

Measuring the Intelligence of a Robot and its Interface

Jacob W. Crandall and Michael A. Goodrich
 Computer Science Department
 Brigham Young University
 Provo, UT 84602
 (crandall, mike)@cs.byu.edu

Abstract— In many applications, the measure of a robot’s intelligence is its usefulness to a user. This implies that a measure of a robot’s intelligence is a measure of how well a human and a robot work together. In human-robot teams, two components determine team efficiency: *neglect tolerance* and *interface efficiency*. In this paper, we a) present an *evaluation technology* which uses secondary tasks to obtain measures of these two components, b) develop the related metrics of instantaneous robot performance and world complexity, and c) evaluate three systems using these measures.

I. INTRODUCTION

Fully autonomous robots do not meet the needs of most users. Rather, most users want robots that will help them accomplish a job. These robots must be able to interact effectively with humans as well as perform tasks semi-autonomously. To date, many robotic systems exist at one of two extremes. At one extreme are systems with purely teleoperated robots, where a human is always attending to a robot and the robot has very little autonomy. At the other extreme are systems with so-called fully autonomous robots that can be programmed and left to do a job, but frequently need to be reprogrammed or re-engineered since systems fail or need to be updated.

Between these two extremes are a set of systems with robots that are autonomous enough to do a lot of work, but require interactions with humans to accomplish meaningful tasks. We want to measure the effectiveness of these systems. There are two components which determine the usefulness of these systems: how much the robot can do autonomously and how much the robot supports human interaction. We capture these notions in two metrics: *neglect tolerance* and *interface efficiency*.

In order to obtain these measures, we first develop the related metrics of instantaneous performance and world complexity. We use these related metrics in an evaluation technology that can be used to estimate interface efficiency and neglect tolerance. The evaluation technology estimates measures of neglect tolerance and interface efficiency by using secondary task experiments in user studies.

In this paper, we will first discuss work related to this topic. In section 3, we will describe neglect tolerance and interface efficiency in human-robot systems and their related metrics. In section 4, we will describe

an evaluation technology for obtaining the measures described in section 3. In section 5, we will describe and evaluate three human-robot systems using this evaluation technology, which includes a user study involving 40 test subjects. Finally, we will summarize the contributions of this paper in section 6.

II. RELATED WORK

Conway et al. in [4] presents a taxonomy of human-machine interaction. The taxonomy includes teleoperation, shared control, traded control and supervisory control. Sheridan discusses both teleoperation and supervisory control in detail in [16]. Various forms of shared-control have been used [7], [15]. Traded control has become popular to avoid undo burden on the operator [9]. Traded control, however, presents serious challenges both from the human’s and the robot’s perspective [11].

Arkin’s group has done a lot of work in robot teaming. Such work includes the teleoperation of a group of robots by a single input from an operator [1]. This same idea was used in [10] for telemanipulation. Goldberg’s work in [8] is related to this idea. However, instead of having one operator control multiple robots, Goldberg has many operators control one robot. This is important because it provides a foundation for multiple user/multiple robot interactions.

A powerful notion in human-robot interaction is adjustable autonomy, which captures the notion that the autonomy level of a robot can be changed. This principle has been used extensively in the literature (e.g., [6], [13]). An important principle related to adjustable autonomy is that of mixed-initiatives [14], which poses the question of who has control in a system at a given moment and who is responsible for initiating control transitions. Scerri and associates have developed methods which address the issues of adjustable autonomy and mixed-initiatives in [12].

III. ASSESSING HUMAN-ROBOT INTERACTIONS

In a situation in which a human interacts with a remote robot over a communication network, there exist two different loops involving three different agents: the human, the robot, and the interface between the human and the robot. The first loop involves the human and the interface. Information about the robot and its en-

environment is delivered from the interface to the human. The human processes this information and determines a course of action which he/she believes should be done. The human’s desired course of action is then communicated to the interface through a control element. The second loop involves the robot and the interface. The interface communicates the human’s input to the robot. The robot then combines this input with its artificial intelligence to act in its world. The robot receives information about the world through its sensors which it forwards to the interface

A lesson learned from process automation is that designing a system without consideration for human factors frequently fails [2], even when humans are well-trained and highly motivated. Therefore, attention should be focused on making the interface and the robot more intelligent in the sense that they support human interaction. Within this context, we define an *interaction scheme* as an *autonomy mode* of the robot and an *interface* between the human and the robot. In order to design a new interaction scheme, we can manipulate either the interface or the robot’s artificial intelligence (e.g. autonomy mode). To be able to compare various interfaces and autonomy modes, we need a way of measuring which ones are better. In the rest of this section we discuss the elements that determine these measures.

