Validating Human–Robot Interaction Schemes in Multitasking Environments

Jacob W. Crandall, Michael A. Goodrich, *Senior Member, IEEE*, Dan R. Olsen, Jr., and Curtis W. Nielsen

***Abstract*—The ability of robots to autonomously perform tasks is increasing. More autonomy in robots means that the human managing the robot may have available free time. It is desirable to use this free time productively, and a current trend is to use this available free time to manage multiple robots. We present the notion of neglect tolerance as a means for determining how robot autonomy and interface design determine how free time can be used to support multitasking, in general, and multirobot teams, in particular. We use neglect tolerance to 1) identify the maximum number of robots that can be managed; 2) identify feasible configurations of multirobot teams; and 3) predict performance of multirobot teams under certain independence assumptions. We present a measurement methodology, based on a secondary task paradigm, for obtaining neglect tolerance values that allow a human to balance workload with robot performance.**

***Index Terms*—Human–robot interaction, neglect tolerance, interface efficiency, neglect impact.**

I. Introduction

RECENTLY, there has been much discussion in the robotics community on creating robot systems that allow a single human to perform multiple tasks, especially managing multiple robots. The possibility for such one-to-many human–robot teams is caused by the ever-increasing autonomy of robots. As a robot becomes more autonomous, its human manager has more free time to do other tasks. What better way to use this free time than to have the human manage multiple robots or manage multiple tasks?

The potential impact of this line of reasoning includes some very desirable consequences, but there are some clear upper bounds on the number of robots and the number of tasks that a single human can manage. These upper bounds are created by how long a single robot can be neglected. Formally, *neglect time* is the expected amount of time that a robot can be ignored before its performance drops below a threshold.

During the time that a robot is being neglected, the human manager can conceivably be doing any other task. However, once the neglect time is exhausted, the human must interact with the robot again. The average amount of time required by the human to "retask" the robot once interaction begins is referred to as the interaction time. Formally, *interaction time* is the expected amount of time that a human must interact with a robot to bring it to peak performance.

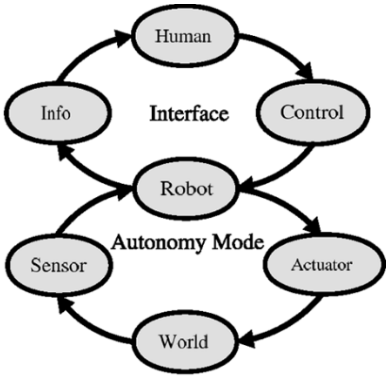


Fig. 1. Components of a human–robot system when the robot is remotely located.

The purpose of this paper is to demonstrate how estimates of neglect time and interaction time can be used to help a designer to create a system with multiple robots/tasks that a single human can manage. To do this, we will focus on multirobot teams, and demonstrate how neglect and interaction time estimates can be used to 1) identify whether a particular team configuration is feasible and 2) predict the performance of a team of robots managed by a single human. Furthermore, we will present an experimental approach for obtaining these estimates using secondary task studies, and show the application of this approach in both a simulation study and a study with real robots. The lessons learned from these multirobot studies apply to more general multitasking environments.

Throughout this paper, we find it useful to distinguish between various components of a human–robot system. We will restrict attention to human–robot interactions between a single human and one or more remote robots. The components of such a system are illustrated in Fig. 1, where the *robot autonomy* label refers to the artificial intelligence or computer control of a robot that allows it to act, for a time, without human interaction. This autonomy includes the mapping from sensor inputs to the robot actuators, possibly modulated by current or past human input(s). In the figure, the *interface* label refers to the software located at the human's location that allows the human to perceive the world and the state of the robot, and send instructions to the robot.

Together, the robot autonomy and the interface represent design variables that affect how long a robot can be neglected and the time required to interact with a robot. Because both robot autonomy and the interface dictate the human–robot interactions, they should be designed together. To emphasize the interrelationships between autonomy and interface, we refer to the pair as an *interaction scheme* which is formally defined as the ordered pair *interaction scheme = (autonomy, interface).*

Manuscript received July 31, 2004; revised March 14, 2005. This paper was recommended by the Guest Editors.

The authors are with the Computer Science Department, Brigham Young University, Provo, UT 84602 USA (e-mail: mike@cs.byu.edu).

Digital Object Identifier 10.1109/TSMCA.2005.850587

II. Related Literature

In this section, we briefly review related work in robot autonomy and interface design as it relates to human–robot interaction. We also briefly touch on human factors issues such as workload and situation awareness, and discuss implications of human–robot interaction for robot teams.

1) Autonomy Support for Human-Centered Robotics: In reviewing human–robot autonomy, we adopt the taxonomy presented in [1], which has four categories: 1) teleoperation; 2) shared control; 3) traded control; and 4) supervisory control. Current robot platforms typically emphasize teleoperation and supervisory control. Sheridan's book is still the seminal and most important treatment of supervisory control [2]. Supervisory control for autonomous vehicles has produced some remarkable demonstrations (e.g., [3]), but introduces many complicated human factors issues [4].

Although straightforward to implement, teleoperation also presents complicated human factors issues [5]–[7]. While research into telepresence, mixed-reality displays, etc., has advanced the state of the art, pure teleoperation is often difficult for human operators [8]. This difficulty is often the result of limited interface capabilities [9], [10] compounded by communications issues such as intermittency and delay [11].

Shared control is becoming an increasingly popular response to the challenges of designing good interfaces that are robust to communications lag. One approach to shared control that grants a great deal of human authority is called safeguarding [12], [13]. A second approach to shared control is when the robot combines the operator's instructions with its own assessment of the environment [14], [15]. The interaction schemes described in this paper employ a mix of these two approaches to shared control.

Pure traded control is common for problems in which the human and machine are colocated and where there is some physical or mental burden associated with human operation [16]. The point-to-point and scripted interactions described in this paper are traded control methods. Central to the issue of trading control are adjustable autonomy [17]–[19] and mixed initiatives [20], [21].

2) Interface Technologies: A sense of *telepresence* [2], workload [22], and situation awareness [23], [24] are frequently driving forces behind interface design. Human factors studies in interface design often point to a lack of situation awareness as a culprit in increasing workload and reducing interface efficiency [23], [25]. Techniques that tend to produce more efficient interfaces include sensor fusion [26], [27], adjustable displays [25], [28], and display-based information storage [29].

In a brief literature review, it is impossible to cite every example of an interface technology. However, it is useful to point to those examples of displays that seek to increase efficiency by making interaction "natural" in some sense. Examples of such displays include personal digital assistant (PDA) interfaces [30], gesture recognition [31], [32], emotive computing [33], and natural language-based interfaces [34]. It is also useful to point to examples that seek to provide a better sense of telepresence, such as virtual reality-based displays [35], predictive displays [36], and augmented virtuality displays [11], [37]. Additionally, there are other approaches to making interfaces more useful for humans, including intelligent interface assistants [38] and learning from human operators [39].

3) Teams, Task Switching, and Operator Workload: If an interface is efficient and the autonomy is tolerant to neglect, then the human interacting with a robot may have "free time." One way to use this free time is to have the human manage multiple robots. Often, multiple robots have a strong team component wherein the human manages the team dynamic rather than individual robots. The idea behind these systems is to allow the human to operate at a higher level of abstraction and therefore allow efficient team interactions. Examples of multirobot teams include work done in the Georgia Tech. *MissionLab*, swarms [40], and work in which multiple robots are teleoperated by a single user [41], [42]. Additionally, there are many problems (see, for example, [40]) for which many small, simple, and cheap agents can accomplish tasks more efficiently than a few big, complex, and expensive agents. Seminal work on behavior-based cooperative robots includes the framework of [43]. A hybrid approach to distributed cooperation among heterogeneous robots that combines elements of behavior-based robotics and higher-level reasoning is presented in [44].

However, team management is not the only possible way to use free time. There are a multitude of potential tasks that can be done during free intervals, including managing independent robots. However, managing multiple tasks with available free time introduces other human factors issues. These issues include the cost of switching between tasks [45], [46] and the design of interfaces to support switching [47].

