

# Principles of Adjustable Interactions

Jacob W. Crandall and Michael A. Goodrich\*

## Abstract

In this paper, we seek to model natural human-robot interactions in human-robot teams that support adjustable autonomy and interactions. We present a theoretical characterization of interaction efficiency. We then perform case studies to validate this theoretical framework. Specifically, in one case study we compare interaction efficiency between a shared control teleoperation algorithm and traditional manual-control teleoperation. We then perform another case study in which we analyze the neglect tolerance of a point-to-point interaction scheme. The principles learned from these case studies should help to build more effective human-robot systems.

## Introduction

In many applications, it is desirable to allow a human to interact with multiple robots. These applications include search-and-rescue, exploration, hazardous waste clean-up, and so on. Unfortunately, there is a limit to how many tasks a human can manage in a given time. This means that the number of robots in a human-robot team is limited. To understand how many robots a human can manage effectively, it is (1) necessary to understand how humans interact with individual robots under varying circumstances and (2) necessary to understand the interactions (how, when and how long) that each individual robot requires from this human under varying circumstances. In this paper, we seek to identify some of the principles that govern human-robot interactions with emphasis on these two points.

Specifically, we seek to develop human-robot systems which support *adjustable interactions*. The term adjustable interactions is tightly connected with the term adjustable autonomy, only it is more descriptive, as it defines the ability of a system to change the interactions between a robot and a human by adjusting the robot's interaction scheme. Formally, an *interaction scheme* consists of the *autonomy mode* of the robot and the *interface* between human and robot. For each

interaction scheme, it seems natural that as a robot is neglected (in any interaction scheme short of pure autonomy), its performance degrades. Additionally, as environmental complexity increases, robot performance is also expected to decrease.

In this paper, we provide a theoretical framework for understanding how the expected performance of a particular interaction scheme changes as robots are neglected and as world complexity increases. This framework shows how the efficiency of human-robot interactions affects robot performance. We then present results from a case study that compares the neglect tolerance of two autonomy modes using identical interfaces (this case study can be found in (Crandall & Goodrich 2002), but we present it here as well). We then present results that more fully describe how the performance of a robot decreases as a function of neglect and environmental complexity for a particular interaction scheme. We conclude by discussing how the framework can be further validated and how the validated framework can be used to guide the design of human-robot systems.

## Related Literature

While different levels of autonomy have been studied extensively, research in teleoperation is most mature (Sheridan 1992). Perhaps the most difficult obstacle to effective teleoperation occurs when there are communication delays between the human and the robot. The standard approach for dealing with these issues is to use supervisory control. Work on teleautonomy (Conway, Volz, & Walker 1990) and behavior-based teleoperation (Stein 1994) are extensions to traditional supervisory control that are designed specifically to account for time delays. Of particular interest are approaches to behavior-based design of robots that can interact with humans. Arkin and Ali's work has been particularly relevant to our research (Ali & Arkin 2000). In their work, they show how potential fields can be used for shared-control teleoperation. They present experimental results for hundreds of test subjects of a shared-control system that allows a human to interact with a team of simple behavior-based robots. In measuring the effectiveness of human-machine interaction, much work has been done on operator workload. Of particular relevance is Boer's work relating workload and entropy (Boer *et al.* 1999). In addition, Boer has used secondary tasks to help evaluate the cognitive workload placed

---

\*J. W. Crandall (email: crandall@cs.byu.edu) is a graduate student at Brigham Young University and M. A. Goodrich (email: mike@cs.byu.edu) is an assistant professor of Computer Science at Brigham Young University.  
Copyright © 2002, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

on human operators.

## Interaction Efficiency

As stated in the introduction, one purpose of this paper is to present a theoretical framework for characterizing the efficiency of human-robot interactions. This framework is built on the intuition that the likely performance of a robot degrades as the human neglects the robot and as world complexity increases. Human-robot interactions should be frequent enough, last long enough, and be efficient enough for the robot to maintain acceptable performance levels.

## Framework

Consider the design of optimal controllers. The design of such controllers is the task of choosing a control law  $\pi$  that maps observations (states) of the environment  $s$  into actions  $a$  in such a way that performance is maximized (or cost is minimized). Formally and in our notation, the objective of an optimal controller can be stated as follows:

Maximize :

$$J(\pi) = E \left[ \sum_k \Phi(s_{k+1}) + \Lambda(\pi(s_k)) \right] \quad (1)$$

Subject to :

$$s_{k+1} = f(s_k, a_k) \quad (2)$$

where  $\Phi(s_k)$  is the payoff of visiting state  $s_k$  on a path to a goal,  $\Lambda(\pi(s_k))$  is the payoff for using control action  $a_k = \pi(s_k)$ , the sum indicates that performance is accumulated over time,  $f(s_k, a_k)$  is a model that describes how action  $a$  at time  $k$  translates the state  $s_k$  into a new state  $s_{k+1}$ , and  $E(\cdot)$  indicates an expectation. Expectation is included since the dynamics model may be probabilistic (e.g., as in Markov decision processes). An optimal control law  $\pi$  is the mapping from states to actions that maximizes the expected payoff subject to the constraint imposed by the way inputs change the state of the world.

