Multiagent Telerobotics: Matching Systems to Tasks

A Thesis
Presented to
The Academic Faculty

by

Khaled Subhi Ali

In Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Georgia Institute of Technology
June 1999

Multiagent Telerobotics: Matching Systems to Tasks

Approved:

Ronald C. Arkin

Albert N. Badre

Jessica K. Hodgins

Alexander C. Kirlik

Sven Koenig

Date Approved

Preface

A multiagent telerobotic system (MTS) is a system that allows a human to control a group of robots. Multiagent robotics has certain desirable properties when compared to single-agent robotics, and telerobotics has desirable properties when compared to autonomous robotics. Whether the combined field of multiagent telerobotics will inherit those advantages, however, remains to be shown. Until now, no comparison of the performance of different multiagent telerobotic systems in terms of tasks has been conducted. Therefore, there were no guidelines to aid developers building a MTS for a particular task.

This research compares the performance of different classes of mobile behavior-based multiagent telerobotics systems in relation to the kinds of tasks they are performing. The systems are compared in terms of safety, effectiveness, and ease-of-use, for applications representing classes of tasks in a newly-developed taxonomy of mobile multiagent tasks. This taxonomy categorizes the tasks in terms of the relative motion of the agents. Four different task classifications from this taxonomy were studied.

A methodology for evaluating the performance of robot systems for the tasks was adapted from standard experimental procedures, and a series of experiments was conducted, in which over a hundred human participants were used to control real robots. The end result of the experiments is a knowledge base relating the systems to the tasks. The systems were ranked in terms of their safety, effectiveness, and ease-of-use for each task. Additionally, where possible, more general results were identified, relating the type of system to the type of task. These results should guide future MTS developers to build safe, effective, and easy-to-use systems.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivations

Multiagent telerobotics is a new and emerging field of research. It has the potential to change the way roboticists solve problems, by combining the power and robustness of multiagent robotic systems with the flexibility of telerobotic systems. The military is already gravitating toward multiagent robotic systems for scouting missions into hostile territory and teams of robots in urban territory. Examples of this include the Unmanned Ground Vehicle (UGV) Demo II project [56] and the Tactical Mobile Robotics (TMR) project [46]. Providing telerobotic control capabilities for these multiagent systems allows rapid development and flexibility, by allowing a human to assist the robots in unforeseen situations and those in which fully autonomous robots are not yet capable enough. Likewise, single-agent telerobotic systems, such as the Sojourner robot [37] sent to explore Mars, and the Pioneer project [34], which will send a robot into the Chernobyl nuclear facility to examine its current state, provide roboticists the ability to rapidly field robots in circumstances where the robot's environment and tasks are not fully known in advance. Both multiagent robotics and telerobotics, the parents of multiagent telerobotics, have certain features that make multiagent telerobotics appear appealing to designers and users of robotic systems. These advantages are described below.

Multiagent robotics has many desirable properties when compared to single-agent robotics. Some of these properties include:

- a larger range of possible tasks
- greater efficiency
- robustness
- lower economic cost
- ease of development

Theoretically, multiple robots should be able to accomplish any task that a single robot can. Additionally, since groups of robots can cover more ground than a single robot, there are tasks that multiple robots can accomplish that a single robot cannot. For instance, a pair of robots could work together on line-of-sight tasks, such as maintaining a communications channel or surveying. In these tasks, coordinated actions must be taken at distant locations. A single robot could not accomplish this task, since it can only be in one place at a time.

Multiple robots may also offer performance benefits. For instance, Balch and Arkin [12] show how multiple robots provide speedup for foraging and similar tasks.

Furthermore, multiagent robotics offers robustness. If one robot malfunctions or is destroyed, the others can continue the task. This is a prime advantage for military robots, which may be under attack from enemy forces or damaged while clearing fields of landmines.