Issues of neglect tolerance and interface efficiency are not new, although the treatment presented herein is original. Workload issues in multitasking settings have been studied in general forms such as *performance resource function* curves [28] in which performance on a task is measured as a function of resources demanded by the task. They also call to mind *attention operating characteristic* plots [48] that cross plot performance on two independent tasks as a function of attention spent with each task.

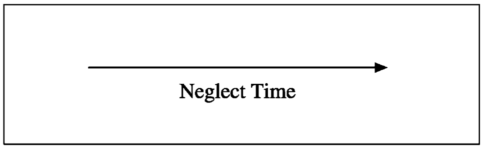
III. Fanout and Feasibility

Prior to presenting a formal definition of neglect tolerance and an experimental approach for estimating tolerance, we present some practical uses of these estimates. We begin by presenting an upper bound on the number of independent,[1] homogeneous robots that can be managed by a single human. We then discuss how the principles of fanout can be used to determine whether it is feasible for a human to manage a team of heterogeneous robots.

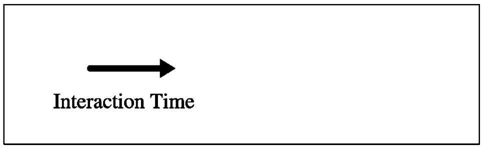
In a problem with multiple robots, neglect time and interaction time dictate the maximum number of robots that a single human can manage. The upper bound on the number of robots can easily be computed when all robots are homogeneous and

[1] Robots are considered independent if one robot cannot perform another robot's job, and if the performance of one robot does not depend on the performance of another.

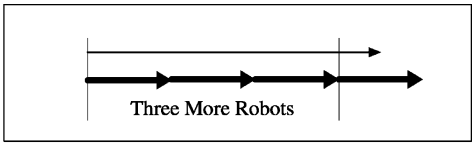
independent. To illustrate, suppose that the neglect time for each robot is represented as a time vector as follows.



Suppose further that the interaction time for each robot is represented as a different time vector as follows.



The maximum number of robots that can be managed by a single human is given by the number of interactions that can be "fit" into a single time.



As illustrated in the figure, because only three interaction times can fit into a single neglect time, the human can manage no more than three additional robots. This means that the human can only manage four total robots. Note that the figure illustrates how trying to include one more robot into the team causes the cumulative interaction times to exceed the available neglect time.

The idea of determining how many independent homogeneous robots can be managed by a single human is captured by the notion of fanout [49]. Roughly speaking, fanout is one plus the ratio of neglect time to interaction time. The ratio represents the number of other robots that the human can manage during the neglect time interval, and the "plus one" represents the original robot. Thus,

$$\text{Fanout} = \frac{\text{NT}}{\text{IT}} + 1 = \frac{\text{NT} + \text{IT}}{\text{IT}}$$

where NT and IT represent neglect time and interaction time, respectively. One contribution of this paper is to present an approach to obtaining NT and IT values for estimating fanout that uses experiments from only a single robot. This complements the work in [50] and [51], where fanout is estimated using large simulation studies involving several sizes of robot teams.

This idea can be extended to teams of heterogeneous robots performing independent tasks. When a team is made up of heterogeneous robots, then each robot has its autonomy level and interface. This, in turn, implies that each robot has a given neglect time and interaction time. Let $N_i = (\text{NT}_i, \text{IT}_i)$ denote the neglect and interaction time of robot i. A team of M robots consists of the set $\mathcal{T} = \{N_i : i = 1 \ldots M\}$.

To determine whether a human can manage a team of robots $\mathcal{T}$, we can use the neglect times and interaction times to determine if a team is infeasible.

$$\mathcal{T} \text{ is} \begin{cases} \text{feasible}, & \text{if } \forall i\ \text{NT}_i \geq \sum_{j \neq i} \text{IT}_j \\ \text{infeasible}, & \text{otherwise} \end{cases}. \tag{1}$$

The idea is to find out whether the neglect time for each robot is sufficiently long to allow the human to interact with every robot in the team.

We now turn attention to formally defining the notion of *neglect tolerance* and establishing a methodology for evaluating neglect tolerance. Our definition allows us to compute NT and IT, and allows us to evaluate the tradeoff between NT, IT, and robot performance.

IV. Neglect Tolerance

One contribution of this paper is to create a methodology for determining neglect time and interaction time. This methodology is built on the intuition that the likely performance of a robot degrades as the human ignores the robot and as world complexity increases. Additionally, the methodology relies on the intuition that a robot performing at less than peak performance will likely improve its performance over time as a human interacts with it.

The thesis statement of this paper is: *Human–robot interactions should be frequent enough, last long enough, and be efficient enough for the robot to maintain acceptable performance levels without placing undo burdens on a human operator*. In essence, we are trying to create a design and evaluation methodology that breaks up time into discrete quanta of alternating neglect and interaction times. Such *quanta* allow us to determine fanout of a team of homogeneous robots and to determine feasibility of a team of heterogeneous robots. Additionally, we want to characterize how these quanta correspond to the expected performance of the team in such a way that we can compare possible team configurations to select the best one.

To accomplish this, we use secondary task studies to create curves that characterize 1) the way neglect affects the expected performance of a robot acting autonomously and 2) the way human interaction affects the expected performance of a robot after being neglected for a period of time. These characteristic curves are referred to as the *neglect impact* and *interface efficiency* curves, respectively. The curves represent robot performance as a function of time spent working with the robot, denoted by t_{on}, and time spent doing other tasks, denoted by t_{off}

Let J denote the expected performance of an interaction scheme $\pi = (\text{Autonomy}, \text{Interface})$ on a specific task $\mathcal{T}$. This performance changes as a function of time depending on whether the human is servicing or neglecting the robot. Thus, we introduce the notation $J_N^{\mathcal{T}}$ to denote the performance of the robot (on task $\mathcal{T}$) while it is being neglected, and $J_S^{\mathcal{T}}$ to denote the performance of the robot (on task $\mathcal{T}$), while it is being

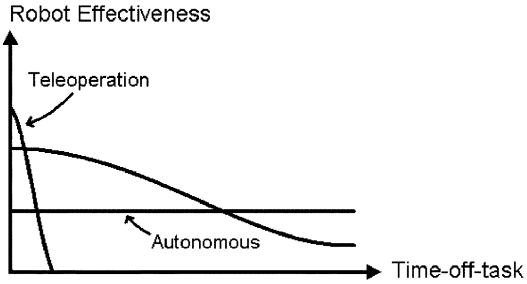


Fig. 2. Impact of neglect (time-off-task) for a world of constant complexity.

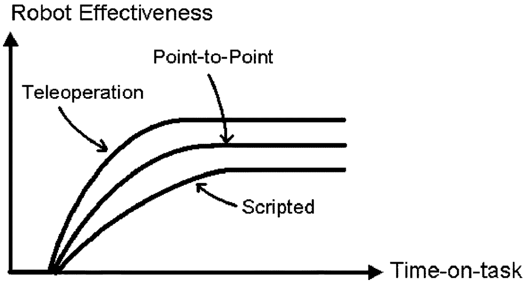


Fig. 3. Qualitative representations of interface efficiency for various presentations of information.

serviced.[2] Thus, J_N and J_S are the notational representations of the neglect impact and interface efficiency curves, respectively. The shapes of these curves are a function of time and world complexity. Thus, we are interested in $J_S(\pi, C, t_{\text{on}}, t_N)$, and $J_N(\pi, C, t_{\text{off}})$, where C denotes a measure of world complexity, t_{on} is the elapsed time since attention was turned to the robot, t_N is the amount of time the robot was previously neglected, and t_{off} is the elapsed time since attention was turned away from the robot.

1) Neglect Impact Curves: Fig. 2 illustrates how expected robot performance deteriorates with time-off-task. The nearly vertical curve represents a teleoperation 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, given the current state of the art, 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.

2) Interface Efficiency Curves: Fig. 3 illustrates how expected robot performance may change when the human interacts with the robots. Conceptual interface efficiency curves are shown for teleoperation, wherein performance rapidly climbs after a delay period during which the human gains awareness of the robot's state, point-to-point autonomy (described in Section V) wherein the human gives a single directive to the robot which the robot begins to execute, and scripted autonomy wherein the human gives a series of points for the robot to follow.[3]

3) Complexity: In addition to the influence of time-off-task, t_{off}, the expected performance $J(\pi)$ of a particular interaction scheme π is also affected by how the world responds to robot actions. Interaction schemes that are designed for a particular level of environmental complexity may not perform well for other environment complexities. To illustrate how world complexity, denoted C, can impact performance, consider worlds with variable branches and 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.