A. Neglect Tolerance

Neglect tolerance is a measure of the effectiveness of a robot’s autonomy mode. This term is used to refer to the way in which a robot’s expected performance changes when it is neglected by humans (i.e., when human attention is focused elsewhere). As a general trend, as neglect increases, robot performance decreases. How much robot performance decreases depends on the interaction scheme that is being employed. Figure 1 conceptualizes how one might expect neglect to affect performance for different kinds of interaction schemes. In the figure, the performance of an interaction scheme using a teleoperated robot degrades quickly as the human neglects the robot. The performance of an autonomous robot does not tend to degrade much over time, although its peak performance usually would not be expected to be as high as a teleoperated robot.

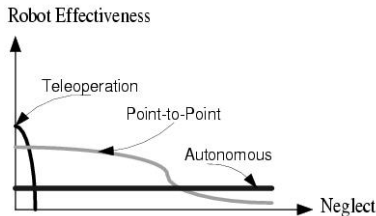


Fig. 1. Hypothesized neglect tolerance of interaction schemes with various autonomy modes for a world of constant complexity.

As discussed in the introduction, teleoperation and full autonomy lie on the extremes of human-robot interactions. There exist a large number of autonomy modes which require different degrees of interactions and are

represented in Figure 1 by a point-to-point scheme in which a robot is given a command, such as “turn left at the next intersection,” and is then expected to carry that command out autonomously, after which more interactions are required.

B. Interface Efficiency

Interface efficiency is a measure of the effectiveness of an interface. When a human operator’s attention is turned to a robot (we use the phrase *servicing the robot* to describe this action), we would expect the robot’s performance to change, hopefully for the better. The way in which the robot’s performance changes during servicing depends on the interaction scheme being employed. The interface of an interaction scheme affects the time it takes for a human to gain relevant situation awareness, decide on a course of action, determine the inputs to give to the robot, and then communicate those inputs to the robot.

A poorly designed interface may cause the process of gathering information by the human to become a task in and of itself. Consider an extreme example in which information about obstacles around a robot is communicated to the human operator via text. In such a situation, the human operator must read the information and create a mental representation of the world around the robot (which could take considerable time) before generating a plan about how to deal with the obstacles. Thus, an interface from which information is hard for the operator to extract extends the time for the human to switch from one task to another.

Figure 2 shows how interface efficiency could hypothetically affect the performance of a robot for different interaction schemes. The figure expresses the idea that changes in an interaction scheme affect the way in which the performance of a robot changes during interactions.

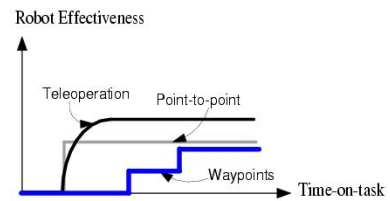


Fig. 2. Qualitative representations of interface efficiency for various presentations of information.

C. World Complexity

Up to this point, we have ignored the effects of world complexity on neglect tolerance and interface efficiency. Consider, however, the two worlds shown in Figure 3. It seems obvious that it would be easier for a robot to navigate through world *b* than to navigate through world *a*. Thus, the complexity of the robot’s environment affects robot performance. Interaction schemes that are designed for a particular level of world complexity may not perform well for other world complexities. Intu-

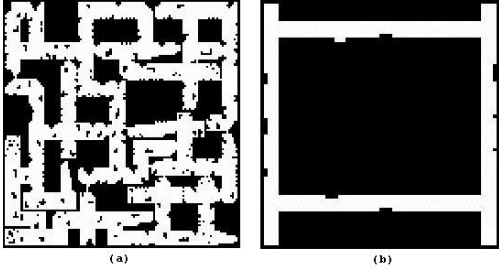


Fig. 3. Two worlds with differing world complexities.

itively, robot performance generally decreases as world complexity increases.

Some interaction schemes scale better to the effects of complexity than do others. An interaction scheme that scales well to complexity (i.e., robot performance changes little with changing world complexity) is said to be *complexity tolerant*. Any metric which claims to estimate robot performance must take into account world complexity.

D. Combining Neglect Tolerance and Interface Efficiency

The performance of a semi-autonomous robot declines as human attention is spent on other tasks and/or the complexity of the world increases. Additionally, effective human-robot interactions should allow performance levels to remain high. This implies that interactions must be frequent enough and last long enough to maintain sufficiently high robot performance levels. The combination of neglect tolerance and interface efficiency determine the frequency and duration of these interactions.

To illustrate this, consider Figure 4. In the figure, (moving from left to right along the horizontal axis), a robot begins at performance level zero (or from standstill). A human operator begins to interact with the robot (Task 1). When this occurs, performance is modeled as an interface efficiency curve (see Figure 2). When a human terminates the interaction and turns his/her attention to another task (Task 2), the robot's performance level begins to deteriorate and is modeled as a neglect tolerance curve (see Figure 1). Before the robot's performance level drops below acceptable levels, the human must again turn his/her attention to the robot (Task 1). Because of time for context switches, these interactions must begin some time before a robot's expected performance declines below an acceptable level. In this way, the average frequency and duration of human-robot interactions for an interaction scheme can be determined.