Finally, a single robot system for a particular task may need to be more complicated than each of the individual robots in a multiagent robot system for the same task. A single robot system will need to handle all aspects of the task, while a multiagent system can divide the subtasks among the robots, requiring each robot to know only how to accomplish its subtask. Therefore, a multiagent system is often less expensive and easier to develop than an equivalent single-agent system [16].

Telerobotics also has several desirable properties when compared to fully autonomous robots, such as:

- greater capability
- providing for opportunism
- social acceptance
- appropriateness for one-of-a-kind tasks
- support for robot learning from humans

It is obvious that autonomous robots are still not as capable at many tasks as a human or a human controlling a teleoperator or telerobot. This allows a telerobot to be rapidly fielded for a particular job, whereas it might be years before an autonomous robot could accomplish the task as well.

Furthermore, telerobots offer a level of opportunism not present in autonomous robots. The human operator may notice an opportunity that the robot has not been programmed for. This is especially true for tasks where the exact specifications for a solution cannot be determined in advance, such as in exploration, artistic and creative tasks, or other poorly defined tasks.

Telerobots also tend to be more socially acceptable than autonomous robots. It seems society may trust a robot under human supervision and control more than one that retains the final authority in the decision-making process. A human operator can recognize failures and situations where the robot should not follow its program, possibly averting a disaster.

Telerobots outshine autonomous robots for tasks that will only be conducted once, where programming an autonomous robot would be far more costly. Finally, even humans are not experts in all areas, and they require coaching or teaching for many jobs. Thus, a robot will never be rid of the need for expert coaching, even if we assume that it can achieve human competence. Telerobotics provides just such an appropriate, easy to implement coaching/teaching method.

It appears that multiagent telerobotics inherits some of these desirable qualities from its parents (multiagent robotics and telerobotics). If this is true, then multiagent telerobotics is not only the next logical step for the telerobotics community, but it is possibly the best solution for many robotics problems. This, however, remains to be demonstrated through analysis and experimentation.

While a significant amount of research has examined telerobotics with a single agent [47, 48, 4], very little work has been conducted in the area of multiagent telerobotics. Therefore, there is little to confirm that the principles of single-agent telerobotics will generalize to multiagent mobile telerobotics. Due to the apparent potential of Multiagent Telerobotic Systems (MTSs), more effort needs to be channeled into this new field. A few multiagent telerobotics systems [57, 40, 1, 18, 7] have been developed recently. As is the case with many new fields, however, each research effort has produced a single system for one particular task. None of the research to date compares the kinds of systems, or states why they are appropriate for particular tasks. Furthermore, no one has presented guidelines for developing MTSs. Experiments need to be conducted to determine the appropriateness of various kinds of MTSs for various classes of tasks. This dissertation's research is intended to help fill that need.

1.2 Terminology

The following terminology is used in the remainder of this section. This terminology, as well as others used in the remainder of this dissertation, can also be found in Appendix A.

Multiagent robotics refers to using more than one robot to complete the same task. The robots may be working on the same or different subtasks, and they are not required to use the same approach to the task.

The strict definition of a *telerobot* is a robot that determines its actions based on some combination of human input and autonomous control. A telerobot can use *shared control* or *strict supervisory control*. With shared control, the instructions given by the human and the robot are combined in some manner to determine the motion of the robot. With strict supervisory control, the human operator instructs the robot and then observes the robot as it attempts to autonomously carry out the instructions. If there is a problem, the human may help out by giving more instructions. *Supervisory control*, in the less strict sense of the word, is often used to refer to shared control, supervisory control, and combinations of the two approaches. These types of supervisory control are described in more detail in Section 2.2.

In this research, the term telerobot will be used to refer to both true telerobots and to *teleoperators*. A teleoperator is a machine that uses *direct manual control*. This means that the human operator has complete control over the robot's actions. Aside from its physical limitations, the robot does not contribute to determining its motion. An example of a teleoperator is a radio controlled toy car.

A *multiagent telerobotic system* (MTS) is a group of more than one telerobot controlled by a human operator.