In human-robot interaction, the behavior of the robot is produced by a control law that accepts human input. Thus, we generalize the notion of a control law to include the closed loop of human-robot interaction, and replace the term *control law* with the term *interaction scheme*. The interaction between a human and a robot is diagrammed in Figure 1 which illustrates the interface between human and robot as well as the autonomy loop between a robot and its world. Recall from the introduction that an interaction scheme consists of the *autonomy mode* of the robot and the *interface* between human and robot. The interface is made up of the *control element* used by the human to communicate information to the robot, and the *information element* used by the robot to communicate to the human. The autonomy mode refers to the closed loop behavior of the robot in the world, and the control and information elements refer to the closed loop behavior of the robot in the interface with the human.

In human-robot interaction, the action  $a_k$  is composed of both robot input and human input. Since human attention is switched between multiple tasks, the action  $a_k$  is not influenced by a human at every sample interval. The effective

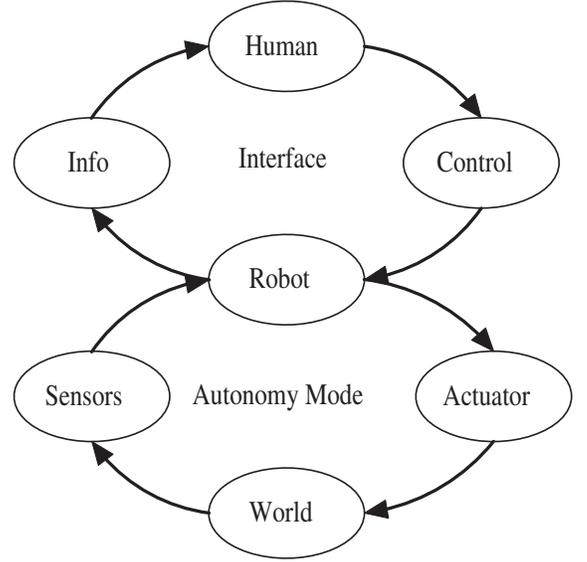


Figure 1: The interface loop and autonomy loop for human-robot interaction.

rate of interaction, defined loosely as the frequency that a human changes  $a_k$  and denoted by  $T$ , between the robot and the human is a random variable that strongly influences the performance  $J$ . Interaction schemes  $\pi$  that are designed for frequent human input will not produce high payoffs when humans interact less frequently.

In addition to the influence of  $T$ , the expected performance  $J(\pi)$  of a particular interaction scheme  $\pi$  is also affected by how the world responds to robot actions. The manner in which the world responds is encoded in Equation (2) as the function  $f(s_k, a_k)$ . Since many of the worlds in which robots will benefit from human interaction are highly dynamic and complex, the environment function  $f$  is a random process. Interaction schemes that are designed for a particular level of environmental complexity may not perform well for other environment complexities.

In Equation (1), the expected payoff  $J$  for a particular interaction scheme  $\pi$  is a scalar value, but when the influence of the interaction rates  $T$  and the world characteristics  $f$  are taken into consideration,  $J(\pi)$  becomes a random process that is influenced by the random variable  $T$  and the random process  $f$ . We will restrict attention to fixed domains whence we assume that the qualitative characteristics of  $f$  stay the same, but the complexity of the environment, denoted by  $C$ , can change. Additionally, we extract from  $T$  two variables:  $t_{\text{off}}$ , which is the time since the last human-robot interaction, and  $t_{\text{on}}$ , which is the time since the human began to *service* the robot (*servicing* refers to the act of the human giving attention and input to the robot). From these variables we obtain the random process  $J(\pi)$ :

$$J(\pi, C, T) = \begin{cases} J_S(\pi, C, t_{\text{on}}) & \text{if servicing} \\ J_N(\pi, C, t_{\text{off}}) & \text{otherwise} \end{cases}$$

Figure 2 shows the trends that are expected for both  $J_N$ . ( $J_S$  is not shown here. We will discuss it later in the paper.) As

neglect or complexity increases, it is expected that performance for an individual robot will decrease. Additionally, as complexity increases, it is expected that more *servicing* time (i.e., more interactions) will be required for performance to be brought back up to high levels.

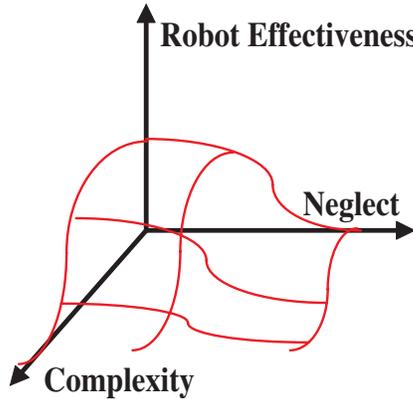


Figure 2: Performance  $E\{J_N(\pi, C, t_{\text{off}})\}$  of interaction scheme  $\pi$  as a function of neglect and world complexity.

### Performance Depends on Neglect

To enable a human to manage multiple tasks (including interacting with multiple robots), it is necessary to know how long a human can give attention to one robot before the performance of the other tasks deteriorate. The relationship between neglect and expected performance can be characterized using the neglect curve illustrated in Figure 3 (top) for a human-robot system under various autonomy modes.

