained by noting that ideal driver following has infinite time to collision and a time headway on the order of 1.5 to 2 seconds [3]. In this context, time headway represents the exposure to risk and time to collision represents urgency.

Because ideal time to collision is infinite, it is very difficult to discretize this state space. In keeping with [3], therefore, we defined the state space as time headway and inverse time to contact. This state space was discretized into nine headway values and nine inverse time to contact values.

Actions. The philosophy of selecting actions is to generate forces that inform drivers of the correct action to take. Three forces were considered, corresponding to a large extra force toward the neutral position, a small extra force toward the neutral force, and no extra force beyond that provided by the passive pedal spring. When an action was selected, it was applied for at least 0.25 seconds and lasted until the state changed.

Rewards. Workload was estimated using Boer's steering entropy metric extended to the pedal and applied over a 2 second sliding window. The discretization of the prediction error distribution was empirically chosen to correspond to the subjective thresholds. Impedance was heuristically estimated as in the previous section, and subjective thresholds were set so that actions were rewarded if both workload and impedance were low.

Penalties. In longitudinal control, a bad action tends to lead to a collision. We penalized a state action pair if it caused the vehicle to enter into the least safe discretization region of the THW-ITTC state space.

4 Experiment Results

In this section, we present results from a study of lateral and longitudinal control. The driving simulator consisted of a CRT, a force feedback steering wheel, a force feedback pedal, and a computer running a simulated driving environment.

In the experiment, drivers are asked to perform a primary driving task while simultaneously performing a secondary math task. The primary task for the lateral control experiment is to guide the vehicle through a curvy course at high speeds. The primary task for the longitudinal control experiment is to maintain an acceptable following distance behind an erratic lead vehicle.

The math task is to perform a two-digit addition or subtraction problem, and then to determine if the answer is greater than or less than a supplied target value.

Subjects press buttons on the steering wheel to indicate whether the value is greater than or less than the target value. This secondary task experiment is a variant of one that we have used extensively in work on human-robot interaction [1]. Rapidly performing two-digit arithmetic places a high cognitive load on most subjects.

The experiment consisted of two phases: training the force feedback algorithm and validating the resulting forces using human subjects. To expose the learning algorithm to a wide variety of driving conditions during training, it is necessary to passively drive the vehicle as an admittance controller. Such driving allows the algorithm to experience consequences of actions under states that lead to lane departures, collisions, and near collisions. Currently, training was done only by a single operator but consistent results were obtained over several training episodes; future work should extend this to include training with multiple operators.

4.1 Lateral Control Results

Using the techniues described in previous sections, the algorithm required approximately twenty minutes of training to find a useful policy where the agent would have enough experience to apply corrective forces to the steering wheel only when a truly dangerous situation occurred.

Subjects were required to perform three sessions of simulator driving while simultaneously performing simple arithmetic comparisons. Each session lasted for approximately ten minutes and used a different control policy for the force-feedback steering wheel. The policies, which were presented to the subjects in random order, were as follows:

1. Nominal steering wheel. The wheel applied a conventional stiffness-based return-to-center force when deflected from the center position.
2. "Non-dominating" reactive control policy. The satisficing action that produced the largest difference between reward and penalty was applied on the steering wheel.
3. Non-intrusive reactive control policy. This policy selected the satisficing action that produced the smallest force possible. This was done by choosing the action with maximum reward from the set of satisficing actions.

12 subjects participated in the experiment. All subjects were college aged students with valid driver's licenses. After having experienced all three control policies, subjects were asked to name their favorite. Seven of the twelve test subjects preferred to have corrective forces superimposed on the wheel, while the others preferred the standard return-to-center behavior. The enthusiasm of some users was counterbalanced by the frustration of other users, both in subjective evaluations and objective performance metrics. As a result, the results of this experiment did not indicate a clear benefit for haptic feedback. However, the nearly even split between subjects who liked the feedback and those who were frustrated with the feedback indicates that the learning algorithm produced a policy that matched nominal performance.

Secondary task performance, measured as the average percentage of arithmetic problems answered correctly by the entire pool of subjects, was almost identical for the control policies. The order in which the policies were tested seemed to be more related to task performance than the policies themselves, and by choosing the policies in random order, the performance differences almost completely averaged out.

Interestingly, some drivers significantly improved their driving with the system enabled, indicating that, for some users, the system was helpful in maintaining vehicle position and reducing lane position drift. Other drivers fought against the system and performed poorly. This suggests that some drivers are more amenable to shifting between position-based and admittance-based control than others.

4.2 Longitudinal Control

Training the algorithm consisted of starting with a uniform Q-table and passively driving (admitting forces) the simulator for approximately fifteen minutes. The agent quickly learned to avoid impeding the trainer's pedal actions by collecting high rewards when it did not increase the pedal resistance, and eventually learned to actuate the pedal to avoid collisions.