E. Mathematical Measures of Usefulness

Let π denote an *interaction scheme*; thus, π represents a particular interface and autonomy level pair. The performance of a robot employing interaction scheme π is defined by a random process indexed by time t , world complexity c , and the interaction rate r

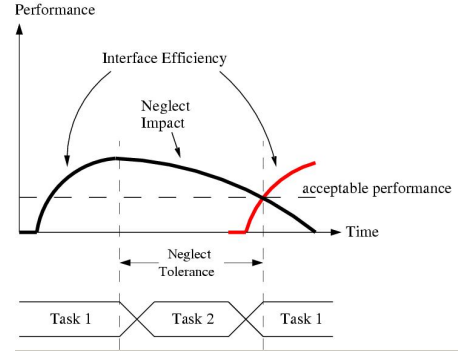


Fig. 4. The average frequency and duration of human-robot interactions can be determined by a combination of interface efficiency and neglect tolerance.

between human and robot ¹. A random process is a sequence of random variables indexed by time, and in this case, world complexity and interaction rate. More formally, the performance p of a robot for a given task is defined as

$$p = V(\pi; t, c, r) \quad (1)$$

where $c = \mathcal{C}(s)$ in which \mathcal{C} is a complexity metric (which we will explain later in this section) and s is a set of states.

Equation (1) uses the generic time term t . However, time is accessed differently by the neglect tolerance metric than it is by the interface efficiency metric. Time is accessed by the neglect tolerance metric as time-off-task t_{off} , which denotes the time elapsed since the robot was last serviced. The interface efficiency metric accesses time by time-on-task t_{on} , which denotes the time elapsed since servicing began. Thus, if the robot is currently being serviced, then t is t_{on} . If the robot is being neglected, then t is t_{off} . Therefore, equation (1) becomes

$$p = V(\pi; t, c, r) = \begin{cases} V_S(\pi; t_{\text{on}}, c, r) & \text{if being serviced} \\ V_N(\pi; t_{\text{off}}, c, r) & \text{otherwise} \end{cases} \quad (2)$$

where the variables are defined as before. Thus, $V_S(\pi)$ is a measure of the interface efficiency of π and $V_N(\pi)$ is a measure of the neglect tolerance of π .

From $V_S(\pi)$ and $V_N(\pi)$, the average frequency and duration of human-robot interactions for an interaction scheme can be determined by the technique illustrated in Figure 4. These average interactions are useful to for two purposes. First, they us to compare various interaction schemes. An interaction scheme which requires both low duration and low frequency of interactions is preferred to a interaction shceme which requires both high duration and high frequence of interactions. Second, they allow us to schedule the attention of an operator between multiple robots or multiple tasks.

¹In this work we simply allow each interaction to last as long as the human operator believes is necessary. Because of this assumption, we can classify an interaction rate with one variable r , which is the length of time in which a robot does not interact with a human.

F. Related Metrics

We mentioned in the introduction that measures of neglect tolerance and interface efficiency depend on two metrics. The first of these is an instantaneous performance metric. The second is a complexity metric.

F.1 Instantaneous Performance Metrics

In this paper, the term *performance metric*² is used to denote the *work* performed by a robot with respect to that robot's, or, perhaps, some other object's, *capacity* to perform work. Robot *performance* is simply the ratio $\frac{\text{work}}{\text{capacity}}$. Note that performance can be either positive or negative, and can take on any value in the range [-1, 1].

Continuous robot performance can sometimes be difficult to measure. In many instances, it is very easy to measure the performance of a robot after it has completed a task, but it is difficult to measure performance while the task is in progress. In this paper, however, we assume that performance can be measured or estimated continuously, and leave situations in which performance can not be measured or estimated continuously to future work.

The way in which performance is measured can be different for each task. The neglect tolerance and interface efficiency metrics require only that at any given time, an estimate of the *instantaneous performance*³ of the robot be available. This implies that we must be able to estimate *instantaneous work* and *instantaneous capacity* for work as well. Assuming we have these estimates, we have

$$ip_t = \frac{iw_t}{ic_t} \quad (3)$$

where ip_t is the instantaneous performance at time t , iw_t is the instantaneous work performed at time t and ic_t is the instantaneous capacity for work at time t .

As an example, consider the task of navigating a robot through a maze world towards a goal position. In this task, a robot's capacity is simply the speed at which it approaches its goal if it takes the optimal path at top speed. Thus, a robot's instantaneous performance is simply the rate at which it is actually approaching its goal divided by this capacity. This must be a value between -1 and 1, so it satisfies the conditions of an instantaneous performance metric.