4) Neglect Tolerance: From these discussions, we can identify the parameters that determine the performance, denoted by J, of a robot:

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

Fig. 4 shows how neglect-impact and interface-efficiency determine neglect tolerance. (For simplicity, curves are illustrated for a fixed level of complexity.) Thus, when we combine the influences of neglect impact and interface efficiency, we derive a *neglect tolerance* relationship.

Beginning at the left, the operator starts managing a robot "from scratch," meaning that the operator brings the task from zero performance to desired performance; the efficiency of this process is represented by the interface efficiency curve. The operator then turns attention to a secondary task, and the impact of this neglect starts to affect performance. This is represented by the neglect impact curve. Eventually, the operator must again attend to the task or else performance declines below the acceptable performance threshold. Before this curve drops below the acceptable level, the operator again starts interacting with the robot. The interface efficiency curve for this portion of time is slightly different than the interface efficiency curve at the beginning of the scenario. Rather than beginning at zero performance, the robot has some level of performance that continues to decline during the "switching interval" while the operator gains awareness of the robot's state. Once awareness is gained, the performance of the robot begins to increase again. (Note that the interface efficiency curve representing performance increase "from scratch" is shown below this curve as a reference point.)

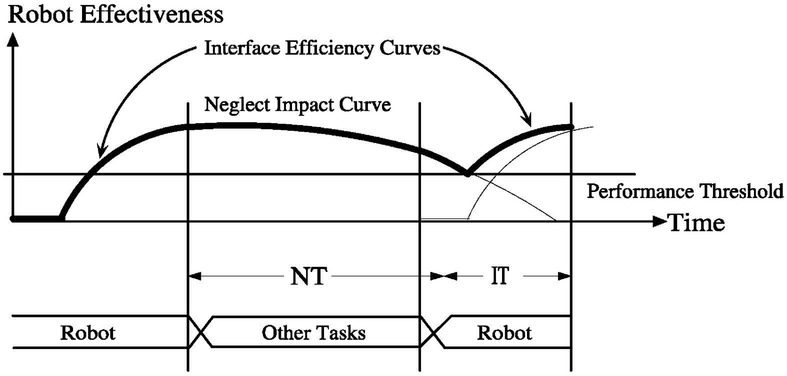
The time between when the operator begins to neglect the robot and when attention is turned back to the robot and performance again climbs is referred to as neglect time and is denoted NT. The expected time to service the robot is referred to as the interaction time and is denoted IT.

In Fig. 4, we have illustrated a switch cost effect that indicates that it takes a human a little time to "come up to speed" when they turn their attention back to the robot. During this switching interval, the robot performance may continue to decline so the switch back to the robot must begin soon enough to keep the performance above threshold.

Neglect tolerance is determined by how often interactions must occur to maintain a level of performance (see Fig. 4). Neglect tolerance is determined by two variables: time-on-task (which is a function of interface efficiency) and time-off task

[2] To simplify notation, we omit the variable $\mathcal{T}$ in the remainder of the paper. However, we emphasize that all measures of J_S and J_N are task dependent.

[3] Human interactions may actually decrease robot performance. Interface efficiency identifies when this occurs.



Summary of Terms

NT	Neglect Time
IT	Interaction Time
Tolerance	(NT,IT)
Neglect Impact	Decrease in robot performance in the absence of human input
Interface Efficiency	Increase in robot performance in the presence of human input
Interaction Scheme	(Autonomy,Interface)

Fig. 4. Neglect impact and interface efficiency curves determine neglect tolerance. The actual robot performance is given by the thick lines. (The average robot performance is given by the average of these thick lines over the neglect and second interaction cycle.)

(which is a function of vehicle autonomy level). To prevent an operator's performance on a particular task from dropping below an acceptable level, the task can only be neglected for a certain period of time. When the performance threshold is selected, time-on-task is discretized into an interaction time quantum, and time-off-task is discretized into a neglect time quantum. These quanta can then be used to determine whether a particular team configuration is feasible.

We define neglect tolerance as the pair $(\mathrm{NT}, \mathrm{IT})$ because both values determine how the robot can be used in a multitasking environment. Note how selecting the performance threshold determines the neglect time and the interaction time. Note further that whereas the neglect impact and interface efficiency curves represent the impact that human neglect and human service have on the robot, selecting the neglect and interaction times determine the performance of the robot averaged over time. In a multitasking situation, the robot's performance naturally fluctuates as the robot is alternately serviced and neglected. If the world is stationary, then the *average robot performance* over all time is equal to the robot's average performance over one neglect-service period.

A high level of average robot performance requires the user to select a high performance threshold, which means that neglect time must be short. A lower level of average robot performance permits a lower performance threshold, which means that neglect time may by longer, but that interaction time may also be longer. Thus, there is a tradeoff between average robot performance and $(\mathrm{NT}, \mathrm{IT})$.

V. Estimating Neglect Tolerance: User Study

In this section, we present the results of a user study involving 40 test subjects operating a simulated robot. We report the neglect impact and interface efficiency of three interaction schemes for a navigation task.[4] These measures help identify the strengths and weaknesses of each interaction scheme. This study accomplishes two purposes in this paper. First, it illustrates the use of the evaluation methodology. Second, it demonstrates how neglect impact and interface efficiency curves can be used to identify which interaction schemes are most appropriate for a given set of problem conditions including world complexity, human free time, and required minimum robot performance.

We use secondary tasks to create the neglect impact and interface efficiency curves. The rate at which the secondary tasks are presented to the human are selected so that they force the subject to neglect the robot for a predetermined amount of time. This allows us to estimate how much a robot's performance declines as a function of how long the subjects's attention has been turned to the secondary task. Thus, the secondary task structure allows us to sample the curves.

Additionally, the secondary task structure allows us to estimate interaction times. In a multitasking environment, cognitive resources must be shared or shifted between the multiple tasks. The most important of these resources are attention, short-term memory, and working memory. Secondary tasks allow us to estimate how long it takes for subjects to bring a robot back up to peak performance after it has been neglected and after the subject's cognitive resources have been dedicated to something other than the robot for a period of time. In this paper, we do not address how different secondary tasks change interaction time. Separate research in this area can be found in [52].

A. Interaction Schemes Summary

In this section, we compare the three interaction schemes according to their neglect tolerance characteristics. The three interaction schemes are: *Teleop*, *P2P*, and *Scripted*. We briefly summarize these schemes, but refer the reader to [53] for a more complete description.

1) *Teleop*: A shared teleoperation interaction scheme wherein the human gives directions via a joystick and the robot follows these directions while avoiding obstacles.
2) *P2P*: A point-to-point (*P2P*) interaction scheme wherein the operator instructs the robot as to what it should do when it reaches the next intersection/decision point (e.g., turn left at the next intersection). Additionally, the operator may give more low-level commands, such as instruct the robot to spin in place.
3) *Scripted*: A scripted interaction scheme wherein a series of waypoints are given to the robot, and the robot attempts to navigate to these waypoints while avoiding obstacles.

[4] Note that these measures are specific to the class of operators used in the user study. Thus, measures of interface efficiency and neglect tolerance should be obtained using operators with the same skill set as those that actually use the systems.

B. Instantaneous Performance Metric

To create the neglect impact and interface efficiency curves, it is necessary to have an estimate of *instantaneous performance*. Instantaneous performance ip is defined as instantaneous work iw divided by instantaneous capacity for work ic. For the navigation of a robot through a maze world toward a goal position, the instantaneous capacity of the robot is the work that the robot would do if it moved optimally (ignoring clutter) toward its goal at top speed. Since the simulated robots can travel at the rate of 30 in/s, we define their instantaneous capacity for work as

$$ic = 30t_\epsilon \qquad (2)$$

where t_ϵ is the time elapsed, usually a small amount of time. The instantaneous work done by a robot in this task is how much closer it is to its goal after time t_ϵ. Let d_i be the distance the robot is from its goal at time i. Then, the instantaneous work iw performed at time i is

$$iw_i = \frac{d_i - d_{i-t_\epsilon}}{t_\epsilon}. \qquad (3)$$

Using Dijkstra's Algorithm, we can obtain d_i by calculating the shortest path from the robot to its goal at time i using the topographical map of the world, which contains information about the distance between nodes (i.e., intersections) in the world. Thus, by combining (2) and (3), the instantaneous performance of a robot at time i is

$$ip_i = \frac{iw_i}{ic} = \frac{d_i - d_{i-t_\epsilon}}{30t_\epsilon^2} \qquad (4)$$

where the variables are defined as before.