Behavior-based robots determine their actions through some combination of the output of one or more simple behaviors. Each behavior takes care of one aspect of the task, such as obstacle avoidance.

A *mobile telerobot* is a telerobot that is capable of moving itself. For instance, a telerobotic car would be a mobile telerobot, while a telerobot arm that is bolted down to a table would not.

1.3 Research Questions

The following research question inspired and directed this research. An additional five subsidiary questions arose from this primary question. Initially, it was intended that this dissertation should answer all of these subsidiary questions in support of the main question. The research has since focused on the primary question and three of the subsidiary questions. The rationale for this follows the presentation of the questions themselves.

1.3.1 Research Question

Given a task, what form of telerobotic system allows a human operator to safely, effectively, and easily control a group of behavior-based mobile telerobots for that task?

Some MTSs exist already [57, 40, 1, 18, 7] that present methods for controlling groups of robots, although no one has evaluated each to determine how it performs in comparison to other types of systems. Additionally, many control techniques exist for single-robot systems, and these also might be appropriate for controlling robot teams. It would benefit both the developers and users of MTSs to know what sort of control techniques are best for which kinds of tasks. Furthermore, it is to everyone's advantage, including single-agent teleroboticists and autonomous roboticists, to have a standard methodology for comparing various robotic system types.

The goal of this research was to discover the relationships between different multiagent telerobotic systems, when compared regarding their use in different types of multiagent mobile robot tasks. These relationships should provide a basis for building safe, effective, and easy-to-use MTSs. *Safe* control indicates that the robots will not cause unintended harm to themselves, other robots, or objects in their environment. *Effective* control indicates that the human operator will be able to accomplish the tasks that he intends to do with the telerobots as quickly as possible. *Easy-to-use* control indicates that the human operator will be able to accomplish these tasks with a minimum of stress and cognitive overload.

To discover these relationships, multiagent telerobotic systems were developed along two different dimensions (described in Section 4.1) and tested for each of these criteria (safety, ease-of-use, and effectiveness). A taxonomy of tasks for multiagent mobile robot systems was developed and is presented in Section 5.1. Each system was examined for four classes of tasks in this taxonomy. Experiments were conducted with real robots to evaluate the telerobotic systems for tasks that represent certain categories in this taxonomy. Figure 1.1 shows the robots used for one of the tasks. The results of these experiments were examined to determine which system types produce the best results for which kinds of tasks. These results can be used by other MTS developers and users to design or choose safe, effective, and easy-to-use systems.

1.3.2 Subsidiary Questions

How much influence should the operator have over the actions of the robot group for particular classes of tasks?

One dimension along which multiagent telerobotic systems can differ is the level

Figure 1.1: Real robot teams were used in the experiments. Here we see them at the start of one of the examined tasks.

of autonomy of the robots. Telerobotic systems with many different levels of autonomy are currently in use. While experiments [47] have been conducted to determine the effect of the level of autonomy for single-agent telerobotic systems, to our knowledge, none have examined the effect of these systems for controlling multiple robots.

Telerobotic systems differing in the amount of autonomy of the robot group were developed and tested for different task classes. Specifically, two points along the dimension of autonomy were tested, namely direct manual control and supervisory control. Details of these systems are given in Chapter 4.

How many robots should the operator control at one time for particular classes of tasks?

Another dimension involves the number of telerobots that the human operator controls at a time. The operator could direct each robot individually or control the entire group of robots at once. Furthermore, subgroups of robots, ranging from one to the entire group could be controlled. In this research, the number of robots controlled at a time was examined in relation to the task classes. In particular, two points in this dimensional space were considered, namely, individual control and group control. These systems are described in more detail in Chapter 4.

What form of control should the human operator have in order to control a group of mobile telerobots for particular classes of tasks?