F.2 Complexity Metrics

Like performance, complexity is also difficult to measure. Complexity is, in fact, somewhat subjective. A

²The actual performance metric should not be confused with the performance prediction which the interface efficiency and neglect tolerance metrics perform. The Interface efficiency and neglect tolerance metrics use an instantaneous performance metric to classify robot actions so that future robot performance can be predicted.

³We use the term instantaneous performance to indicate the performance of a robot over a small time interval.

world can be considered relatively simple or very complex⁴, depending on the task being performed. Additionally, to one set of abilities a world may be considered very complex, whereas to another set of abilities the same world may be considered quite simple.

This being said, complexity metrics are an important part of the neglect tolerance and interface efficiency metrics. We do not specify how complexity must be measured for all tasks, as such a specification would be impractical. We only say that an estimate of world complexity is required. How this is done is left to the system designer. Good complexity metrics, however, tend to assign high complexity estimates to environments which make a task difficult for a robot to perform, and low complexity estimates to environments which make a task easy to perform.

We consider, again, the task of navigating a robot through a maze world towards a goal position. The two dominant factors that make navigation difficult are the branching factor (number of intersections per area) of the robot's world and the amount of clutter (amount of obstacles per area) in the robot's world. The branching factor of the world can be estimated by calculating from robot sonar signatures the number of different paths the robot can take over a certain distance traveled. The clutter of the environment can be estimated by combining (a) directional entropy⁴, (b) change in velocity over time, and (c) change in sonar values over time. Branching factor estimates and clutter estimates can then be combined as a weighted sum to obtain a world complexity estimate between 0 and 1.

This complexity metric, although certainly not perfect, does a fairly good job of estimating world complexity for the experiments reported herein. As an example, Figure 3 shows two worlds used in the experiments described in this paper. Using results from a teleoperation interaction scheme, the world in Figure 3(a) had an average complexity of 0.373 and Figure 3(b) had an average complexity of 0.216. These numbers indicate that indeed this complexity metric returns a significantly higher value for a world that would be subjectively described as more complex.

Because of the ways in which a robot moves, this complexity estimate will tend to be slightly different for each interaction scheme. However, complexity estimates made by this complexity metric have shown to be similar for all the interaction schemes we have used for the navigation task.

IV. EVALUATION TECHNOLOGY

In the previous section, we discussed the random process $V(\pi)$ ⁵, which is a measure of the neglect tolerance and interface efficiency of the interaction scheme π . In this section, we discuss how this random process can

⁴Directional entropy is loosely defined as how often the robot changes direction over time. High entropy correlates well with complex environments and is computed using the techniques described in [3]

⁵We use the shorthand notation $V(\pi)$ to denote $V(\pi, t, c, r)$.

be estimated nonparametrically by designing and performing user experiments which sufficiently sample the domain space of the random process $V(\pi)$.

The domain of the performance random process includes time t , interaction rates r , and complexity c . As we discussed in the previous section, time t is separated into time-on-task t_{on} and time-off-task t_{off} . To sufficiently sample the time domain, we need users to spend time both servicing and neglecting a robot. To do this, we require that the user perform secondary tasks in addition to performing the primary task of servicing the robot. To sample the interaction rate domain thoroughly, we must vary how long the robot is neglected. This is achieved by varying the length of time that a user must perform a secondary task before returning to service the robot. The complexity domain can easily be sampled by simply performing the user experiments in worlds of various complexities.

Since the domain of the random process is continuous, it must be discretized so that it can be sampled sufficiently. Each data sample from the user study is placed in a bin defined by the discretized domain to form a nonparametric estimate of the random process $V(\pi)$.

Even after discretizing the domain of the random process, an impractical number of test subjects must be used in order to sufficiently sample the domain in this manner. This is because each complexity estimate is a sample from an unknown random variable. We overcome this problem by applying a gaussian filter to the data. Such an approach is justified by the central limit theorem. A large number of test subjects must still be used, but not nearly as many.

To summarize, the evaluation technology requires that humans and robots must actually interact in real systems to measure the neglect tolerance and interface efficiency of these systems. Secondary tasks must also be used to thoroughly sample the domain space of the random process.

V. EVALUATING THREE HUMAN-ROBOT SYSTEMS

We applied the evaluation technology described in the previous section to analyze the effectiveness of three different interaction schemes in performing the task of navigation through a maze world. In this section, we will describe the three interaction schemes and the user experiment used to estimate the neglect tolerance and interface efficiency of these systems. We will then show the results obtained from the user study..

