

# Behavioral Entropy in Human-Robot Interaction

M. A. Goodrich<sup>1</sup>, E. R. Boer<sup>2</sup>, J. W. Crandall<sup>1</sup>, R. W. Ricks<sup>1</sup>, and M. L. Quigley<sup>1</sup>

<sup>1</sup> Computer Science Department  
Brigham Young University  
Provo, UT, USA

<sup>2</sup> Erwin R. Boer Consulting  
San Diego, CA, USA

## ABSTRACT

The ability to quickly and accurately measure how various design decisions affect human workload is an important need in human-robot interaction (HRI) and other HMI domains. Although various techniques allow workload to be estimated, it is important to develop methods for obtaining workload estimates objectively and in real-time without interfering with the normal operation of human. In this paper, we develop *behavioral entropy* as a technique for estimating human workload in HRI domains. We develop relevant theory and present case studies that help validate the power of behavioral entropy.

## 1 Introduction

In a recent article on useful metrics in human-robot interaction (HRI), Fong et al. identified the need to find “nonintrusive measures of workload that can characterize operator stress in real-time” [3]. The importance of having a real-time estimate requires an objective (rather than subjective) measure of workload that is reliable and applicable to many interfaces. The purpose of this paper is to present a technique, called *behavioral entropy* [6] that measures human workload in HRI domains. This metric efficiently utilizes operator activity to estimate human workload.

A real-time measure of workload in HRI has several possible applications.

- **Design of adjustable autonomy systems.** Intelligent interfaces could be used to identify high workload situations, and the resulting information could be used to adjust robot autonomy or alert other humans to support the operator. This facilitates design of more efficient mixed-initiative systems [1] that follow principles of situation-adaptive autonomy [4].
- **Comparison of interfaces and autonomy modes.** Various HRI systems, including various interfaces and robot autonomy modes, could be compared over time. This ability to compare designs

over time allows not only comparison of average workload, but also comparisons of peak workload, minimum workload, and workload patterns.

- **Diagnosis of causes of high workload.** External events that trigger high workload could be identified and diagnosed. By associating a real-time estimate of workload with external events, those events that cause workload spikes could be identified. These events might include environmental contingencies, robot failures, interface issues, and so on.
- **Design of Adaptive Systems.** Interfaces or robots that learn to support human activity could be improved. Most HRI learning systems either learn by direct teaching or learn by observing a human teleoperating a robot. These systems could be augmented to include implicit human cues, such as identifying robot behaviors that cause workload spikes, and thereby improve interaction efficiency through interface adaption.

The idea of behavioral entropy was developed in for use in estimating driver workload in an automobile driving context. This first application restricted attention to human activity as recorded in the steering wheel of a vehicle and was called “steering entropy.” Subsequently, Boer generalized this concept to general human activity, and denoted the concept as behavioral entropy [6].

Behavioral entropy differs in a number of ways from the three other primary methods for evaluating workload: physiological measurements, secondary task studies, and *post hoc* workload measurements (such as NASATLX). Physiological measurements exploit the strong correlation between human effort and the body’s physical response. Such measures are objective and near to real-time, but much work needs to be done to understand the precise nature of the correlation between effort and response; this work includes developing signal processing techniques that rapidly and correctly separate signal from noise. Secondary task studies allow diagnosis of human workload by measuring how performance declines as other work is added. However, such measures are invasive and change the way the primary task

is performed. *Post hoc* measurements exploit a human’s ability to express their perceived workload after the fact. Such measures are important because they allow a human to be able to state how they perceived their experiences, but they are subject to many psychological biases, such as recency effects. Moreover, they are not real-time.

Behavioral entropy exploits patterns observed in human activity within an HRI context. Generally speaking, when intelligent operators perform a practiced skill under conditions of good information, they use an anticipatory control strategy. This means that they are able to predict the consequences of their actions or inactions, and select efficient behaviors that alter these consequences. When human operators are under conditions of high workload or other form of degraded performance, they anticipate less and react more and, as a result, their action selection tends to be more exaggerated. Anticipatory behaviors tend to be more smooth with less dramatic magnitudes and less frequent changes than reactive behaviors. Behavioral entropy is sensitive to this difference between reactive and anticipatory behaviors.