We note that since "cutting corners" can perhaps decrease the shortest path to the goal and we do not want to worry about calculating the optimal way to cut corners for our distance measure, it is possible for ip_i to be greater than 1 (or less than -1). In such a case, ip_i is truncated to 1 (or -1). Thus, the sum of all ip over a period of time is not necessarily equal to the performance of the robot over that same period of time. It is, however, close to the overall performance, which makes it an acceptable performance metric.

C. Estimating World Complexity

As we mentioned previously, interface efficiency and neglect impact curves require estimates of the world's complexity. For this user study, world complexity was dynamically calculated by estimating the branching factor and clutter of the robot's environment using the robot's sensor information. The branching factor was estimated using the robot's sonar signatures and clutter was estimated using the robot's directional entropy, changes in its sonar values, and changes in its velocity. These estimates were then combined into a single number (between 0 and 1) representing world complexity. Details on these methods can be found in [53].

D. User Study Design and Protocol

In this user study, we used two secondary tasks. The first secondary task that the human operator was asked to perform was to control a second robot. This made it possible to gather twice as much data during each test session, so fewer test subjects were needed. The other secondary task was to perform two-digit addition and subtraction problems. The part of the GUI that displays robot sensory information was replaced by a math display when this secondary task was to be performed.

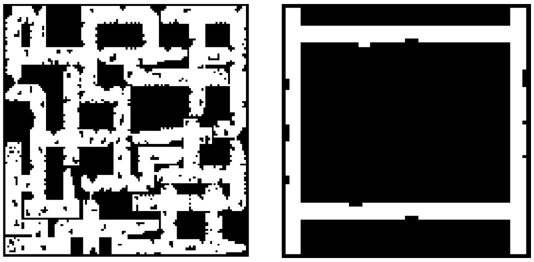


Fig. 5. Two of the simulated worlds used in the user study.

The operator was allowed to service that robot as long as he/she desired. When he/she was done, he/she clicked a button and was given a different task. A random neglect time[5] was then assigned to this robot and the operator was not allowed to service this robot again until the neglect time had elapsed. After the neglect time had elapsed, the task of servicing this robot was reassigned to the operator provided that the operator was not currently servicing the other robot. When both of the robots were being neglected, the operator was given the arithmetic task until it was time to service a robot again.

We created 21 different worlds of different makeup and complexity (two of which are shown in Fig. 5). Each of the 21 worlds had different branching factor and clutter. The first world, the *training world*, was used to train test subjects on the interaction schemes they were to use. This world included a wide variety of world complexities. The other 20 worlds were selected for use randomly during test sessions, but restrictions were made on how many times a world could be used.

Instructions on how each test subject was to proceed with the experiment was read from a prepared script. The experiment consisted of a series of training and testing sessions, counterbalanced to mitigate the effects of learning. 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 (volunteer undergraduate computer science and engineering students with no prior experience driving the robots) 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 schemes, and 57 to the *Scripted* interaction scheme.[6]

E. Results

The neglect impact and interface efficiency curves obtained in the user study for all world complexities are given in [53]. In this paper, we show the neglect impact and interface efficiency

[5] We used different ranges of neglect times for each interaction scheme (since they are each impacted by neglect differently). For *Scripted*, we used neglect times of 10, 20, 30, 40, 50, and 60 s. For *P2P*, we used 5, 10, 15, 20, 25, and 30 s. For *Teleop*, we used 10 s.

[6] Numbers of sessions were chosen to adequately sample the domain space of J_S and J_N for each interaction scheme.

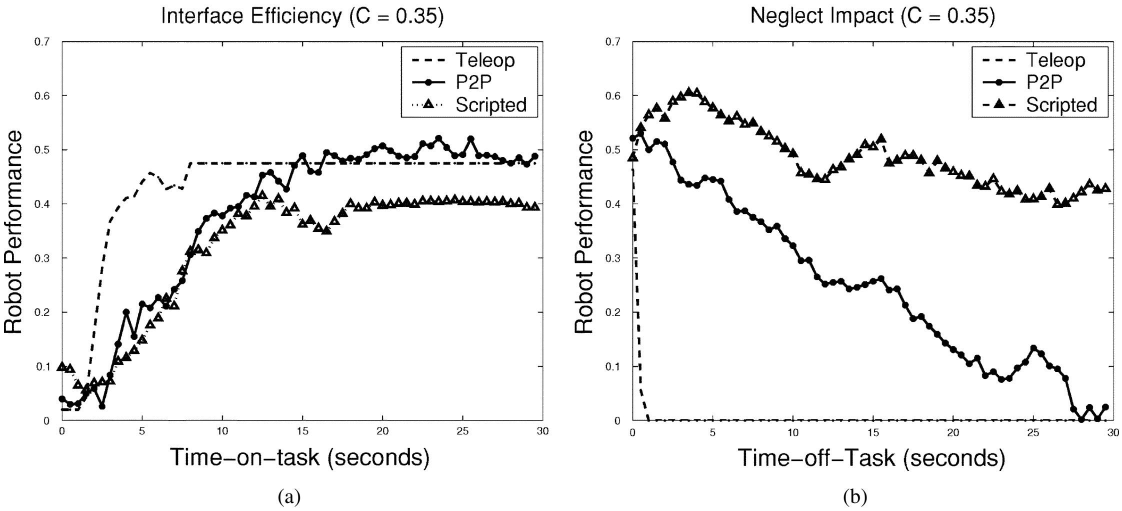


Fig. 6. Plots comparing the expected performance levels of robots employing the three interaction schemes at a world complexity of 0.35. (a) Expected robot performance levels during interactions (after significant neglect times). (b) Expected robot performance when the robots are being neglected.

curves at world complexity 0.35 (these results are typical of the other results found in [53]). Fig. 6(a) compares the interface efficiency of the three interaction schemes at this complexity. It shows that the performance of a robot employing *Teleop* reaches peak expected performance levels much quicker than do robots employing the other two interaction schemes. The other two interaction schemes peak at about the same time. $P2P$ peaks at about the same level that the *Teleop* interaction scheme does. However, the *Scripted* interaction scheme peaks at lower levels. This result can be somewhat deceiving since this graph was created by assuming that an operator quit servicing a robot when the robot had reached peak performance levels. This assumption is false with the *Scripted* interaction scheme, as seen in Fig. 6(b), which shows expected performance level during times of neglect.

Fig. 6(a) illustrates a switching cost for each interaction scheme. The interface efficiency of an interaction scheme estimates the switching cost by the amount of time it takes for robot performance to begin to increase substantially. We note that this estimate may not always be accurate. We leave further study of switching costs to future work.

Fig. 6(b) shows the differences in neglect impact for the three interaction schemes. After 30 s of being neglected, a robot employing *Scripted* is still expected to be performing at about 40% of capacity, while robots employing the other two interaction schemes have already reached, or approached, zero. Thus, *Scripted* is more tolerant to neglect than are the other two interaction schemes.

Fig. 7 shows the interaction rates (NT, IT) (i.e., neglect tolerance) necessary to maintain an average robot performance of approximately 0.30 for world complexities 0.30 (left) and 0.50 (right) for the three interaction schemes. The values NT and IT in the figure are found using the technique demonstrated in Fig. 4.[7] By changing the performance threshold, (NT, IT) can be found such that average robot performance is 0.30. The figure shows that the human operator must devote nearly constant attention to a robot employing *Teleop* to maintain this average robot performance level. Also, a human must devote much more attention to a robot employing $P2P$ than to a robot employing *Scripted* to maintain the same average robot performance level.