A. Three Interaction Schemes

A snap shot of the GUI used by each interaction scheme is shown in Figure 5. A god’s eye view of the world (in the form of a topographical map) is shown in the center portion of the GUI. The sensory information of the robot is depicted graphically as well. Each of the autonomy modes uses a shared-control algorithm described in [5]. The robot takes a vector as input and combines this input with its sonar information to de-

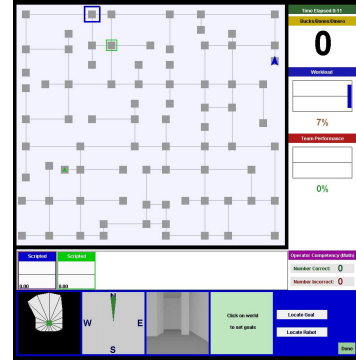


Fig. 5. The graphical user interface used in the user study.

termine, by using an algorithm which is a variant of potential fields, which direction to travel. The way in which the input vector is derived is what makes the autonomy mode for each interaction scheme different. A brief description of each of the three interaction schemes follows.

Teleop With this interaction scheme, the operator uses a joystick to control the robot. The robot uses this input as the input vector to the shared-control algorithm.

P2P With this interaction scheme, the operator tells the robot what to do at the next intersection (e.g., “turn right at the next intersection”). The operator uses a mouse to click buttons on the GUI to indicate what the robot should do next. The robot uses its sonars to determine if it is currently in an intersection or not. If it is not in an intersection, the input vector to the shared-control algorithm is simply a vector which points the robot straight ahead. If the robot believes that it is in an intersection and it has been told to turn right (or left), the input vector is simply a vector pointing 45° to the right (or left).

Scripted With this interaction scheme, the operator uses a mouse to drop a sequence of goal markers on the topographical map to lead the robot to its goal. The input vector is obtained by using the next goal marker in the sequence of goal markers it must traverse. The vector V_g between the goal marker and the robot is calculated. This vector is compared to the vector V_d , which points in the direction the robot is facing. If the angle between these vectors is greater than 45° , then the robot simply spins in place (in the direction which decreases the angle between the two vectors). If the angle is less than or equal to 45° , then the robot simply inputs V_g into the shared-control algorithm. If there is no goal marker placed, the robot stays in place.

B. A User Experiment

The user study was performed with simulated robots. The simulated robots were designed with a sixteen-sonar ring around the robot, a black and white camera image, and a compass. While the estimates of neglect tolerance and interface efficiency with simulated robots for the three interaction schemes do not apply directly to robots in the real world, they are sufficient to illus-

trate how the measurement technology is used. The use of simulated worlds also makes it easy to perform tests in a large variety of worlds.

The task to be performed in the experiment was the navigation task discussed earlier. The robot and its goal position were randomly assigned locations in a simulated world. The user was instructed to guide the robot, using the assigned interaction scheme, to the goal position. When the robot reached the goal position, another goal was randomly placed in the world for the robot to go to.

There were two secondary tasks performed by the operators in the user study. The first was to service a second robot. This made it possible to gather twice as much data per test session. The second secondary task was to perform two-digit addition and subtraction problems. This secondary task was performed when both robots in the system were being neglected.

The basic protocol followed in the experiments was to first train the test subject on the interaction scheme to be used in the next test session. When the operator felt comfortable with the interaction scheme, the training session was terminated and a test session began in one of twenty different worlds. In the test session, the operator first serviced one of the robots. When the operator was done servicing that robot he/she pushed a button on the GUI, after which the operator was assigned one of the secondary tasks. If it was time to service the other robot, interactions with that robot began. Otherwise, the operator was asked to do arithmetic problems until it was time to service the other robot. This process continued for ten minutes. The operator was asked to reach as many goals as possible as well as answer correctly as many arithmetic problems as possible during each ten-minute test session.

A slight variation was made to the above protocol when the assigned interaction scheme was *Teleop*. Since the performance of a robot employing *Teleop* quickly goes to zero when the robot is neglected, there was not very much incentive for the operator to ever neglect the robot. Thus, interactions between the operator and the robot being serviced were automatically terminated after ten seconds, after which the operator was assigned another task.

Each test subject took part in three ten-minute test sessions, using a total of two different interaction schemes. A total of forty test subjects were used in all, so 120 test sessions were performed. Of these sessions, 15 were dedicated to the *Teleop* interaction scheme, 48 to the *P2P* interaction scheme, and 57 to the *Scripted* interaction scheme.

As we mentioned previously, the domain space of the random process, consisting of the variables r , t , and c , must be properly discretized. In order for r to be sampled sufficiently for each interaction scheme, some neglect times, which are determined by a computer, must be extended until the expected performance of the robot approaches zero. This is a different length of time for each interaction scheme so r must be dis-

cretized differently for each interaction scheme. For *Teleop*, interaction rates took on only one value since robot performance immediately dropped to zero upon being neglected. For *P2P*, the interaction rate domain was divided into bins of 5, 10, 15, 20, 25 and 30 seconds. For *Scripted*, the interaction rate domain was divided into bins of 10, 20, 30, 40, 50, and 60 seconds. The time (t) dimension of the domain space was discretized into half second increments and the complexity (c) dimension of the domain space was discretized into chunks of 0.05 units.