This paper is organized as follows. We first review and develop the key concepts associated with behavioral entropy. We then present three case studies that utilize behavioral entropy in HRI-related domains. The first two case studies help establish the hypothesis that average behavioral entropy is a useful and objective metric for comparing design decisions. The third case study helps illustrate that behavioral entropy can be used in real-time. We conclude by presenting future work with an emphasis on work needed to allow behavioral entropy to be used in broad-reaching HRI studies.

## 2 Behavioral Entropy

Behavioral entropy estimates workload by first observing patterns of human activity under normal conditions, and then noting deviations from these patterns. Consider, for example, how a human might teleoperate a robot via a joystick under laboratory conditions (good communications, alert operator, etc.). Under these ideal conditions, joystick activity follows observable patterns.

Such patterns of activity can be captured in a model of activity. A well-known phenomena associated with modelling is that simple models often explain most activity, but extending these models to explain all activity often makes the models grow exponentially in their complexity. This is true in human-robot interaction domains as well. For example, much of what is done with the joystick under teleoperation can be described with simple ARMA models [5], but modelling all joystick activity requires very sophisticated models.

Norbert Wiener once said, “It is my thesis that the physical functioning of the living individual and the operation of some of the newer communication machines are precisely parallel in their analogous attempts to control entropy through feedback” [8]. Through repeated interactions with robot or interface systems, humans build an understanding of various effects and relationships. Perhaps most importantly, they build an understanding of (a) the effect of their actions on the systems and (b) the dynamics of the environment.

Such an understanding translates into an efficient interaction. To paraphrase Wiener, people work to reduce entropy so skilled behavior minimizes entropy. This manifests itself in human behavior that is anticipatory, of the lowest possible bandwidth, and of the lowest possible magnitude. Such behavior lends itself to modelling and prediction.

### 2.1 Modelling

Suppose that we identify a simple model that describes how the operator uses the input device to a human-robot interface. (Such input devices can include a joystick, mouse, stylus, etc.) Formally, let  $x_t$  denote the state of the world at time  $t$  and let  $a_t$  denote operator activity at time  $t$ . A model  $M$ , denoted by,

$$M : X_t \times X_{t-1} \times \dots \times X_0 \times A_t \times A_{t-1} \times \dots \times A_0 \rightarrow A_{t+1}$$

can be used to predict operator activity at time  $t + 1$ ,

$$\hat{a}_{t+1} = M(x_t, x_{t-1}, \dots, x_0; a_t, a_{t-1}, \dots, a_0),$$

where the  $\hat{a}$  indicates a prediction. Given this model we can generate a prediction of what we think the operator will do next.

If we adopt Wiener’s hypothesis that people work to control entropy, then we can believe the hypothesis that people’s behavior patterns have lower magnitude, have lower bandwidth, and are anticipatory when good information is present and the task is well practiced. If so, then low frequency components of their observed activities represent the anticipatory aspects of their behavior. Consequently, we should be able to identify a model of this behavior.

There are several possible choices for these models. We could use general linear models, such as ARMA or state-space models, but in the interest of simplicity we restrict attention to only one type of model in this paper<sup>1</sup>: a Taylor series expansion. The Taylor series expansion supposes that behavior is a smooth function of past activities, and then uses the first derivatives to model the key elements of this function.

<sup>1</sup>Note that it might sometimes be better to use a sample and hold model to predict joystick movement because joystick operation, under some conditions, tends to be “bang-bang.” This is left as an area for future work.

## 2.2 Model Errors

Clearly, a model will not correctly predict all operator activity. Let  $e_t = \hat{a}_t - a_t$  denote the error in this prediction. The statistical properties of this error are useful in estimating operator workload. To illustrate this, suppose that the prediction error sequence,  $e_t$ , has been observed for  $0 \leq t \leq N$ . Given this sequence,  $\{e_0, e_1, \dots, e_N\}$ , we can create a histogram of prediction errors. By normalizing this histogram, we create a probability mass function that is a non-parametric estimate of the prediction error density function. Let  $p_E(e; t)$  denote this estimate of the prediction error density function.

The key idea behind using behavioral entropy is to look at the type of information that exists in the prediction error density function. More precisely, we will look at the information present in the prediction error density functions under two conditions. If one condition is produced under circumstances that allow better anticipatory control than the second condition, then operator activity under the first condition should be more predictable. In other words, there will be less information in the prediction error density function. Since good interfaces and autonomy modes support operators in their desire to minimize entropy, good designs should have more predictable behaviors. To better understand how to describe the information available in the prediction error density function, it is useful to review the relationship between creating a model and the notion of information.