The idea of the neglect curve is simple. Interaction scheme A's likely effectiveness, which measures how well the human-robot system accomplishes its assigned task and how compatible the current task is with the human's objective, decreases when the human turns attention from the task to a secondary task; when the task is neglected the interaction scheme becomes less effective.

Neglecting a robot is analogous to interacting with a remote robot over a communication channel that suffers from time delays and intermittency. Time delays are a common problem that arise in much of the literature on operating a remote robot. For example, round trip time delays between earth and Mars are on the order of 40 minutes, between earth and the moon are around 5 seconds, and between a laptop and a robot over a local wireless ethernet up to one second. Since neglect is analogous to time delay, we can use techniques designed to handle time delays to develop a system with adjustable autonomy.

The neglect curve can be used to determine how often we would expect interactions to occur to maintain a level of performance; see Figure 4. In the figure, neglect tolerance is displayed as a function of time-off-task. To prevent the performance of an interaction scheme from dropping below an acceptable level, the robot can only be neglected for a certain period of time defined as the time spent off the task plus

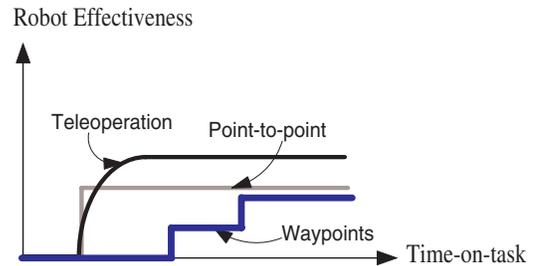
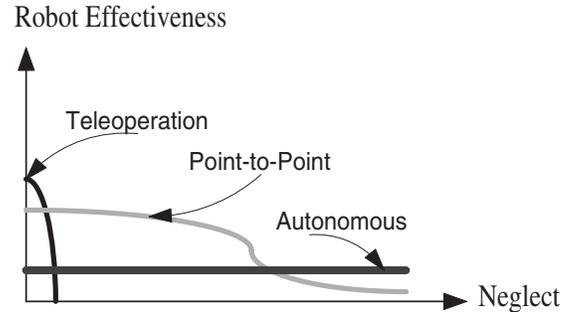


Figure 3: The neglect curve (top) is a plot of  $J_N(\pi, c, t_{\text{off}})$  for constant complexity  $c$  as a function of  $t_{\text{off}}$  (which is the *Neglect* axis). The nearly vertical curve represents an interaction scheme which includes the potential for great effectiveness but which fails if the operator neglects the robot. The horizontal line represents a fully autonomous robot which includes less potential for effectiveness but which maintains this level regardless of operator input. The sloping curve represents intermediate types of interaction for which effectiveness decreases as neglect increases. The graph of  $J_S(\pi, c, t_{\text{on}})$  (bottom) shows expected performance increase for a robot as it is being serviced. The fully autonomous interaction scheme isn't shown (fully autonomous robot don't get serviced). We substitute in a waypoints scheme as a highly autonomous (yet not fully autonomous) interaction scheme.

the time spent on the task bringing the performance back to a high level. The acceptable neglect time (time-off-task) includes both the time spent on other tasks as well as the time to switch attention.

### Performance Depends on Interactions

Each interaction scheme requires different interactions; see Figure 3 (bottom). Variations in frequency of interactions and duration of interactions effect how well a robot performs. The previous section discussed how expected (mean) robot performance decreases over time as a robot is neglected. Efficient interactions at key times can help the robot maintain high performance levels. If sufficient interactions are allowed, the neglect tolerance for an interaction scheme increases. As is seen from the figure, different interaction schemes effect robot performance differently.

It should be noted that, in addition to the above discussion, many aspects of interactions for an interaction scheme are closely related to interface elements. We'll discuss that in more detail later on.

### Performance Depends on Complexity

To illustrate how world complexity can impact performance, consider the scenarios diagrammed in Figure 4. In the figure, worlds of two different complexities are illustrated. For each world, the neglect tolerance curve is a function of the number of branches and amount of clutter. If the world has minimal clutter and very few branches, then the robot can be neglected for an extended period of time. If, however, the world is cluttered and has many branches, then uncertainty will increase causing the robot to be less tolerant to neglect. Thus, performance decreases as complexity and neglect increase.

### Change in Performance Depends on Information and Control

Given the curves that describe the expected performance of interaction as a function of neglect and complexity,  $J(\pi; C, T)$ , it is appropriate to explore how presenting information affects this efficiency. An information system can increase neglect tolerance primarily by decreasing the amount of time required for the human to switch attention from another task and gain relevant situation awareness for the particular robot. The information presented by such systems performs three objectives: it triggers an attention switch from a secondary task to a relevant robot interaction task, it speeds up the time to switch between the secondary task and the interaction task by helping the human get "in the loop" faster, and it helps the human perform the task more quickly thereby decreasing time-on-task. Unfortunately, a poorly designed information system may cause the process of gathering information to become a task in and of itself. This effectually extends the time to switch from a secondary task by compelling the human to attend to the information task before attending to the primary decision task.