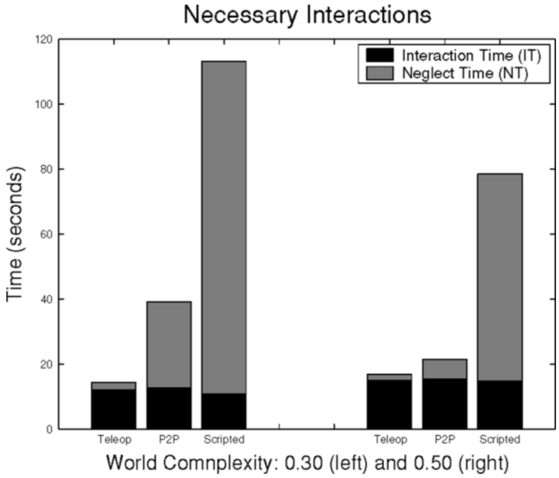


Fig. 7. Interactions necessary to maintain an average robot performance of approximately 0.30 at world complexities 0.3 (left) and 0.5 (right).

VI. Estimating Neglect Tolerance: Real-Robot Study

The results described in the previous section are true for simulated worlds. However, real world results could vary from these results, as noisy sensor readings and other phenomena change the nature of interaction schemes. However, the technique of how to identify the interface efficiency, neglect impact, and neglect tolerance of an interaction scheme is valid in the real world as well.

To demonstrate this, we estimated the interface efficiency and neglect impact for the P2P and *Scripted* interaction schemes using a Pioneer II robot (equipped with a camera, a laser-range finder and sonar) in a condemned building. Boxes were added to the environment to provide clutter as well as maze-like characteristics. Experiments took place in two different "worlds." The

[7] IT was determined to be the average interaction time used by the operators when a robot was neglect for time $t_N = \mathrm{NT}$.

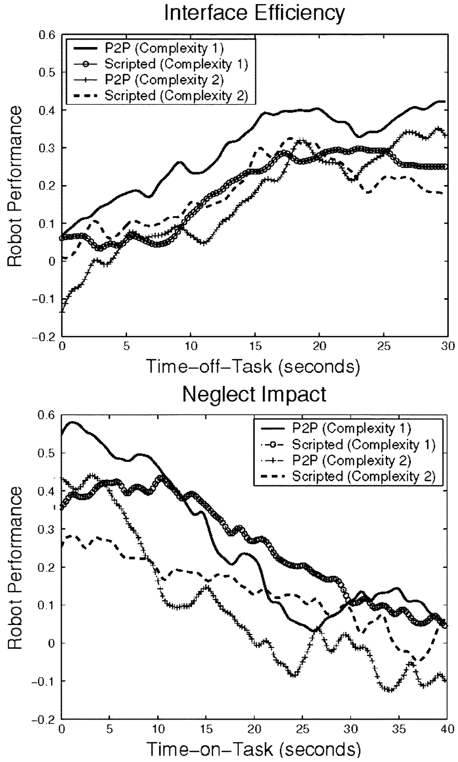


Fig. 8. Plots comparing the expected performance levels of robots employing $P2P$ and *Scripted* for two different world complexities in the real world. Interface efficiency is shown above and neglect impact below.

first world had very little complexity (i.e., low branching factor and clutter) and the second was more complex (more branching factor and clutter). To simplify the results, rather than calculating world complexity using the robot's sensor information as before, complexity was chunked into two categories: complexity 1 for the first world and complexity 2 for the second world.

These real world user studies used the same secondary task user study protocol as before, only users played tetris as a secondary task rather than performing arithmetic problems and interacting with a second robot. Eight users, college students with previous experience driving the robot using a similar interface, were used in the study. The mean of the interface efficiency and neglect impact curves is shown in Fig. 8.

Fig. 8 shows that *P2P* allows robot performance to reach higher levels in the real world then *Scripted*, but decays much quicker with neglect. Thus, selecting which of the two interaction schemes is appropriate depends on the needs of the multitasking system. If high robot performance is desired, then *P2P* should be selected, but at the cost of higher operated workload. If low operator workload is needed, then *Scripted* should be selected, at the expense of lower robot performance.

While *Scripted* dominated *P2P* in the simulator, it does not completely dominate *P2P* in the real world. This difference can be traced largely to the robot localization problem. In the real world, our robot localization was not precise, meaning that waypoints dropped on a map of the world were not always translated correctly into the robots reference frame. However, even with this difficulty, *Scripted* demonstrated more robustness to neglect in the real world then did *P2P*.

VII. Predicting Heterogeneous Team Performance

In order to predict the performance of a multirobot team, we first measure the neglect tolerance of two control schemes: *P2P*, which was described previously, and *region-of-interest* (ROI) in an exploration and goal-finding experiment. The ROI interaction scheme uses automated path planning and robot exploration to generate a series of waypoints that the robot follows using the *Scripted* interaction scheme described previously.

After measuring the neglect tolerance characteristics, we present a method for combining the expected performance of individual interaction schemes to predict the performance of multirobot systems. We use this performance-prediction algorithm to predict the performance of a three-robot system where a user guides the robots via various combinations of the two interaction schemes. We then compare the predicted performances with the actual performances of teams consisting of a human managing three robots.

A. Team Performance

Let π_i be an interaction scheme employed by robot i, and let $N_i(\pi_i) = (\mathrm{NT}_i, \mathrm{IT}_i)$ denote the neglect characteristics (made up of neglect time and interaction time) associated with a preselected performance threshold. Suppose that we have M robots. Let $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_M)$ denote the vector of interaction schemes, and let

$$\mathcal{N}(\boldsymbol{\pi}) = \big(N_1(\pi_1), N_2(\pi_2), \ldots, N_M(\pi_M)\big)$$

denote the vector of the neglect and interaction times for the team of robots for a given selection of interaction schemes.

This formalism allows us to not only talk about a team of M robots, but to also consider various possible interfaces and autonomy modes within this team. For example, robot 1 might be capable of operating in teleoperation mode, waypoint mode, or path-planning mode. We may want to determine which of these autonomy modes is most compatible with the autonomy modes of the rest of the team.

Associated with each $\mathcal{N}(\boldsymbol{\pi})$ is the set of average performance levels for each robot. Recall that when we select a performance threshold, the neglect impact and interface efficiency curves dictate (NT, IT) values. When the robot is operated at those values, the performance of the robot rises and declines as the robot is serviced and neglected over time. The temporal average over these fluctuations is, however, consistent if the world is stationary. This means that selecting $\boldsymbol{\pi}$ dictates not only the set of (NT, IT) values for each robot, but also the average performance for each robot, which we denote $\bar{J}_i(\pi_i)$.

Given the average performance of each robot, the expected average performance of the robot team as

$$\mathcal{J}(\boldsymbol{\pi}) = \frac{1}{M} \sum_{i=1}^{M} \bar{J}_i(\pi_i). \tag{5}$$

The summation indicates that we assume independence of the robots; the team performance consists of the sum of the individual performances of the robots. If we wish to select interaction schemes for a team of M robots that maximize expected team performance, then we should choose $\boldsymbol{\pi}$ to maximize $\mathcal{J}(\boldsymbol{\pi})$ subject to the constraint that the neglect characteristics, $\mathcal{N}(\boldsymbol{\pi})$ imposed by $\boldsymbol{\pi}$ are feasible.

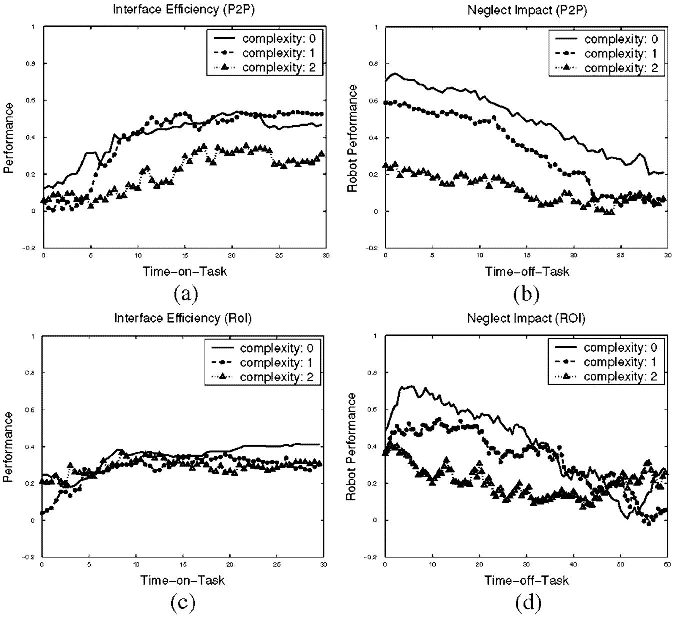
The bottom line of this analysis is that, for a given team size, we can search through the set of feasible team configurations to find the set of interaction schemes and performance thresholds that maximize performance. Doing so requires us to obtain neglect impact and interface efficiency curves, select appropriate performance thresholds, and limit attention to robots performing independent tasks. In this methodology, the predicted performance is obtained from the neglect impact and interface efficiency curves of individual robots. Insofar as the performance predicted in this way agrees with the true performance values, this methodology for selecting team configurations is valid. Validating that predictions agree with reality is addressed in the next section.