## 2.3 Models and Information

One way to interpret a model of a phenomena is as a mechanism that gives you *information* about the phenomena. In this sense, we use the term “information” in the information theoretic sense as the number of bits required to describe the phenomena. If the model is very good, then deviations from the model predictions likely arise from randomness; if the model is poor, then deviations from the model predictions likely arise from structured aspects of the phenomena that are not captured in the model. For example, consider a phenomena where two variables are related to each other by a cosine function. If we create a linear model for this sinusoidal relationship, then deviations from the model predictions arise from the fact that the underlying phenomena is a sinusoid and not a line. If, by contrast, we create a sinusoid model for this relationship, then deviations from model predictions arise from random perturbations in the relationship.

We can use this relationship between model predictions and information to create a mechanism that identifies when activity is no longer ascribed to the phenomena encoded in the model. In other words, we can use the prediction error density function to detect when things

are different from what we predict and therefore detect when the phenomena is behaving oddly. Since predictions are subject to random error, we are actually going to use the prediction error density function to detect when things are *different enough* to conclude that the observed phenomena is not consistent with what was predicted.

Consider the amount of information available in the prediction error density. Under ideal conditions (e.g., laboratory setting, alert human, no interruptions) the prediction error density,  $p_E(e; t)$ , has a certain amount of information in it. This information is attributable to random noise and to small unmodelled aspects of the pattern of human activity. If conditions of high workload occur, then the pattern of human activity changes and so does the resulting prediction error density. By comparing the amount of information present in the prediction error density function under ideal conditions to the information present under loaded conditions, we can detect when these loaded conditions have occurred.

For example, consider the problem of teleoperating a robot via a joystick. We can create a simple model for how the joystick moves under ideal conditions and measure the information in the corresponding prediction error density. When the task suddenly becomes more difficult, operator activity tends to become more erratic and more pronounced. Instead of seeing small changes in the joystick position made relatively infrequently, large and rapid changes in joystick position are more frequently observed. If we were to compare the prediction error density under ideal conditions with the density under the loaded conditions, we would see that the density under loaded conditions is much more spread out; this is illustrated in Figure 1. This increased density spread in-

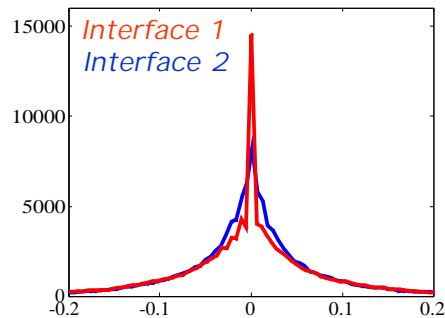


Figure 1: Prediction error histograms under two workload conditions: nominal and loaded.

icates that there is information in the system not captured by the model; it indicates that the operator is doing more than we predicted. Such things can occur, for example, when an operator overcompensates after hav-

ing attention diverted or when an operator is confused because information is presented poorly.

## 2.4 Model Information and Prediction Error Entropy

To create a metric that represents this change in prediction error density, we return to the information theoretic interpretation of the model. We use *entropy*,  $H(E;t)$ , defined as  $H(E;t) = -\sum_{e \in E} p_E(e;t) \log p_E(e;t)$ , as the measure of information available in the prediction error density. If we identify baseline entropy using ideal conditions then we can detect periods of high workload by comparing  $H(E;t)$  against the baseline entropy. Similarly, if we can identify entropy under one HRI system design, then we can compare this entropy with another design to help determine which design better supports the human.

We refer to  $H(E;t)$  as behavioral entropy, indicating that it is the amount of information present in a human’s behavior that was not captured by a model. Experiments in automobile driving indicate that this objective measure of entropy correlates well with subjective measures of workload [6].

## 2.5 Segue

In the remainder of this paper, we present three case studies that use behavioral entropy to perform various HRI-related tasks. In the case studies, we will first present the goal of the experiment, describe what the operator was asked to do, discuss characteristics of the environment and the interface, present the model used to predict operator activity, and present what we use as a baseline. The first two studies use average behavioral entropy and lend support to the thesis that behavioral entropy discriminates between good and bad operating conditions. The third case study uses a real-time version of behavioral entropy to learn proper force feedback; this case study uses a reinforcement learning technique to show that real-time estimates of behavioral entropy are informative.