The instantaneous performance and complexity metrics described in section 3 were used to estimate the instantaneous performance of the robot and the complexity of its world. These estimates along with time, operator actions (such as mouse clicks and joystick movements), and robot state information were logged for use in computing the random processes for each interaction scheme.

C. Results

Figure 6 shows the mean of the random processes $V(Teleop)$, $V(P2P; r = 30sec.)$, and $V(Scripted; r = 60sec.)$. The trends of the graphs reflect the trends we hypothesized earlier in this paper. As complexity increases, performance decreases. Additionally, as a robot is neglected, performance decreases. This is true for each interaction scheme, although at varying degrees. The mean of the random processes also illustrates the neglect tolerance and interface efficiency of each of the interaction schemes.

Figure 7 shows the expected performance of a robot using each of the three interaction schemes in an environment with complexity 0.35. Figure 7(left) shows the interface efficiency of the interaction schemes. As can be seen, the *Teleop* interface is the most efficient at bringing the robot from low performance levels to high performance levels, as it takes only a few seconds for it to do so. The other two interaction schemes take about ten seconds longer to reach peak expected performance levels than does *Teleop*. Figure 7(right) shows the neglect tolerance of the three interaction schemes. It is obvious from this graph, as well as from Figure 6, that *Scripted* has a much higher tolerance to neglect than does *Teleop* and *P2P*, as expected performance levels decay much slower as the robot is neglected for increasing amounts of time.

Given $V(Teleop)$, $V(P2P)$, and $V(Scripted)$, we can estimate average interactions required by the interaction schemes by setting a minimum performance level as shown in Figure 4. These results are shown in Figure 8 for most world complexity levels. The minimum acceptable performance level used to obtain these interactions was 50% of peak expected performance levels. As can be seen from the figure, the *Scripted* interaction scheme requires less frequent interactions than do the other interaction schemes. Additionally, for most levels of complexity, the average interaction time required by *Scripted* is less than that required by *P2P*.

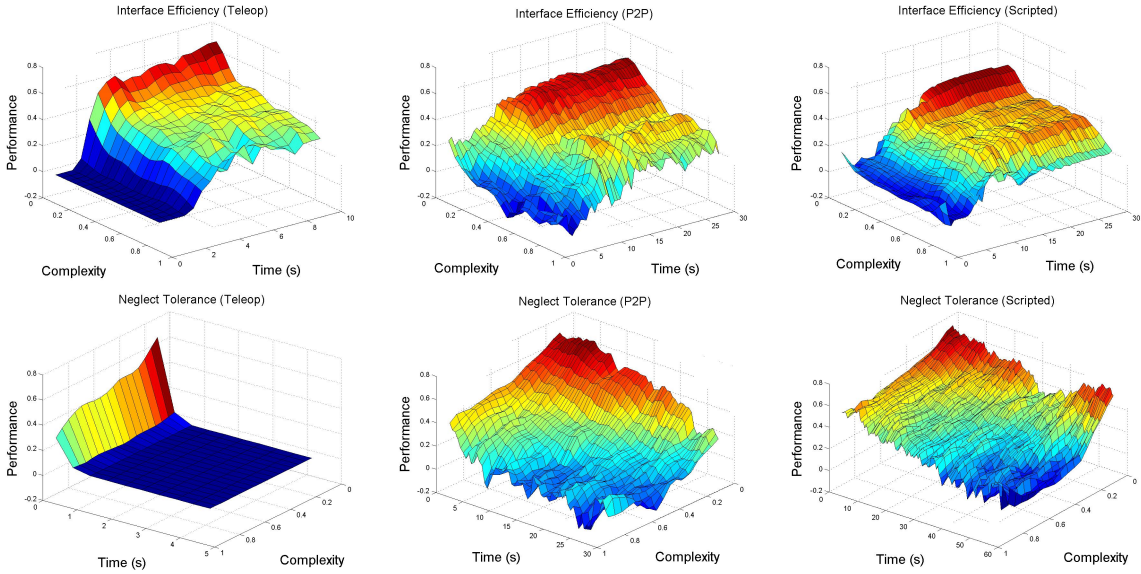


Fig. 6. Plots of the mean of the random processes $V(\text{Teleop})$, $V(\text{P2P}; r = 30\text{sec.})$, and $V(\text{Scripted}; r = 60\text{sec.})$.