B. User Study

To validate that the predicted team performance agrees with the actual performance, we used a between-subjects[8] experiment design. The first group consisted of 13 subjects who were trained to perform a goal-finding and exploration task using a robot with one of the specified interaction schemes. Operators were shown a grid-based map of the explored areas of a simulated world. The unexplored portion of the world, however, was left blank on the map. The position of the robot and its goal was shown in the world, along with the sensory information of the robot. Note that neglect impact and interface efficiency curves are constructed as before using secondary task studies. To gather data more efficiently, both math problems and guiding a second robot (using the same interaction scheme) were used as secondary tasks. Each robot had its own unique goal to find so the two robots were almost entirely independent.[9]

Subjects were trained on the two interaction schemes (*P2P* and ROI) and then performed six 5-min sessions during which data was gathered. Each session took place in a different simulated world. Each world was classified as one of three different complexities (referred to as complexities 1–3),[10] depending on the number of dead ends which the world contained. The interface efficiency and neglect impact curves for the two interaction schemes are shown in Fig. 9.

The data obtained from this set of subjects was then used to generate the predicted performance for teams of three robots using all possible unique combinations of *P2P* and ROI. Thus, there were four possible team configurations, denoted PPP, PPR, PRR, and RRR, and a predicted team performance for each configuration. For example, PPR indicates that two of the robots were to use *P2P* and the other robot was to use ROI. The performance thresholds that were used to determine (NT, IT) were those that maximized (5), subject to the constraint that $\mathcal{N}(\pi)$ was feasible.



(a) (b) (c) (d)

Fig. 9. Interface efficiency and neglect impact curves for *P2P* and ROI.

A second group of 24 subjects were then asked to manage a team of three robots in the same worlds used with the first group of subjects. The between-subjects experiment design compared predicted performance from the first group to the observed performance from the second group. Since the subjects in the second group controlled three robots, three different goals were present at any time. Any of the three robots could collect any of the three goals. When a goal was collected, another goal appeared. The session concluded when nine goals had been gathered. During a session, the user could interact with any of the three robots at any time by clicking on that robot in the map of the world.

Since any robot could collect any goal, the performance of the robots was not completely independent. We will show that this interdependence is not problematic for some situations, but causes problems for predictions in others. For each world complexity and team configuration, 9–15 samples of performance were obtained.

C. Results

The average (across trials) of the time required to complete the task is plotted against the predicted completion time in Fig. 10. In the figure, lower values indicate superior performance because the task was finished more quickly.

Overall, the predicted and observed completion times are in close agreement, especially for low complexity (although the data is not statistically significant). However, as complexity increases, a discrepancy between prediction and observation appears for PPP $(\mathrm{C} = 1, \mathrm{C} = 2)$, and PPR $(\mathrm{C} = 2)$. The predicted completion times are much higher than the actual completion times. This discrepancy is caused by a shift in the strate-

[8] Subjects used in this user study were male and female volunteers from a wide range of backgrounds and ages (13–50+) with no prior experience driving the robots.

[9] The exception to this independence is that having two robots causes the environment to be explored more rapidly.

[10] This was done, again, to simply the user study.

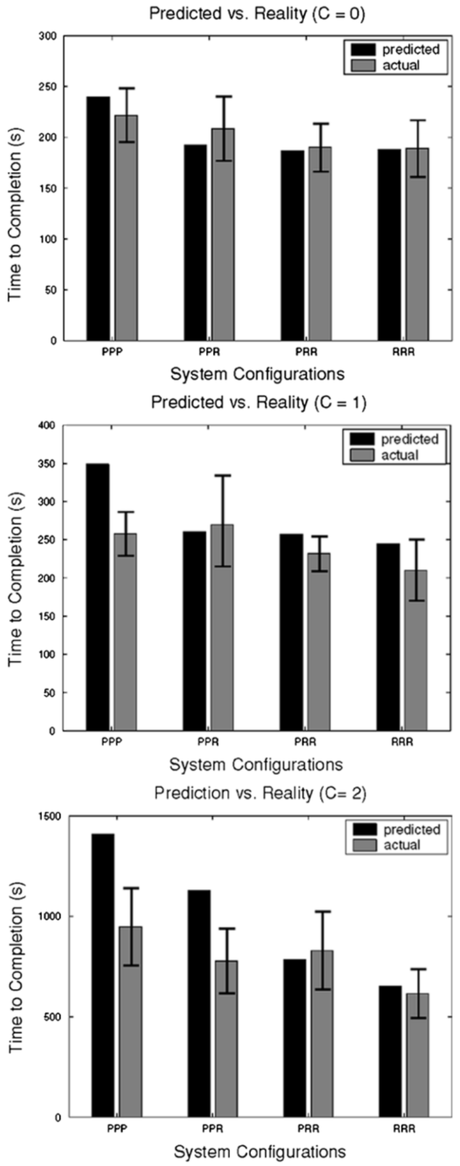


Fig. 10. Predicted performance versus measured performance. Confidence intervals (95%) are also shown. From bottom to top, complexity is increasing.

gies used by subjects as they managed the teams in complex worlds with simple interaction schemes. The workload in these experiments was extremely high, so subjects exploited the fact that the tasks were not completely independent to deal with the workload. They typically chose to place robots in different regions of the world (a "zone defense" approach) and wait until a goal popped up in one of those regions before guiding a robot to the goal. If, as goals were collected, no goals popped up in a particular region, the robot in that region was ignored and the other robots received attention.

With this exception caused by the interdependence of the tasks, predictions are in close agreement with observed completion times. This suggests that the predictions are informative enough to distinguish between various team configurations to select a configuration that will likely maximize team performance. Interestingly, the situations in which predictions were not accurate were situations in which the model predicted that using only two robots would be as effective as using three robots. In this way, the model seems to identify situations of operator overload.

Another important point needs to be made. Even with the slight prediction errors in the absolute values of the performance, the ordering of the predicted team performance is almost identical to the ordering of the observed team performance. This suggests that the team configuration that is likely to produce the highest performance can be determined using the analysis. This appears to be true even when there are minor violations of the independence assumptions.

VIII. Workload-Performance Tradeoffs

There is a tradeoff between the amount of human workload and the expected performance of an individual robot. More specifically, if a human is immersed in a situation where a given set of neglect times and interaction times are required because of workload conditions, then the average performance of the robot is determined by those conditions. Conversely, if a particular level of average performance is dictated by the problem, then neglect time and interaction time follow because of the characteristics of the neglect impact and interface efficiency curves.

To explore the tradeoff between workload and performance, it is useful to have a single parameter to characterize workload. Robot attention demand (RAD) is a very useful parameter for doing this. RAD is defined as the fraction of an interaction cycle that is consumed by interaction ($\mathrm{RAD} = (\mathrm{IT})/(\mathrm{IT} + \mathrm{NT})$, and is an estimate of the fraction of a human's time that is dedicated to a robot. Insofar as time commitment predicts workload, RAD is a suitable estimate of workload.

By creating cross plots of RAD against average robot performance, we can visualize the tradeoff between average robot performance and operator workload. By varying the performance threshold, we can "sweep" out a range of RAD and average performance values. This allows us to compare various interaction schemes according to the performance-workload tradeoff. If one interaction scheme is superior (meaning it has higher performance and lower workload) to a second interaction scheme for all complexity values, then the second scheme is never an appropriate design choice.