## 3 Case Study 1: Comparing Usability of Two Teleoperation Schemes

In the first case study, behavioral entropy was used to compare two different robot autonomy modes to determine which autonomy mode was easier for humans to use. The hypothesis is that differences in behavioral entropy correlate well with other measures of performance

and are therefore useful in comparing different robot autonomy modes. We compute the prediction error density function using a prediction error sequence from the entire experiment, and compare the entropy of this density function with other performance measures under two robot autonomy modes.

Subjects were asked to drive a robot around the top floor of the Computer Science Department at Brigham Young University using two different autonomy modes: manual teleoperation and shared-control teleoperation [2]. In addition to driving the robot with their right hand (with a joystick), the users were asked to answer multiple choice (two-digit) addition and subtraction problems with their left hand. This experiment setup is illustrated in Figure 2. Subjects were told to guide the robot through the hallways as quickly as possible while answering as many math questions as possible. The video feed from the robot’s onboard camera was displayed on the same screen as the math problems. In this case study, entropy calculations were taken of joystick movements. Only the angle (not the magnitude) from the joystick input was used to calculate entropy.

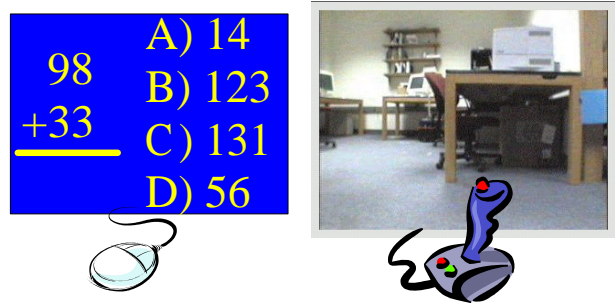


Figure 2: Interface used to compare the two autonomy modes.

## 3.1 Methods

A second-order Taylor series model of operator behavior was used. This means that the operator activity at time  $t$ ,  $a_t$ , was determined using observations of activity at times  $t - 3$  through  $t - 1$  (i.e., using  $a_{t-1}$ ,  $a_{t-2}$ , and  $a_{t-3}$ ). In this experiment, only joystick angle was used, and it is a reasonable assumption that if the operator is using angle  $a$  at times  $t - 3$  through  $t - 1$  then they will likely use this same angle at time  $t$ .

An important aspect computing entropy is selecting how to reliably create a discretized probability mass function from histogram data. In this experiment, a single operator guided the robot through the maze using the shared control autonomy mode without performing the secondary task. The history of joystick angles was

recorded, and the prediction error histogram was created. This histogram was discretized into 9 unequally spaced bins.

The bins were created using the following procedure. Using the resulting baseline prediction error density, we identify the parameter,  $\alpha$ , which encapsulates 90% of the data,  $Pr(-\alpha \leq \text{error} \leq \alpha) = 0.90$ . This value of  $\alpha$  is used to classify each angle, or the error from the predicted angle, into nine bins,

$$\{(-\infty, -5\alpha), [-5\alpha, -2.5\alpha], [-2.5\alpha, -\alpha], \dots \\ [-\alpha, -0.5\alpha], [-0.5\alpha, 0.5\alpha], (0.5\alpha, \infty)\}.$$

Since the bins were created from a single operator, this implementation of behavioral entropy is not sensitive enough to allow comparisons between individuals. Simply put, these values will be slightly different for each individual under ideal circumstances so the entropy computed from these values will differ under loaded conditions. As a result, entropy calculations should not be used to compare two individuals. However, since the same model and binning scheme were used under the two experimental conditions (with shared control and with direct control), it is possible to compare entropy for a given individual on the two different tasks.

### 3.2 Results

This experiment was performed in the real world and in a simulated world. Various results are shown for these case studies. Table 1 and 2 show the results from experiments in real and simulated worlds, respectively. In the tables, high values are good and low values are bad, with the exception of the entropy measurement which is reversed. In the tables, *Neglect* indicates the percentage of time that the operator spent doing arithmetic problems, *Performance* indicates how efficiently the primary task was completed as a percentage of the maximum possible performance, *# per min* indicate the number of arithmetic problems that were attempted per minute, and *% Correct* indicates what percentage of the attempted arithmetic problems were answered correctly by the subject.