Similar to the way in which information can change the characteristics of interaction, the manner of giving information to the robot also changes these characteristics. For example, if a control scheme is very complex, the human may

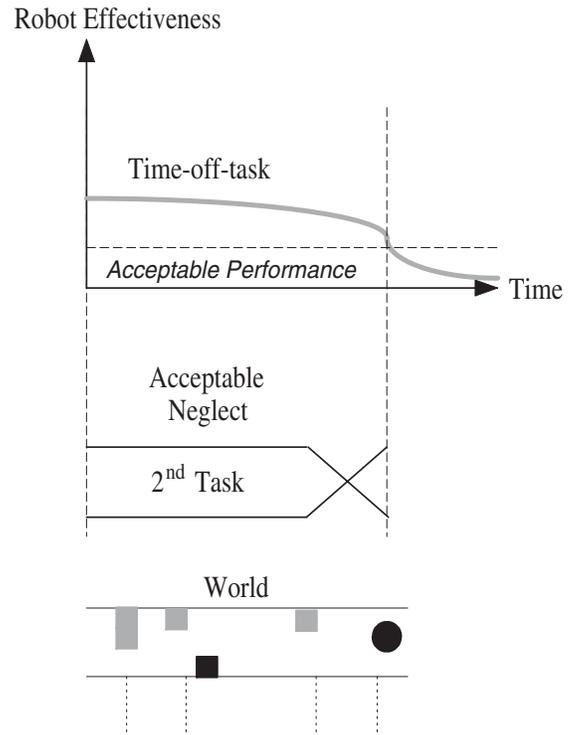


Figure 4: The amount of time that a particular autonomy mode can operate above an acceptable level of performance depends on the complexity of the world. Without the gray obstacles, the world is relatively clutter free and has no decision points, so the robot (represented by the dark oval) can be neglected for an extended period of time. If the gray clutter or dashed paths are added, the neglect curve decreases faster so the neglect tolerance decreases.

have difficulty forming an efficient mental model of the interaction. Without an efficient mental model, the process of presenting information to the robot may become a task in and of itself. This effectually extends the time to switch from a secondary task by compelling the human to attend to the control task after attending to the primary decision task.

### Shared Control

The purpose of this section is to explain the shared-control teleoperation system that we have created and show how changing an interaction scheme changes neglect tolerance. The development of this system was described in (Crandall & Goodrich 2001), but we will review this algorithm and present more complete experimental results in this paper. The system consists of a Nomad SuperScout robot and a remote computer. The remote computer and the robot are connected via an 11Mb/s wireless ethernet. A GUI displays video, sonar readings, and compass information from the robot. Through a Microsoft SinderWinder II Force Feedback Joystick, the human guides the robot.

Our approach to shared-control teleoperation uses a variant of potential fields. In the algorithm, the angle of each

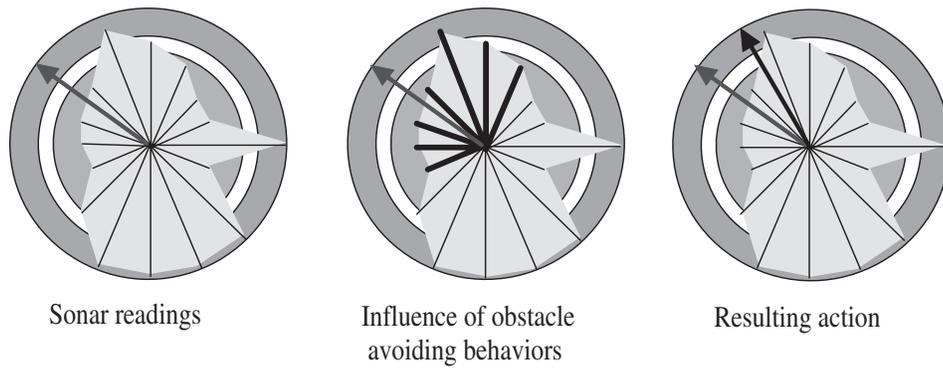


Figure 5: A graphical depiction of the algorithm for a robot positioned in a hallway with a door open on the robot's right. Raw sonar readings (left) are translated into relevant behaviors (middle) and combined with the human input to produce the actual robot action (right).

sonar is associated with a behavior. Sonars that measure nearby obstacles return repelling behaviors, and sonars that measure open spaces return attracting behaviors. More specifically, sonar distances are classified into three categories: repelling, neutral, and attracting. If the sonar returns a distance greater than a pre-defined *safe distance* (65 inches in our experiments) then the corresponding behavior is categorized as an attracting behavior. If the sonar returns a distance less than a pre-defined *risk distance* (40 inches in our experiments) then the corresponding behavior is categorized as a repelling behavior. For other sonars, the corresponding behavior is categorized as a neutral behavior.

Given these categorizations, the attracting behaviors are assigned strengths according to how close their angles are to the human input. Angles that are nearby are given large strengths, and angles that are far away are given zero strength. Similarly, the repelling behaviors are weighted by how close their angles are to the human input. However, unlike attracting behaviors, the strength of each repelling behavior is also weighted by the distances they return; small distances indicate obstacles that are very close and are therefore given high strength. After the strength of each behavior is obtained, the behavior vectors are summed with the human input vector to produce the resulting direction that the robot will move. The strengths used in the experiments presented herein are given in (Crandall & Goodrich 2001).