Fig. 11 compares the *P2P* and *Scripted* interaction schemes from the real world experiment discussed in Section VI in terms of RAD and average robot performance. Plots are shown for both world complexities. Interaction schemes are typically better if their points are in the upper left-hand corner of the plots, and are not as good if they are toward the bottom right-hand corner of the plots. The plots show that while *Scripted* requires less operator workload, *P2P* yields higher average robot performance.

A similar analysis for the workload/performance tradeoffs of the simulated worlds user study of Section V and of the heterogeneous robot teams of Section VII-B can be found in [53] and [54], respectively.

IX. Conclusion and Future Work

Since improving robot autonomy allows a robot operator to have free time, it is important to determine the neglect toler-

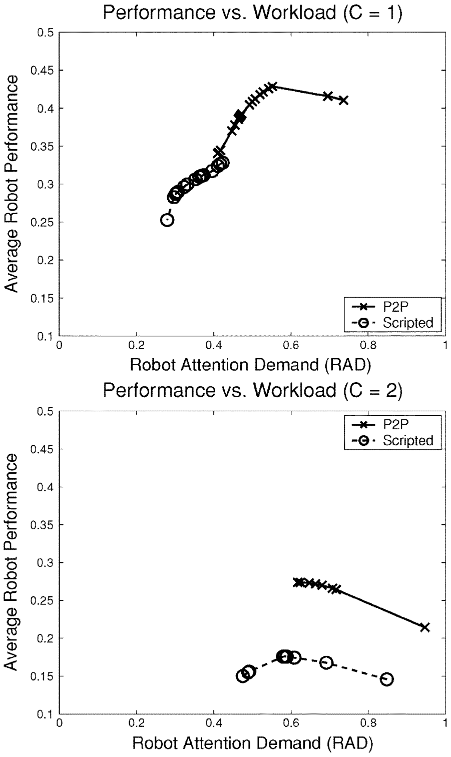


Fig. 11. *P2P* and *Scripted* in the real world in terms of RAD and average robot performance for world complexities 1 (top) and 2 (bottom).

ance characteristics of an interaction scheme. These neglect tolerance characteristics include neglect time, interaction time, and average performance. Fanout, the maximum number of independent homogeneous robots that can be managed by a single operator, can be determined using the ratio of neglect time and interaction time. The feasibility of a team of independent heterogeneous robots can be determined likewise. The predicted performance of such various heterogeneous team configurations can be used to select a team configuration that is likely to maximize performance subject to given workload conditions of the operator. In this paper, we have chosen to illustrate how neglect tolerance characteristics can be used to compare and discriminate between various autonomy modes. However, the same technology can be used to compare various interface designs, especially as the effects of task switching become more important.

Future work should include extending these results to predicting performance in general multitasking domains. The (NT, IT) characteristics are applicable in general scheduling domains, and therefore apply to general multitasking problems. However, predicting the performance of a robot in a general multitasking domain needs to be verified. A second area of future work is determining efficient ways to select the performance threshold. In this paper, we used average performance as the basis for determining the threshold. While this may be appropriate for some circumstances, alternatives such as being "90% confident that the robot won't fail" can also be used to select thresholds. A third area of future work should include determining the neglect tolerance characteristics of teams of interdependent robots. A fourth area of future work should explore more thoroughly how task switching affects the neglect characteristics of a robot team. The final area of future work is to identify how various choices made when designing the robot's autonomy or the user's interface affect performance and workload. Ideally, this would consist of a "toolbox" of autonomy and interface choices that are known to be appropriate for a given set of performance and workload constraints for a given task.

References

[1] L. Conway, R. A. Volz, and M. W. Walker, "Teleautonomous systems: Projecting and coordinating intelligent action at a distance," *IEEE Trans. Robotics Autom.*, vol. 6, no. 2, pp. 146–158, Apr. 1990.

[2] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, MA: MIT Press, 1992.

[3] D. Pomerleau, "Ralph: Rapidly adapting lateral position handler," in *Proc. IEEE Symp. Intell. Veh.*, Detroit, MI, Sep. 1995.

[4] M. L. Cummings and S. Guerlain, "Developing operator capacity estimates for supervisory control of autonomous vehicles," Presence, 2005, submitted for publication.

[5] J. L. Burke, R. R. Murphy, M. D. Coovert, and D. L. Riddle, "Moonlight in miami: A field study of human–robot interaction in the context of an urban search and rescue disaster response training exercise," *Human–Comput. Interaction*, vol. 19, pp. 85–116, 2004.

[6] J. Casper and R. R. Murphy, "Human–robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 3, pp. 367–385, Jun. 2003.

[7] H. A. Yanco, J. L. Drury, and J. Scholtz, "Beyond usability evaluation: Analysis of human–robot interaction at a major robotics competition," *J. Human–Comput. Interaction*, vol. 19, no. 1/2, pp. 117–149, 2004.

[8] D. D. Woods, J. Tittle, M. Feil, and A. Roesler, "Envisioning human–robot coordination in future operations," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 2, pp. 210–218, May 2004.

[9] P. L. Alfano and G. F. Michel, "Restricting the field of view: Perceptual and performance effects," *Perceptual Motor Skills*, vol. 70, no. 1, pp. 35–45, 1990.

[10] K. Arthur, "Effects of field of view on performance with head-mounted displays," Ph.D. dissertation, Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, 2000.

[11] B. Ricks, C. W. Nielsen, and M. A. Goodrich, "Ecological displays for robot interaction: A new perspective," in *Proc. IROS*, Sendai, Japan, 2004.

[12] T. Fong, C. Thorpe, and C. Baur, "A safeguarded teleoperation controller," in *IEEE Int. Conf. Adv. Robot.*, Budapest, Hungary, Aug. 2001.

[13] E. Krotkov, R. Simmons, F. Cozman, and S. Koenig, "Safeguarded teleoperation for lunar rovers: From human factors to field trials," in *IEEE Planetary Rover Technol. Syst. Workshop*, Minneapolis, MN, 1996.

[14] *IEEE Robot. Autom. Mag.*, vol. 8, no. 1, pp. 4–73, Mar. 2001.

[15] R. C. Arkin, "Reactive control as a substrate for telerobotic systems," *IEEE AES Syst. Mag.*, vol. 6, no. 6, pp. 24–31, Jun. 1991.

[16] M. A. Goodrich and E. R. Boer, "Designing human-centered automation: Tradeoffs in collision avoidance system design," *IEEE Trans. Intell. Transport. Syst.*, vol. 1, no. 1, pp. 40–54, Mar. 2000.

[17] G. A. Dorais, R. P. Bonasso, D. Kortenkamp, B. Pell, and D. Schreckenghost, "Adjustable autonomy for human-centered autonomous systems on mars," in *Proc. 1st Int. Conf. Mars Soc.*, Boulder, CO, Aug. 1998.

[18] G. Cheng and A. Zelinksy, "Supervised autonomy: A framework for human robot systems development," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 2, Tokyo, Japan, Oct. 1999, pp. 971–975.

[19] ——, "Supervised autonomy: A paradigm for teleoperating mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Grenoble, France, Sep. 1997.

[20] T. Fong, C. Thorpe, and C. Baur, "Collaborative control: A robot-centric model for vehicle teleoperation," in *AAAI 1999 Spring Symp.: Agents Adjust. Auton.* Stanford, CA, Mar. 1999.

[21] D. Perzanowski, A. C. Schultz, W. Adams, and E. Marsh, "Goaltracking in a natural language interface: Toward achieving adjustable autonomy," in *Proc. IEEE Int. Symp. Computat. Intell. Robot. Autom.*, 1999, pp. 208–213.

[22] E. R. Boer, O. Nakayama, T. Futami, and T. Nakamura, "Developing of a steering entropy method for evaluating driver workload," in *Proc . Int. Congress Expo.*, Detroit, MI, 1999.

[23] M. R. Endsley, “Design and evaluation for situation awareness enhancement,” in *Proc. Human Factors Soc. 32nd Annu. Meeting*, Santa Monica, CA, 1988, pp. 97–101.

[24] A. A. Nofi, “Defining and measuring shared situation awareness,” in *Center Naval Anal.*, Alexandria, VA, Nov. 2000.