For all measurements in both tables, subjects tended to do better using shared control than using direct control. Behavioral entropy is consistent with these other measurements since the highest entropy measure for shared control is lower than the lowest entropy measure for manual control. Also, entropy is highly correlated with performance (lower entropy corresponds to higher performance) and the amount of time the human “neglected” (i.e., did math problems) the robot (lower entropy means, generally, more neglect). There also appears to be correlation between secondary task proficiency and entropy.

#### Shared-Control Results

Participant	A	B	C	D	Ave.
% Neglect	51%	67%	46%	63%	57%
% Performance	77%	96%	81%	86%	85%
# per min.	9.5	18.9	8.9	10.6	12.0
% Correct	74%	98%	94%	66%	83%
Entropy	0.56	0.42	0.51	0.35	0.46

#### Direct-Control Results

Participant	A	B	C	D	Ave.
% Neglect	36%	31%	22%	62%	38%
% Performance	57%	76%	58%	60%	63%
# per min.	6.4	9.1	3.9	9.8	7.3
% Correct	72%	85%	79%	61%	74%
Entropy	0.72	0.79	0.67	0.63	0.70

Table 1: Results from the experiment in the real world.

The key to understanding how this data supports the use of entropy as a measure of workload lies in the dual task nature of the experiment. Adopting a limited resource model for cognitive information processing [7], we can assume that motivated subjects spend most of their cognitive effort either guiding the robot or solving math problems. This assumption is supported by the observation that the shared control autonomy mode was easier to use and freed subjects to spend more time solving math problems.

In the absence of a secondary task, it is reasonable to assume that performances using the two autonomy modes would have been closer. The presence of the secondary task provided stronger evidence that the shared control autonomy mode was easier to use, but this secondary task also changed the nature of the task that the operator was asked to perform.

Behavioral entropy data was consistent with the conclusion that direct control required more work. Since behavioral entropy only required observations of operator activity (and did not require an intrusive secondary task), we could have used behavioral entropy without the secondary task to conclude that the shared control autonomy mode was easier to use than the direct control autonomy mode.

In summary, since higher entropy values occurred under direct control, the evidence supports the hypothesis that entropy allows us to identify which autonomy mode imposes higher human workload.

## 4 Case Study 2: Comparing the Usability of Two Interfaces

In the previous case study, we used behavioral entropy to measure the differences between two autonomy modes. In this case study, we determine whether entropy is a reliable method for determining which of two interfaces provides better support for robot teleoperation.

### Shared-Control Results

Participant	A	B	C	D	E	F	G	Ave.
% Neglect	74%	72%	77%	61%	73%	72%	74%	72%
% Performance	97%	88%	94%	98%	85%	92%	97%	93%
# per min.	12.0	12.4	10.3	12.1	13.8	16.3	15.8	13.2
% Correct	71%	63%	39%	94%	85%	88%	78%	74%
Entropy	0.37	0.49	0.45	0.32	0.39	0.55	0.29	0.41

### Direct-Control Results

Participant	A	B	C	D	E	F	G	Ave.
% Neglect	65%	70%	70%	34%	70%	68%	73%	64%
% Performance	83%	74%	96%	96%	88%	75%	81%	84%
# per min.	10.2	12.5	9.8	6.4	11.5	12.7	13.4	10.9
% Correct	57%	63%	38%	79%	71%	88%	77%	67%
Entropy	0.68	0.77	0.69	0.57	0.66	0.72	0.67	0.68

Table 2: Results from the simulated world.

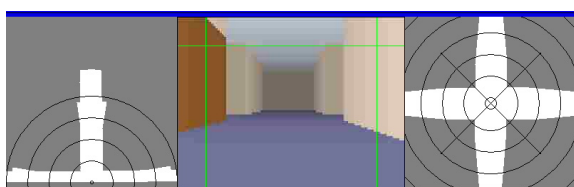


Figure 3: Interface that displays sensor readings side by side.