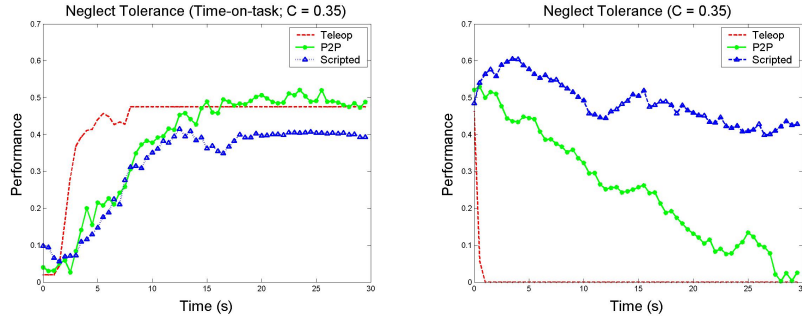


Fig. 7. Shows the measures of interface efficiency (left) and neglect tolerance (right) with world complexity 0.35 for *Teleop*, *P2P* and *Scripted*.

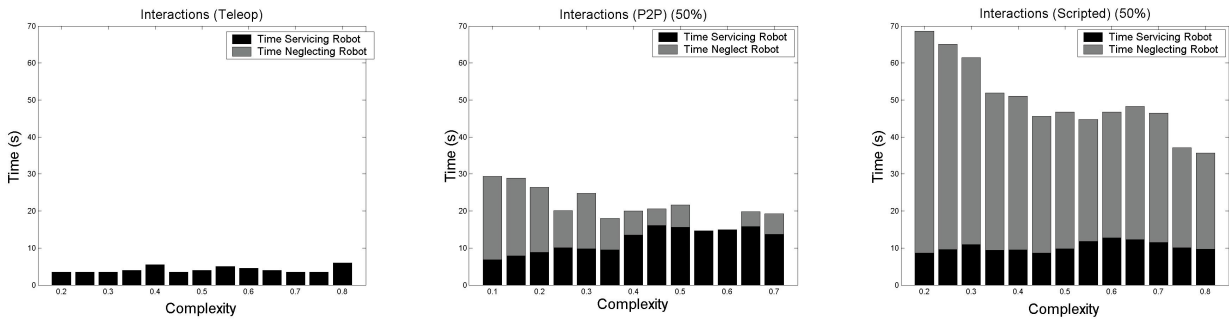


Fig. 8. Shows the average interactions which should take place (based off a minimum acceptable performance level of 50% of peak values) for the three interaction schemes.

Thus, human-robot interactions with *Scripted* require less operator workload than do the other two interaction schemes.

The frequency and duration of interactions, encoded as time-to-task and time-off-task in Figure 8, define the operator workload of an interaction scheme. Figure 9 shows this operator workload (shown as $\frac{\text{time-on-task}}{\text{totaltime}}$) plotted against the average expected performance of the interaction scheme when such interactions are followed. Plots are shown for three different levels of world complexity. In general, as complexity increases, points

tend towards the bottom-right corner of the plots (from the top-left corner). An interaction scheme's complexity tolerance is shown by how slowly it approaches the bottom-right corner as complexity increases. Note that *P2P* approaches the bottom-right corner faster than the other two interaction schemes. Thus, *Scripted* and *Teleop* are more complexity tolerant than is *P2P*.

Figure 9 also illustrates the tradeoff that occurs between operator workload and robot performance. Consider the results when world complexity is equal to 0.20 (at left). In this figure, *P2P* has a higher expected per-

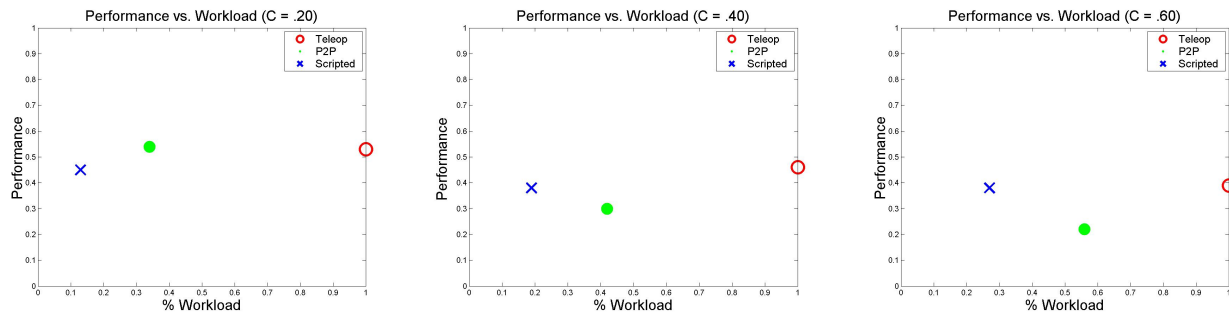


Fig. 9. Compares the interaction schemes in terms of % operator workload and robot performance for different levels of complexity c .

formance than does *Scripted*. However, this comes at the cost of increased operator workload. This tradeoff means that unless one interaction scheme completely dominates the other, the best interaction scheme to be used is dependent on the circumstances of the system.