The human operator needs to control the telerobots by giving them instructions. But what form should the instructions take? For example, the operator can give directional instructions, speed instructions, task-related instructions (such as, "Go to the mailroom" or "Assume column formation"), or instructions to change behavioral parameters. This dimension, however, was not examined in the experiments, and the rationale for its exclusion appears in Section 1.3.3. Section 4.2 describes the forms of control used with the systems that were examined.

What form should the interaction between the human operator and the robot group take for particular classes of tasks?

Once it is decided what kind of instructions the operator should give to the robot group, there must be some method to give these instructions through the human interface. For example, if directional instructions are to be given, then the instructions can be given either in terms of an amount to turn (yaw and/or pitch), a direction to move in, or a location to go to. Similarly, speed instructions can be given in terms of acceleration, velocity, or a deadline to arrive at a location. Parameter changes can also be specified in multiple ways. The parameter(s) to change can be given in terms of parameter names or abstract groupings of parameters. The forms of interaction used for the experimental MTSs are described in Section 4.2.

How can multiagent telerobotic systems be evaluated based on the criteria of safety, ease of use, and effectiveness?

Tests were conducted in which a large number of human participants used the different systems to control a group of telerobots for certain tasks. Measurements were taken to determine how well each system satisfied the criteria of safety, effectiveness, and ease-of-use. Each system configuration was tested for each class of tasks, to determine the dependencies between the task and the multiagent telerobotic system.

An experimental methodology that suits this sort of large-scale testing was developed to evaluate the systems for each task. This methodology combines elements from multiple fields, borrowing heavily from the areas of user-interface evaluation and human-factors studies. This methodology is appropriate for further evaluations, either of multiagent or single-agent systems.

1.3.3 Comments on Research Questions

Four of the five subsidiary questions present dimensions along which multiagent telerobotic systems can differ, namely:

- the amount of autonomy of the robots
- the number of robots controlled at a time
- the form of control
- the form of interaction between human and robot

It was initially intended that all of these dimensions would be examined in a series of four-factor experiments. Further examination, however, revealed that it would not be possible to conduct a study this broad and still pay enough attention to each dimension to generate usable results. The number of system/task combinations grows exponentially with the number of dimensions being examined. Since the number of experimental subjects available for testing is effectively bounded, it is better to examine fewer system types. This allows more subjects to be used per system, which produces more reliable results.

Therefore, only two of the four dimensions were chosen for study, and a series of two-factor experiments were conducted. The two dimensions chosen for examination are (1) the amount of autonomy of the robots and (2) the number of robots controlled at a time. The methodology that is presented, however, is just as appropriate for studying the relationship of the other two dimensions, or many other dimensions of telerobotic systems. It is a general methodology for telerobotic system evaluation, both multiagent and single-agent. Only time and resource limitations prevented their study here.

1.4 Research Overview

The purpose of this research is to compare different classes of mobile behavior-based multiagent telerobotic systems in order to determine which one is best for what types of tasks. Large-scale evaluations were conducted with the systems, comparing them in terms of safety (both for the robots and their environment), effectiveness (in terms of task completion and speed of completion), and ease-of-use.

Four classes of systems were chosen for examination. These systems varied in the amount of autonomy the robots possessed and the number of robots the human operator controlled at a given time. **Direct manual** control systems and **supervisory** control systems were tested. Likewise, **individual** and **group** control was evaluated.

In order to determine which MTSs perform best for which tasks, it is necessary to know what types of tasks exist. Additionally, when discussing tasks for multiagent systems, it is helpful to have a formal means of specifying their nature, such as a taxonomy. No taxonomy of mobile multiagent tasks previously existed, and so one was developed. This taxonomy classifies mobile multiagent tasks in terms of the relative motion of the robots. The experiments examined four classes of tasks from this taxonomy: *movement-to-coverage*, *movement-to-convergence*, *movement-while-maintaining-coverage*, and *movement-while-maintaining-convergence*.