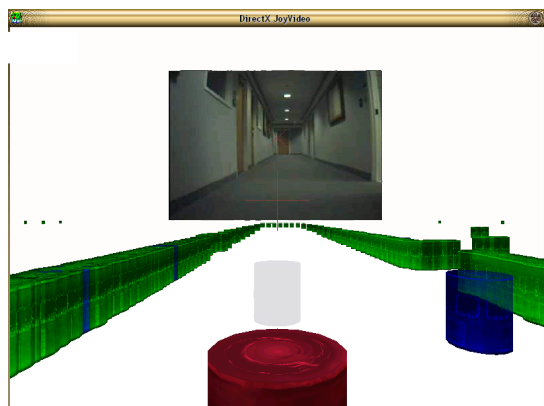


Figure 4: Interface that integrates sensor readings into perspective view.

The two interfaces are shown in Figures 3-4. The first interface displays, from left to right, laser range finder readings, video, and sonar in a side by side format. The second interface integrates these three sensor readings in a pseudo-perspective view, with a representation of the robot displayed in this view.

We conducted a series of experiments to compare the two interfaces. In a balanced experiment design with a randomized schedule, subjects teleoperate a simulated robot through three mazes while performing a memory task where they must remember five images. After completing the maze, subjects complete a memory test by se-

lecting the images they saw before from a list and putting the images in order.

## 4.1 Methods

As in the previous case study, the model of joystick angles was based on a Taylor series. Given studies on human control characteristics, we used a sample interval of 150ms and averaged all joystick angles within a 150ms window as our sample. Given the series of joystick angles, we created the prediction error density using the difference between the predicted value and the observed value. From the set of prediction error densities (one for each maze and for each interface), it is necessary to identify a baseline density from which bins are created. We did this by having each subject guide the robot through one maze without performing the memory task using the side by side interface. Prediction errors from this entire data set were then used to create the bins used to determine entropy using the technique described in the previous case study.

## 4.2 Results

The following data were collected for 32 subjects: time to guide a robot through a maze, behavioral entropy, average velocity, number of collisions, and performance on a memory task. (Most likely, the memory tasks were not hard enough because about 70% of the test subjects aced the memory task.) With the exception of the memory task, for which we did not get any meaningful data, all of these measures demonstrate that the new interface is effective for helping people control a robot.

Figure 5 summarizes the data, and shows that the side by side interface is inferior to the perspective interface for each measurement. These findings support the conclusion that behavioral entropy is a useful measure for determining when one interface is more difficult to use than another. Moreover, the data is strongly supported



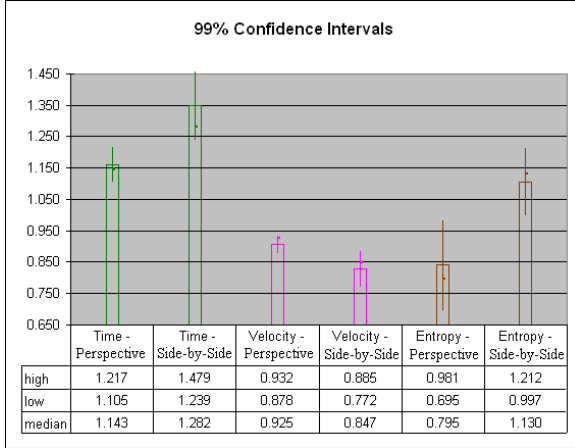


Figure 5: A comparison of the performance metrics averaged over all subjects and all test worlds.

by the number of collisions experienced; the number of collisions using the side by side display were more than doubled the number of collisions using the perspective display.

The perspective interface tends to be easier to use because it helps people predict where the robot will be heading and updates this information frequently. The side by side interface requires people to do their own prediction and only updates sensor values when new information is received. Using the side by side interface caused people to change their control input when new information was received from the robot. This causes the position of the joystick to be somewhat erratic and jump from position to position. People driving the perspective interface often make more frequent but less dramatic corrections. This can be attributed to lower workloads or finer control.

## 5 Case Study 3: Using Behavioral Entropy to Build an Interface

In this section, we present a case study that uses a real-time estimate of behavioral entropy as a major factor in constructing an estimate of driver workload. This workload estimate is then used to learn haptic control policies for an accelerator pedal that increase the safety of the driver without significantly increasing workload. We use the ability of reinforcement learning to detect patterns in stochastic reinforcers to support the conclusion that the real-time estimate of behavioral entropy contains useful information about when people have workload spikes.