Subjects were required to follow a lead car whose behavior was quite erratic and unpredictable, simulating the difficulties of rush-hour traffic on a freeway. The nominal cruise speed of the lead car was adapted to the speed of the subject-driven car in such a way that the THW between the lead car and the subject-driven car was almost constantly between 0 and 4 seconds. This adaptation occurred with lag so that the linking between subject behavior and lead car did not confound the data. If a subject dropped too far behind the lead car, "Speed up!" was printed in large letters across the screen. As a result, subjects were motivated to operate in the difficult region between an overly safe following distance and a dangerously close "tailgating" situation. Subjects were also asked to perform the same style of comparative arithmetic problems as in the lateral control experiment.

The agent learned to increase pedal resistance in boundary states that border regions of certain disaster and regions of no risk. Interestingly, the agent did not learn to activate the pedal motor when disaster was imminent, resulting in a control policy where the pedal resistance was only given on the threshold between "no risk" and "certain disaster." This critical dividing line represents the "point of no return" beyond

which the benefits of haptic information diminish. The exact placement of the dividing line is naturally dependent on the driving style demonstrated to the agent by the human trainer. The pedal was trained multiple times and consistently converged to similar policies, even under a variety of discretization schemes.

Subjects were asked to drive two 10-minute segments, once with the active haptic forces and once with only the passive spring resistance of the pedal. The order of the control policies was randomized for each user. Nine of the twelve formal preferred the pedal forces. A representative comment was, "I noticed that with the second pedal I didn't feel as pressured, and it seemed like I didn't have to work as hard to keep tract [sic] of both the math problems and the driving. I felt more on 'auto-pilot.' "

Those subjects who did not prefer the pedal forces seemed to miss the point of the haptic channel: "It was annoying, I didn't understand it," one subject said. Like the lateral control trials, such reactions possibly represent a direct conflict between the driving style of the trainer and the test subject, and not necessarily a failure of the algorithm or interface.

The ability of the pedal algorithm to help the driver avoid crashes was well supported by the data. When a time headway value of 0.7 seconds was set as an "imminent danger" threshold, drivers spent 45% less time in the "imminent danger" zone with the haptic signal on the pedal versus without the signal. This large difference is the major advantage provided by the pedal forces.

The average NASA TLX score only decreased from 70.65 to 70.47 indicating that the system increased safety without altering comfort; both average headway went up and minimum headway went up, but subjective workload estimates remained the same.

Since the number of subjects was small, it is difficult to determine whether the differences were caused by a preference for the haptic system of by random deviations. Fortunately, there is additional qualitative evidence to suggest that the system learned something useful. This evidence is obtained by viewing the regions of the THW/TTC state spaces where forces were applied and where forces were not applied. We compared these regions to data from a study of where drivers initiate braking in response to a "cut-in" [2]. The learned force regions correspond to locations where braking occurs with high probability, and the learned no-force regions correspond to locations where drivers rely on engine braking with high probability. This suggests that the learned profile supported humans in their ideal system behavior.

5 Discussion

We presented the satisficing Q-learning algorithm and showed how it could be applied to learning forces for both lateral and longitudinal control. We analyzed data from two secondary task studies: one for lateral control and one for longitudinal control. We showed that the learned forces for lateral control could match but not exceed unsupported performance. Two technical obstacles prevent better results. First, the number of actions and number of states that we used is too small. Second, and probably more importantly, our simulator does not allow damping effects and mass effects to be included in the learned force profile, and these effects appear necessary to support lateral control.

By contrast, the learned forces for longitudinal control show good improvement. People followed at a higher headway, and they had far fewer near collisions. Furthermore, this effect was obtained without increasing people's perception of workload. This means that the learned forces enhanced safety without reducing comfort. We believe that these results could be improved by adding more states, more actions, and damping/mass effects.

Acknowledgements

The authors would like to thank Nissan Motor Company for funding that supported this work, and for providing the force feedback gas pedal.

References

[1] J. W. Crandall and M. A. Goodrich. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *Proceedings of the 2002 IEEE /RSJ International Conference on Intelligent Robots and Systems*, Lusanne, Switzerland, 2002.

[2] M. A. Goodrich and E. R. Boer. Model-based human-centered task automation: A case study in acc design. *IEEE Transactions on Systems, Man, and Cybernetics — Part A: Systems and Humans*, 33(3):325–336, May 2003.

[3] M. A. Goodrich, E. R. Boer, and H. Inoue. A model of human brake initiation behavior with implications for ACC design. In *IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems*, pages 86–91, Tokyo, Japan, October 5-8 1999.

[4] T. M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[5] O. Nakayama, T. Futami, T. Nakamura, and E. R. Boer. Development of a steering entropy method for evaluating driver workload. In *SAE Technical Paper Series: #1999-01-0892: Presented at the International Congress and Exposition*, Detroit, Michigan, March 1-4, 1999.

[6] H. A. Simon. *The Sciences of the Artificial*. MIT Press, 3rd edition, 1996.