Over 100 human subjects used the MTSs for tasks representing each of the chosen task classes. Telerobot evaluations (as well as fully autonomous robot evaluations with real robots) with this many trials are believed to be unprecedented. This sort of rigorous examination, however, is necessary to turn robotics into a science. The end result of these experiments is a knowledge base relating the examined systems to various tasks in terms of safety, effectiveness, and ease-of-use. This knowledge base includes rankings of the system types for each of the tasks, as well as more general findings relating the levels of the system dimensions to the tasks.

The methodology that was followed for this dissertation's experiments is presented, both in terms of the current application and in a general manner, so that it can be used for other robot system evaluations. The general presentation of the methodology in Chapter 3 includes a few recommendations that were not followed in the current application. These suggestions are based on the experience gained from conducting these experiments. If this dissertation's evaluations were to be repeated, they would follow the process outlined in Chapter 3, including the additional recommendations.

In summary, the contributions of this work include:

- Rankings of four types of MTSs for four classes of tasks

- General findings relating the levels of system dimensions to task classes
- A methodology tailored for large-scale evaluation of telerobotic systems
- A taxonomy of mobile multiagent tasks

1.5 Dissertation Structure

- Chapter 1 is this introduction. It presents the motivations for conducting this research, as well as the research questions that guided it.
- Chapter 2 discusses research related to this thesis. Multiagent robotic work is discussed first, and then telerobotic research. Since these two fields are well studied, only closely related multiagent and telerobotic work is described. Next, research in multiagent telerobotics is explored. Since very few projects have concentrated on multiagent telerobotics, even those systems only remotely related to the present work are discussed. The last section of Chapter 2 presents related experimental evaluation of telerobotic systems.
- Chapter 3 presents the methodology for conducting telerobotic system evaluations. This methodology is presented so other researchers can easily use it to compare their own telerobotic systems.
- The systems compared in the experimental evaluations are described in Chapter 4. This chapter presents the dimensions of MTSs and the corresponding examined levels along those dimensions. Representative systems for each of these classifications are described. Lastly, the discussion turns to the underlying robot architecture for the system types.

- In Chapter 5, the taxonomy of mobile multiagent tasks is explained. Then, the experimental tasks used to represent each taxonomic category are described.

- Chapter 6 reviews the experimental design in terms of factors, treatments, replications, responses, and subjects (these terms are explained in that section). This is followed by a detailed description of the procedures followed during the experiments.

- The techniques used to analyze the data are explained in Chapter 7. Brief details of each of these techniques are given. These are standard experimental analysis methods, and more details can be found in most statistics textbooks, such as [41].

- Chapter 8 reports the results of the evaluations. Additionally, the importance and generalizability of these results are described.

- Chapter 9 relates a predictive study conducted after the initial evaluations. This study demonstrated that the results presented in Chapter 8 could be used to make effective predictions.

- The contributions of this research are set forth in Chapter 10.

- The appendices describe lower level details of this work. Appendix A serves as a glossary for this dissertation. Appendix B presents the details of the motor schemas and assemblages that were used in the robotic control systems. Appendix C reports the data values collected during the experiments. The confidence intervals and other important values derived from the analysis of the data are presented in Appendix D. Appendix E relates the distribution of the types of human subjects used in the experiments. Finally, Appendix

F presents the forms (consent form, survey, etc.) that were used during the experiments.

Chapter 2

Related Work

Multiagent robotics and telerobotics are both areas in which a large amount of research has been previously conducted. Little work, however, has been done in the combined area of multiagent telerobotics. Section 2.1 provides an overview of the work in multiagent robotics, and Section 2.2 provides an overview of the telerobotics research. After that, the few multiagent telerobotics projects are discussed. Finally, research in the experimental evaluation of telerobotic systems is described. All of this work is examined regarding its relation to this thesis.