## 5.1 Methods

In the experiment, subjects followed an erratic lead vehicle with and without the learned force profile. During the experiment, subjects solved two-digit arithmetic problems that appeared on the simulator by pushing buttons on the steering wheel.

We trained an artificial agent using satisficing Q-learning, a dual attribute version of the standard Q-learning algorithm, to minimize workload while preserving safety. This was done by creating the following dichotomous goals: Goal #1: Don't allow the vehicle to experience a crash or a near-crash. Goal #2: Reduce driver workload as much as possible. Clearly, these two goals are in conflict with each other whenever the vehicle is in a non-trivial situation. Goal #1 was realized by penalizing policies that lead to a collision or near collision. Goal #2 was realized by only rewarding actions that induced a low user workload. Both behavioral entropy and impedance (i.e. the extent to which interface actions directly opposed driver actions) were used to estimate driver workload and determine whether actions produced a workload low enough to be rewarded.

## 5.2 Results

A control policy for a force-feedback gas pedal was learned using the methodology described above. Entropy of the accelerator pedal position was calculated in real-time and combined with instantaneous impedance to form an estimate of driver workload. This estimate of driver workload was compared against baseline driving and empirically chosen thresholds to determine whether an action induced too much workload to be rewarded. The learning algorithm was trained during ten minutes of exploratory driving by a single operator who allowed several rear-end collisions to occur in order to propagate penalty data throughout the state space. The agent learned to balance driver workload with expected risk, applying forces to the pedal only in states where experience demonstrated it to be useful.

Test subjects responded enthusiastically to this haptic support. Pedal entropy remained similar to drivers that were in unassisted trials, but the overall safety (as measured by time spent with time to contact less than 0.7 seconds) was reduced by 45%. Using high entropy to prevent rewarding an action during the training period was very helpful in this context, as the agent learned a control policy that informed the user of danger without significantly increasing overall entropy. This evidence supports the conclusion that the online estimate of behavioral entropy contained useful information about the workload experienced by a distracted driver with and without force feedback support. This evidence is bolstered by plotting the prediction error density functions

and noting that the density corresponding to no forces is shorter and fatter than the other; these densities were shown in Figure 1.

## 6 Discussion and Future Work

In this paper, we presented three case studies that demonstrated how behavioral entropy can be used in HRI studies. These case studies showed that behavioral entropy reliably predicted workload and correlated well with other measures of human performance. The third case study also demonstrated how a real-time estimate of behavioral entropy provided useful information to a machine learning algorithm; this algorithm decreased the number of near collisions in a driving simulator without increasing subjective workload.

Two areas of future work need to be explored before entropy can be used widely. First, a guide for selecting parameters in the entropy computation algorithm need to be identified. These parameters include what models should be chosen, how model parameters should be chosen, how binning should be performed, and how a window size for real-time entropy estimates should be selected.

Second, the relationship between entropy and other human factors measures should be better established. This includes researching how average entropy or its variations (e.g., peak entropy, minimum entropy) correlate with, for example, trust, neglect tolerance, interface efficiency, and so on.

## Acknowledgements

The authors would like to thank DARPA (under contract #NBCH1020013) and Nissan Motor Company for their partial support of this work.

## References

- [1] David J. Bruemmer, Julie L. Marble, David D. Dudenhoefter, Matthew O. Anderson, and Mark D. McKay. Mixed-initiative control for remote characterization of hazardous environments. In *HICSS*, 2003.
- [2] 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*, Luasanne, Switzerland, 2002.
- [3] T. Fong, D. Kaber, M. Lewis, J. Sholtz, A. Schultz, and A. Steinfeld. Common metrics for human-robot interaction. Submitted to IROS 2004, 2004.
- [4] T. Inagaki and K. Inoue. Adaptive choice of a safety management scheme upon an alarm under supervisory control of a large-complex system. *Reliability Engineering and System Safety*, 39:81–87, 1993.
- [5] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 1987.
- [6] 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.
- [7] C. D. Wickens and J. G. Hollands. *Engineering Psychology and Human Performance*. Prentice Hall, third edition, 2000.
- [8] N. Wiener. *The Human Use of Human Beings: Cybernetics and Society*. Free Association Books, London, 1989.