This process is illustrated in Figure 5. In the figure, the human tells the robot to go forward and left (see the image on the left). Sonar readings that are relevant are identified (see the image in the middle). Those behaviors that would move the robot toward an opening (as indicated by the sonar reading terminating in the outer shaded circle) in the world pull the robot toward the opening, and those behaviors that would move the robot toward a nearby obstacle (as indicated by the sonar reading terminating in the inner shaded circle) push the robot away from the obstacle. These pulls and pushes are combined with the human input to specify the direction that the robot will go (see the image on the right); in the example, the robot will still go forward and left, but will not go as far to the left as suggested by the human.

Since vector summation in a potential fields algorithm allows for some obstacle-avoiding behaviors to cancel out, sometimes undesirable emergent behaviors occur. In our case, under certain circumstances, the robot can be directed into an obstacle. To avoid this, we include a safe-guarding (Fong, Thorpe, & Baur 2001; Krotkov *et al.* 1996) behavior, which can veto the direction. Using all sixteen sonar readings we define a *safe region* by simply finding the points at which the sonars indicate that there are objects. Connecting these points yields a polygon with sixteen sides, which makes up the safe region. By predicting where the robot will be at some future time  $t$ , the robot can determine if it will leave this region anytime in the near future if it continues the course it has selected. If the robot thinks it will leave this safe region anytime in the near future, the direction is vetoed and the robot defaults to a behavior that causes the robot to rotate slowly in place towards the nearest perceived clear pathway.

### Shared Control Teleoperation: Case Study 1

In this section, we present an experiment to compare the shared-control teleoperation system described above with a direct-control teleoperation system. The two schemes both use a joystick as the control element and both use a video display and graphical depictions of sonar readings as the information element. The interaction schemes differ by the autonomy mode, shared control or direct control. First, we describe the experiment and then we explain the criteria and results.

### Experiment Description

The primary task in the experiments is to guide a robot through a cluttered course with simple decision points. The course is illustrated in Figure 6. In experiments involving human cognitive load, experiment participants are sometimes asked to perform a secondary task (or tasks) as they perform a primary task (Boer *et al.* 1999). In our experiment, subjects must solve two digit addition problems while performing the robot guidance task.

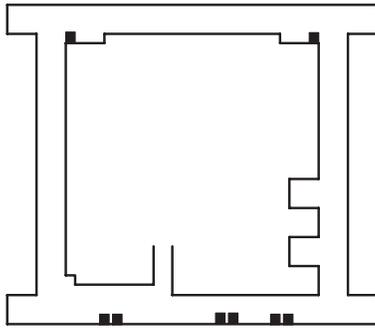


Figure 6: *The environment used to measure neglect. Notice the nominal amount of clutter.*

As a rule, experiment participants should not have experience driving the robot. This ensures that no biases are introduced due to past training. For each participant, the following steps are followed:

**Step 1.** The math proficiency level of the participant is determined. Two digit addition problems are displayed on the screen along with four multiple choice answers (only one being the correct answer). The participant is given 5 seconds to answer the question. A log of math proficiency is kept. After the participant answers the question, he or she may proceed to a new problem by clicking on a button. This proficiency test lasts for two minutes. If the participant cannot successfully complete 60% of the problems, the difficulty level is reduced to adding a two-digit number to a one-digit number.

**Step 2.** Next, the participant must be trained to guide the robot using a particular autonomy mode. Scheme S is the Shared-control teleoperation scheme, and Scheme D is a traditional Direct-control teleoperation scheme. In order to not bias results, some participants are trained and tested on Scheme A first, and others are trained and tested on Scheme B first. After completing initial training, the participant is asked to guide the robot through the course as quickly as possible. While doing so, he or she must look out for the safety of the robot. Training is complete when the subject has successfully guided the robot through the course one time.

**Step 3.** The participant is asked to again guide the robot through the course. This time, the participant is asked to do math problems as he or she drives the robot. The participant is instructed to guide the robot through the course as quickly as possible, and to answer as many math problems in this time as he or she can, while making sure the robot is safe.

**Steps 4–6.** The participant repeats steps 2–3 using the other control scheme. That is, if the participant started with Scheme S, then he or she is next tested on Scheme D and vice versa.

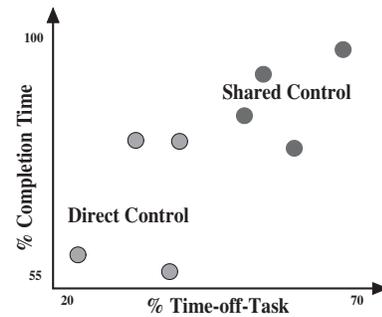


Figure 7: A plot of robot effectiveness versus neglect rate. The vertical axis represents robot performance (as a percentage to maximum effectiveness) and the horizontal axis represents neglect (in terms of percentage of time off task).