2.1 Multiagent Robotics

Most research to date in multiagent robotics has focused on either group architecture, resource conflicts, origins of cooperation, learning, or geometric problems [16]. Efforts in group architecture have examined issues such as centralization/decentralization, heterogeneity/homogeneity, communication structures, and modeling of other agents [16]. Work in resource conflicts has studied ways to share space, tools, and communication media. Studying the origins of cooperation involves examining ways that cooperation results without being explicitly implemented in the system, both in naturally occurring systems such as animal societies [39], and in robot systems [5]. Geometric problems include the study of multiple-robot path planning and formation and coverage behaviors [13, 23]. In studying how many robots an operator should control at one time, the research conducted in this thesis deals with the area of group architectures, but from a different perspective, one that involves a human in the loop. Some well-known examples of multiagent robot systems include ACTRESS [10], CEBOT [22], and the work of Mataric [35].

CEBOT [22] was among the first mobile multiagent systems [6]. It was a "cellular robotic system", with small robots that could dock together to produce a larger robot. The robots communicated positional information to each other. The CEBOT research now uses an architecture that has multiple parallel behaviors which are integrated by vector summation [6].

ACTRESS (ACTor-based Robots and Equipments Synthesis System) [10], is a multirobot system designed for heterogeneous agents, which focuses on communications issues. Normally, the robots act independently, but if the need arises, they "negotiate" with other robots to form a cooperative group to handle the problem.

Mataric [36] has created behaviors for multiagent systems using a subsumption style architecture [15]. She created homing, aggregation, dispersion, following, and wandering behaviors, and used them for a foraging task [6].

Rather than giving a detailed discussion of many of the well-known examples of multiagent robotics, the following sections discuss only the work that is most closely related to this research. Arkin's schema-based approach [3] (Section 2.1.1) is an architecture that has been used for multiagent robotics, while Gage's work [23] (Section 2.1.2) and Balch and Arkin's formations [13] focus on methods for coordinating the movement of the robot group as a whole. The following discussion focuses on these systems.

2.1.1 Schema-Based Multiagent Robotics

Arkin's schema-based approach to behavioral robotics [5, 12, 13] has been used in a significant number of multiagent robotics systems. In the schema-based approach [3], each reactive behavior, or motor schema, tries to influence the behavior of the robot. Each schema produces a vector in the direction that the schema wants the robot to go and with a magnitude that reflects the importance of going in that direction. The vectors of all the active schemas are summed and normalized, and the resulting vector is sent to the robot for execution.

The schema-based approach has been used in multiagent robotics research examining issues related to communication between robots [5, 9, 8, 12], and movement in formation [13]. The robots in this thesis use a schema-based approach for reactive navigation and as the basis upon which teleoperation is built.

2.1.2 Gage's research in control of many-robot systems

Gage [23] looks at the command and control of a system of more than 100 robots for military-type missions. He is investigating ways to control the movement and positioning of the robot group as a whole, rather than by controlling the movement of individual agents. Each individual agent's motion is based on the motions of the other agents, and is most strongly influenced by its nearest neighbors. He focuses on coverage and formation behaviors as a means to accomplish this. A coverage behavior maintains a spatial relationship which adapts to local conditions to optimize some function, such as the detection rate/range of targets or the probability of undetected enemy penetration. A formation behavior is similar to a coverage behavior, except that the group maintains an explicitly specified spacing.

Gage looks at two ways to control group movement. The first is to bias the motion of each agent in the desired direction. The second is to directly control the movement of a small number of agents and let the others follow due to a coverage or formation behavior. In the research conducted in this thesis, an approach similar to Gage's first method is used to control the movement of the robots under supervisory control. The robots' motion is biased in the direction that the human operator specifies. The "operator as a behavior" approach, however, described in [4] actually served as the basis for the movement control used in this thesis. This technique is presented in Section 2.2.1.

2.2 Telerobotics

Telerobotics methods can be separated into three types: manual control, supervisory control, and fully automatic control [48]. In manual control, the human specifies all robot motion by continuous input. In supervisory control, robot motion is caused by either human input or computer generated input. In fully automatic control, all robot motion is specified by computer input. There are two primary subsets of supervisory control: supervised autonomy and shared control [48]. With shared control, the input from the human is sent during execution of a motion and merged with the closed-loop motion generated by the computer. In supervised autonomy, commands are generated through human interaction, but sent to the robot for autonomous execution. One of the most common reasons for using any form of supervisory control is to deal with time delay in teleoperation [47, 48, 55, 50]. Other common reasons include safety and ease-of-use [47].