[25] J. Scholtz, “Human–robot interactions: Creating synergistic cyber forces,” in *Proc. NRL Workshop Multirobot Syst.*, Washington, DC, Mar. 2002.

[26] T. W. Fong, C. Thorpe, and C. Baur, “Advanced interfaces for vehicle teleoperation: Collaborative control, sensor fusion displays, and remote driving tools,” *Auton. Robots*, vol. 11, no. 1, pp. 77–85, 2001.

[27] R. Meier, T. Fong, C. Thorpe, and C. Baur, “A sensor fusion based user interface for vehicle teleoperation,” in *Proc. Int. Conf. Field Service Robot.*, Pittsburgh, PA, 1999.

[28] C. D. Wickens and J. G. Hollands, *Engineering Psychology and Human Performance*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[29] H. A. Yanco and J. L. Drury, ““Where am I?” Acquiring situation awareness using a remote robot platform,” in *Proc. IEEE Conf. Syst., Man, Cybern.*, The Hague, The Netherlands, Oct. 2004.

[30] T. Fong, N. Cabrol, C. Thorpe, and C. Baur, “A personal user interface for collaborative human–robot exploration,” in *Int. Symp. Artif. Intell., Robot., Autom. Space*, Montreal, Canada, Jun. 2001.

[31] D. Kortenkamp, E. Huber, and R. P. Bonasso, “Recognizing and interpreting gestures on a mobile robot,” in *Proc. AAAI*, Portland, OR, 1996.

[32] R. Voyles and P. Khosla, “Tactile gestures for human/robot interaction,” in *Proc. IEEE/RSJ Conf. Intell. Robots Syst.*, vol. 3, Pittsburgh, PA, 1995, pp. 7–13.

[33] C. Breazeal, “A motivational system for regulating human–robot interaction,” in *Proc. AAAI*, 1998, pp. 54–61.

[34] D. Roy and A. Pentland, “Learning words from sights and sounds: A computational model,” *Cogn. Sci.*, vol. 26, no. 1, pp. 113–146, 2002.

[35] F. Steele, G. Thomas, and T. Blackmon, “An operator interface for a robot-mounted, 3-D camera system: Project pioneer,” in *Proceedings of the 1998 IEEE. Virtual Reality Conference* Houston, TX, 1998.

[36] J. C. Lane, C. R. Carignan, and D. L. Akin, “Advanced operator interface design for complex space telerobots,” in *Veh. Teleoper. Interfaces Workshop, IEEE Int. Conf. Robot. Autom.*, San Francisco, CA, Apr. 2000.

[37] C. W. Nielsen and M. A. Goodrich, “Iconic images for semantic maps,” in *Proc. IEEE Conf. Syst. Man, Cybern.*, The Hague, The Netherlands, 2004.

[38] R. R. Murphy and E. Rogers, “Cooperative assistance for remote robot supervision,” *Presence*, vol. 5, no. 2, pp. 224–240, 1996.

[39] R. M. Boyles and P. K. Khosla, “A multiagent system for programming robots by human demonstration,” *Integr. Computer-Aided Eng.*, vol. 8, no. 1, pp. 59–67, 2001.

[40] D. D. Dudenhoeffer and D. J. Bruemmer, “Command and control architectures for autonomous micro-robotic forces,” Idaho Nat. Eng. Environ. Lab., Idaho Falls, ID, FY00 Project Rep. INEEL/EXT-2001-00232, 1992.

[41] K. Ali and R. C. Arkin, “Multiagent teleautonomous behavioral control,” *Mach. Intell. Robot. Control*, vol. 1, no. 2, pp. 3–10, 2000.

[42] R. C. Arkin and K. Ali, “Integration of reactive and telerobotic control in multiagent robotic systems,” in *Proc. 3rd Int. Conf. Simul. Adaptive Behav.*, Brighton, U.K., 1994, pp. 473–478.

[43] T. Balch and R. A. Arkin, “Behavior-based formation control for multirobot teams,” *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 912–925, Dec. 1999.

[44] L. E. Parker, “ALLIANCE: An architecture for fault tolerant, cooperative control of heterogeneous mobile robots,” in *Proc. IEEE/RSJ/GI Int. Conf. Intell. Robots Syst.*, vol. 2, Munich, Germany, Sep. 12–16, 1994, pp. 776–783.

[45] I. Koch, “The role of external cues for endogenous advance reconfiguration in task switching,” *Psychon. Bullet. Rev.*, vol. 10, pp. 488–492, 2003.

[46] N. J. Cepeda, A. F. Kramer, and J. C. M. G. de Sather, “Changes in executive control across the life span: Examination of task-switching performance,” *Develop. Psych.*, vol. 37, no. 5, pp. 715–730, 2001.

[47] E. M. Altmann and J. G. Trafton, “Task interruption: Resumption lag and the role of cues,” in *26th Annu. Conf. Cogn. Sci. Soc.*, Chicago, IL, 2004.

[48] H. E. Pashler, *The Psychology of Attention*. Cambridge, MA: MIT Press, 1997.

[49] D. R. Olsen and M. A. Goodrich, “Metrics for evaluating human–robot interactions,” in *Proc. PERMIS*, Gaithersburg, MA, 2003.

[50] D. R. Olsen Jr. and S. B. Wood, “Fan-out: Measuring human control of multiple robots,” in *Proc. Conf. Human Factors Comput. Syst.*, 2004, pp. 231–238.

[51] D. R. Olsen Jr., S. B. Wood, and J. Turner, “Metrics for human driving of multiple robots,” in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, 2004, pp. 2315–2320.

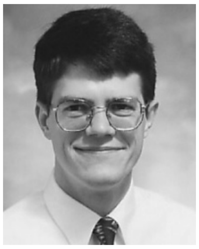
[52] M. A. Goodrich, M. Quigley, and K. Cosenzo, “Task switching and multirobot teams,” in *Proc. 3rd Int. Multirobot Syst. Workshop*, Washington, DC, Mar. 2005.

[53] J. W. Crandall, “Toward developing effective human–robot systems,” M.S. thesis, Dept. Comput. Sci., Brigham Young Univ., 2003.

[54] J. W. Crandall, C. N. Nielsen, and M. A. Goodrich, “Toward predicting robot team performance,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2003, pp. 906–911.

Jacob W. Crandall received the B.S. and M.S. degrees in computer science in 2001 and 2003, respectively, from Brigham Young University, Provo, UT, where he is currently pursuing the Ph.D. degree in computer science.

His research interests include multiagent learning, human–robot interaction, and robotics.



Michael A. Goodrich (S'92–M'92–SM'05) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering from Brigham Young University, Provo, UT, in 1992, 1995, and 1996, respectively.

From 1996 to 1998, he was a Research Associate with Nissan Cambridge Research, Nissan Research and Development, Inc., Cambridge, MA. Since 1998, he has been with the Computer Science Department. Brigham Young University, where he is currently an Associate Professor. His research interests include human–robot interaction, decision theory, multiagent learning, and human-centered engineering.

Dan R. Olsen, Jr., received the B.S. and M.S. degrees in computer science from Brigham Young University, Provo, UT, in 1976 and 1978, respectively, and the Ph.D. degree in computing and information from the University of Pennsylvania, Philadelphia, in 1981.

He is currently a Professor of computer science at Brigham Young University (1984–1996 and 1998–present). He was formerly a Professor of human–computer interaction at Carnegie Mellon University, Pittsburgh, PA, from 1996 to 1998. His research interests are in the technologies of human–computer interaction. His recent work has focused on the interaction between human intelligence and machine learning.

Prof. Olsen was the Founding Editor-in-Chief of ACM's *Transactions on Computer–Human Interaction*. He has served as the Vice Chair for both finance and publications of the ACM's Special Interest Group on Computer–Human Interaction (SIGCHI). He is the recipient of SIGCHI's Lifetime Service Award (2002) and his research has been recognized by induction into the CHI Academy (2004).

Curtis W. Nielsen received the B.S. and M.S. in computer science in 1999 and 2003, respectively, from Brigham Young University, Provo, UT, where he is currently pursuing the Ph.D. degree in computer science.

His research interests include computer graphics, robotics, human–robot interaction, and search and rescue.