## Evaluation Criteria and Results

In this experiment, we fix the level of complexity and explore how interaction efficiency is affected by human neglect. The best interaction scheme for a given level of complexity is the system that can move the knee of the neglect curve as far to the right as possible. In general, a lower workload imposed by an interaction scheme means the operator is free to neglect the robot more. This, in turn, means that the knee of the curve will be moved right, assuming that performance level doesn't decrease. There are several ways that we show neglect and workload in our system, and these measurements and results are described in the following subsections. Figure 7 shows robot effectiveness versus neglect for the task performed in the experiment. It is interesting to note from this graph that the shared-control system (represented by the outlined circles) dominates the direct-control system (represented by the dark circles) for each participant on the given task.

**Neglect Rates** Neglect time is the amount of time spent doing other tasks. Thus, neglect is the time spent solving arithmetic problems divided by the total time of the trial run.

In the experiments, the four participants were able to neglect the robot an average of 50% more using shared control than direct control.

**Joystick Steering Entropy** We obtain the joystick steering entropy for each participant using the algorithm described in (Boer *et al.* 1999). Slight adjustments are made to this algorithm, but they are small. Note, however, that entropy data from this paper should not be compared to entropy readings in (Boer *et al.* 1999). Entropy ratings range between 0 and 1. A high entropy rating means that joystick movements are choppy and thus indicates that the operator is under a higher workload. Thus, lower entropy ratings indicate that the operator has a lower and more manageable workload.

In the experiments, joystick steering entropy was considerably higher on this task for the direct-control system. On

average, entropy increased by just over 50% when the direct-control system was used. This indicates that the cognitive workload was higher for direct-control than shared-control. These results are consistent with the results on neglect rates, since workload and neglect rates should have a significant negative correlation.

**Primary Task Effectiveness** This is how well the participant did in driving the robot through the course. To keep things simple and objective, the judgement of how well a task was performed is established simply by how much time it takes to get the robot around the building. The distance the robot is required to travel and maximum robot speed dictates that it take at least 170 seconds to get through the course. We base performance off this number:  $Performance = \frac{170}{TimeElapsed} \times 100$ .

In the experiments, performance levels for the shared-control system exceeded performance levels of the direct-control system by an average of about 35%.

**Secondary Task Effectiveness** This is a measurement of how well the participant performed on the arithmetic problems. Both the number of problems completed per minute and the problem proficiency are important. Since each participant's math abilities differ, only comparisons between how well a participant performed in different control schemes is relevant. It is theorized that participants should perform better on the secondary task when they have a lower workload imposed by the primary robot control task.

In the experiments, the secondary task results correlate with the results of all the other recorded data for this experiment. The average arithmetic proficiency on the shared-control system exceeded the average arithmetic proficiency on the direct-control system by 9%. Additionally, the average number of arithmetic problems attempted per minute increased from 7.3 problems per minute when participants used the direct-control system to 12.0 problems per minutes when participants used the shared-control system. That represents an increase of about 65%.

**Subjective Rating** Each participant is asked to tell which system was better. The judgement criteria of what is better should be based on a general perception of how the participant felt they did on each scheme.

In the experiments, the participants of the experiment unanimously have indicated that the shared-control system is better than the manual-control system for the task tested in the experiment.

Further tests in similar simulated worlds varified more fully the data we obtained in the real world experiments, and also showed that the simulator we have developed is real-world-enough to obtain valid results.

## Analyzing the Point-to-Point Interaction Scheme: Case Study 2

The purpose of this section is to further validate the theoretical framework developed in section 3. In this section we

describe a point-to-point (P2P) interaction scheme. We then present results obtained from running a simulated robot in multiple worlds with different complexities to obtain data which estimates the random process  $J_N(P2P, C, t_{off})$ .

### The Point-to-Point Interaction Scheme

The point-to-point autonomy mode uses the shared control algorithm described in the previous section. The difference is that the robot gets input through either a mouse or speech interface instead of a joystick. In this control scheme, the human operator pushes buttons to tell the robot what to do at the next branching point that it finds. The robot can be told to either turn right, turn left, or go straight through the intersection. In addition, the human can direct the robot to go backwards, spin right, or spin left.

This interaction scheme is more autonomous than the teleoperation system described in the previous section because the human must only provide an input vector for every branching point (i.e., intersection) that the robot faces. Additionally, if the robot gets turned around by obstacles, misses an intersection, or thinks it has found an intersection when it, in reality, has not, the operator must provide additional help. When the robot is not at a branching point, it automatically inputs a "straight ahead" vector into the algorithm described in the previous section to obtain its new direction. In this way it can move through somewhat cluttered environments effectively. When the robot arrives at a decision point it inputs a vector into the system that should cause it to do what the human indicated (for example, to take a right-hand turn), after which it notifies the human that it has fulfilled that command.

In summary, the human operator must only tell the robot what he/she wants it to do at the next intersection (or branching point). At that point, he/she may neglect the robot and perform other tasks that need to be done. The robot then performs the necessary command when it comes to what it thinks is a branching point. The robot then sends notifies to the operator that it has completed the command.

### Obtaining the Neglect Random Process

We next explain how we measure the random process  $J_N(P2P)$ . We first describe how to estimate complexity. We then discuss how we obtain performance measurements. Lastly, we explain the experiments we ran to obtain the results.