To summarize, the *Scripted* interaction scheme has a higher tolerance to neglect than do the other interaction schemes. Since *Scripted* requires no more interaction times than does *P2P*, it is usually a more effective interaction scheme (in the simulator used in the use study) than is *P2P*. While *Teleop* has the most efficient interface efficiency of the three interaction schemes, it requires constant attention from the operator, and thus is not desirable for many situations. Additionally, for most complexity levels, the average performance of a *Scripted* robot is about the same as that of a *Teleop* robot.

VI. SUMMARY

Since most users want robots that will help them accomplish tasks, human-robot interactions are required. We want robots that interact effectively with humans and are capable of performing complex tasks with varying degrees of autonomy. In this paper, we discussed two components which determine the usefulness of a system: how much the robot can do autonomously and how much the robot supports human-robot interactions. We captured these components in the notions of *neglect tolerance* and *interface efficiency*, and developed metrics for them.

To estimate measures of neglect tolerance and interface efficiency, we described an evaluation technology. The evaluation technology requires the use of secondary task in user studies. We performed a user study using this evaluation technology to measure the interface efficiency and neglect tolerance of three human-robot systems. These measures allowed us to compare the three systems.

Although the metrics described in the paper are powerful for the analysis of interaction schemes, the user studies can be very time consuming and sometimes impractical. Thus, finding more efficient methods for measuring the neglect tolerance and interface efficiency of human-robot systems is needed.

REFERENCES

- [1] K. S. Ali and R. C. Arkin. Multiagent teleautonomous behavioral control. In *Machine Intelligence and Robotic Control*, volume 1, no. 2, pages 3–10, 2000.
- [2] L. Bainbridge. Ironies of automation. In *Automatica*, volume 19, pages 775–779, 1983.
- [3] E. R. Boer, O. Nakayama, T. Futami, and T. Nakamura. Development of a steering entropy method for evaluating driver workload. In *International Congress and Exposition*, 1999.
- [4] L. Conway, R. A. Volz, and M. W. Walker. Teleautonomous systems: Projecting and coordinating intelligent action at a distance. In *IEEE Transactions on Robotics and Automation* 6(2), 1990.
- [5] J. W. Crandall and M. A. Goodrich. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '02)*, 2002.
- [6] G. A. Dorais, R. P. Bonasso, D. Korenkamp, B. Pell, and D. Schreckenghost. Adjustable autonomy for human-centered autonomous systems on mars. In *Proceedings of the First International Conference of the Mars Society*, August 1998.
- [7] T. Fong, C. Thorpe, and C. Baur. A safeguarded teleoperation controller. In *IEEE International Conference on Advanced Robotics (ICAR)*, 2001.
- [8] K. Goldberg, B. Chen, R. Solomon, and S. Bui. Collaborative teleoperation via the internet. In *IEEE International Conference on Robotics and Automation*, April 2000.
- [9] M. A. Goodrich and E. R. Boer. Model-based human-centered task automation: A case study in acc system design. In *IEEE Transactions on Intelligent Transportation Systems* 1(1), March 2000.
- [10] A. Kheddar, P. Coiffet, T. Kotoku, and K. Tanie. Multi-robots teleoperation-analysis and prognosis. In *6th IEEE International Workshop on Robot and Human Communication*, pages 166–177, 1997.
- [11] D. Korenkamp, R. P. Bonasso, D. Ryan, and D. Schreckenghost. Traded control with autonomous robots as mixed initiative interaction. In *AAAI Spring Symposium on Mixed Initiative Interaction*, March 1997.
- [12] M. Tambe P. Scerri, D. V. Pynadath. Towards a theory of adjustable autonomy. In *Journal of Artificial Intelligence Research*, volume 17, pages 171–228, 2002.
- [13] R. Parasuraman, T. B. Sheridan, and C. D. Wickens. A model for types and levels of human interaction with automation. In *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, volume 30, no. 3, pages 286–297, May 2000.
- [14] D. Perzanowski, A. C. Schultz, W. Adams, and E. Marsh. Goal tracking in a natural language interface: Towards achieving adjustable autonomy. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '99)*, pages 208–213. IEEE Press, 1999.
- [15] T. Rofer and A. Lankenau. Ensuring safe obstacle avoidance in a shared-control system. In *Proceedings of the 7th International Conference on Emergent Technologies and Factory Automation*, pages 1405–1414, 1999.
- [16] T. B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. The MIT Press, 1992.