**Estimating Complexity** The complexity of an environment consists of two things. First, complexity is a function of the amount of clutter in an environment. It is generally more difficult for a robot to traverse its environment in the desired directions when there are more obstacles to go around and pass through. Second, complexity is a function of the number of branches (points at which the human and robot must decide which way the robot should go) per area. Complexity can be difficult to measure in real time. One reason for this difficulty is that one moment a robot may be in a very complex environment, full of branches and obstacles, and the next moment it may be in an environment that has few intersections and obstacles. Additionally, although

complexity is real, it is a somewhat subjective notion. Thus, complexity can only be estimated. However, we can obtain fairly good estimates.

In these experiments, we use entropy measurements to estimate the complexity of an environment. We use sonar entropy  $E_S$ , speed entropy  $E_V$ , and turning entropy (how often the robot changes directions)  $E_T$ . More formally

$$C = w_s E_S + w_v E_V + w_d E_T$$

where  $w_s$ ,  $w_v$ , and  $w_d$  are the weights that each different kind of entropy has on environmental complexity. The result  $C$  is a value between 0 and 1. Lower values indicating lower environmental complexity.

This formula for obtaining environmental complexity proves to be quite effective as complexity calculations seem to be fairly accurate in determining how complex an environment is. However, future work should include a more complete analysis of ways in which world complexity should be estimated.

**Obtaining Performance Measures** In most real world applications, it is difficult to obtain the performance of a robot from moment to moment. In this application however, we are able to fix goals in the environment and calculate distances so that we can see how the robot is doing in approaching its goal. This will provide us with the ability to better estimate in real time the performance of a robot (within a certain confidence interval) in environments in which the performance of a robot cannot be determined at all times.

**Experiment and Results** The task of the experiment was to get the robot to a designated location in the world from a fixed starting location. Again, we fixed the information element. The operator was provided with a god's eye view of the world as well as the position of the robot in that world. Additionally, the sensor information given to the operator from the robot was the 16 sonar values, compass readings, and video images updated in real time. Seven different worlds of varying complexity were used. Figure 8 shows two of these worlds. These worlds included environments with low clutter and low branching, low clutter and high branching, high clutter and low branching, and high clutter and high branching so as to model many kinds of environmental complexity. No secondary tasks were used in these experiments. Future work should include the use of multiple robots and secondary tasks to better analyze the affect that increased workload has on any random process  $J(\pi)$ .

The operator was given as much time as was needed to service the robot. When the operator was done servicing the robot, he/she clicked a button, and the robot was neglected for a about 40 seconds (in further experimentation we can see what would happen after 40 seconds, but we chose to cut it off at that point in this experiment because it seemed like a good amount of time). At that point, the operator again serviced the robot. Data was gathered throughout all of the experiments and combined to obtain the results.

The results are in Figure 9. The figure includes the mean and the variance of the random process  $J_N(P2P, C, t_{\text{off}})$ . Although the data is relatively noisy, general trends are obvious. The data matches the hypothesized shape of Figure 2.

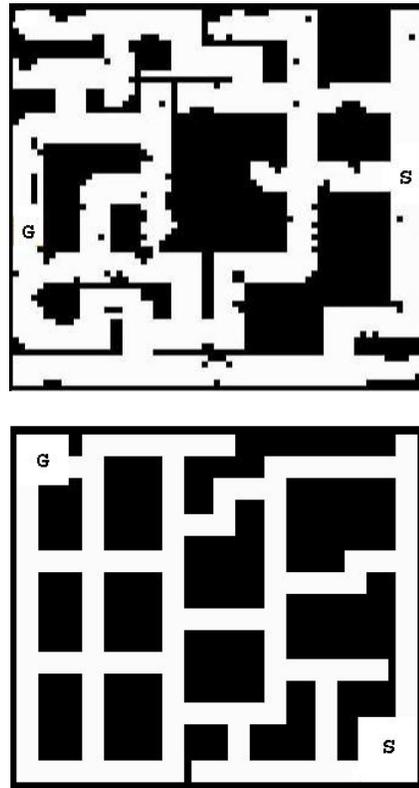


Figure 8: Two typical worlds used in the experiments. The symbol S refers to the robot's start position and the symbol G represents the robot's goal position.

The plot of the expected (mean) performance of the robot (top) shows that as the robot is neglected, performance decreases over time. Additionally, as the environment becomes more complex, performance decreases as well. In very complex environments, the robot performs quite poorly in this interaction scheme. Depending on the workload requirements of the system, this interaction scheme may not be appropriate for such worlds/environments.

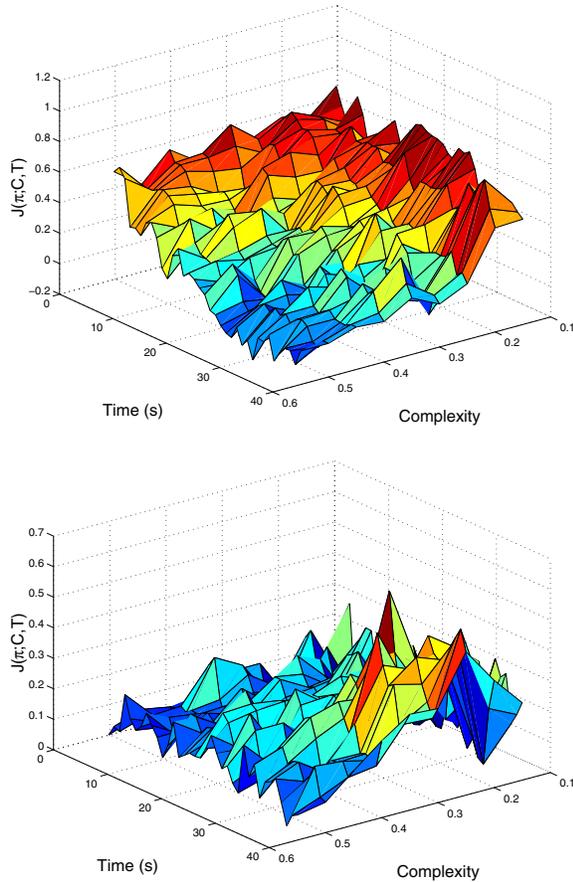


Figure 9: The mean (top) and the variance (bottom) of the random process  $J_N(P2P, C, t_{off})$ . The results are somewhat noisy, but the trends are clearly exist.

The variance graph of the random process (bottom) gives further insight into the nature of this interaction scheme. Even for environments of low complexity, the robot performs poorly quite frequently as indicated by the high variance (after the robot is neglected for a short time). This could indicate that even for low environmental complexities, the interaction scheme is not very neglect tolerant. It is, however, much more neglect tolerant than teleoperation. Further analysis of confidence intervals and analysis of the types of random variables contained in the random process is needed. This will be done in future work.

It should be noted that the random process  $J_N(P2P)$  was obtained when the robot was serviced for sufficient periods of time. By sufficient it is meant that the operator was given sufficient time to service the robot (i.e., the robot was

brought to high performance levels before it was neglected). This random process is not necessarily an accurate estimate for instances in which a robot isn't serviced sufficiently.

### Obtaining the Interaction Random Process

We have not yet obtained estimates for the random process  $J_S(P2P)$ . This is future work that we hope to complete in the near future. We expect that increased servicing time will be necessary for robots found in more complex environments. Analysis of this subject is very important to understand the nature of the point-to-point (and any other) interaction scheme.

Additionally, we would like to study how different interfaces (control element and information element) change the  $J_S(\pi)$  random process. For many applications it would have a significant influence.

### Summary and Future Work

We have presented a framework for evaluating the expected efficiency of an interaction scheme in terms of its sensitivity to neglect and environmental complexity. We then performed a case study that evaluated the observed interaction efficiency of a shared control teleoperation algorithm and compared this efficiency to the efficiency of direct teleoperation. We showed that, for the level of complexity used in the experiments, the shared control scheme was more tolerant to neglect. These results correlated well with measures of human workload and ease of use, suggesting that the framework is valid in some cases. Additionally, we performed experiments for a point-to-point interaction scheme in which we analyzed the neglect tolerance of this scheme in many worlds with different complexities. From these experiments, we extracted an estimate to the neglect random process, which shows the neglect tolerance for this interaction scheme for different world complexities. The results further validated the framework.

Future work includes further validation of the framework by conducting more experiments that control both neglect levels as well as complexity levels. We also desire to further analyze the data that we already have to find trends and principles that will help us better understand the interactions that must occur for both human and robot to perform effectively in varying environments. These experiments and analysis will allow us to identify design principles to create efficient teams of robots that interact effectively and naturally with a human.

### Acknowledgment

This work was partially funded by the DARPA Synergistic Cyber Forces project.

### References

Ali, K. S., and Arkin, R. C. 2000. multiagent teleautonomous behavioral control. In *Machine Intelligence and Robotic Control, Vol 1, No. 2*, 3–10.

Boer, E. R.; Nakayama, O.; Futami, T.; and Nakamura, T. 1999. Development of a steering entropy method for

evaluating driver workload. In *International Congress and Exposition*.

Conway, L.; Volz, R. A.; and Walker, M. W. 1990. Teleautonomous systems: Projecting and coordinating intelligent action at a distance. In *IEEE Transactions on Robotics and Automation* 6(2).

Crandall, J. W., and Goodrich, M. A. 2001. Experiments in adjustable autonomy. In *IEEE Systems, Man, and Cybernetics*.

Crandall, J. W., and Goodrich, M. A. 2002. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Fong, T.; Thorpe, C.; and Baur, C. 2001. A safeguarded teleoperation controller. In *IEEE International Conference on Advanced Robotics (ICAR)*.

Krotkov, E.; Simmons, R.; Cozman, F.; and Koenig, S. 1996. Safeguarded teleoperation for lunar rovers: from human factors to field trials. In *IEEE Planetary Rover Technology and Systems Workshop*.

Sheridan, T. B. 1992. *Telerobotics, Automation, and Human Supervisory Control*. The MIT Press.

Stein, M. R. 1994. *Behavior-Based Control for Time-Delayed Teleoperation*. Ph.D Dissertation, University of Pennsylvania.