A common strategy for developing telerobotic systems is to automate the lower-level functions while relying on humans to provide the overall guidance and to handle difficult situations [48, 47]. Another strategy for developing telerobotic systems is to automate the lower-level functions and as much of the higher-level functions as possible [24, 57, 54, 50]. This allows the robot to proceed with its task without any input from the human. The human can observe the progress, however, and intervene if he or she desires. Some of these systems also allow the robot to signal the operator when it needs assistance [57, 54, 50, 45]. Examples of each type of these approaches are given in the remainder of this section as well as the discussion of related Multiagent Telerobotics research.

This section is a discussion of single-agent telerobotics research that is closely related to this research, and mutliagent telerobotics work is presented in the next section. Table 2.1 shows how the described research fits in the telerobotic system dimension of the amount of influence that the robots have. Arkin [4], Graves [24], and Guo, *et al*, [25] all provide forms of shared control. In Arkin's (Section 2.2.1) and Graves's (Section 2.2.2) work, the amount of influence is easily altered. Guo (Section 2.2.4) provides the operator with a set amount of influence, allowing the human to influence, but not totally control the execution of the predetermined plan. Noreils [42] (Section 2.2.3) provides the operator with supervised autonomy, such that the operator has complete control at the planning level, but no control of the robot's actions at the execution level, although he proposes an interface which would allow minimal control at the reactive level.

Arkin and Guo allow directional commands to be given. Arkin also allows parameter modification, while Guo provides speed control. Noreils provides control in the form of task-related instructions given in his visual programming language.

2.2.1 Arkin's Telerobotics Approach

Arkin presents two methods [4] for teleoperation of a single agent using the schema-based reactive robotics architecture [3]. The first method is to treat the human operator as a schema. In this method, a teleoperator schema takes the desired gain and desired direction of movement as input. The teleoperator schema outputs a

Table 2.1: Locations of the related telerobotics systems in the *Amount of influence* dimension. Since these are single agent systems, the *Number of robots controlled at once* dimension is not applicable to them.

Designer	System	Amount of human influence	Reference
Arkin	Schema-based telerobotics	shared	[4]
Graves	ASIAGO	shared, influence changes dynamically	[24]
Noreils	Man/Machine interface	not specified	[42]
Guo	Function-based control sharing	shared	[25]

vector in the desired direction of movement with a magnitude relative to the size of the input gain. This vector is summed and normalized with the vectors from other active schemas and then passed to the robot for execution.

The second method is to treat the human operator as a supervisor. In this method, the operator adjusts the values of the gains and internal parameters of the active schemas. This changes the overall behavior of the robot. This type of control requires a deeper understanding of the schemas by the operator in order to be effective [4].

One of the robot control techniques used in this research is based on Arkin's work in single agent telerobotics. The underlying strategy used for directional control is the same as the first method described above, except that it has been generalized for multiple robots.

2.2.2 ASIAGO

Graves's ASIAGO (A System for Intelligent Action Generation for teleOperation) [24] is an action-selection mechanism for a telerobot. It selects actions by fusing control decisions using a blending approach from a variety of command sources, including human control. Figure 2.1 is a simplified illustration of the fusion technique. Each device on the telerobot has a current mode, which specifies how to do the blending. The *Integrator* for each device uses this mode to select an action. A mode consists of a blending matrix and an input map. The blending matrix is a matrix of weights that controls how much influence an input has on a degree of freedom of the device. The rows of the blending matrix represent individual inputs from various sources. The columns represent degrees of freedom of the device. The input map specifies which input sources map to which row in the blending matrix.

The current mode for a device may be changed during execution. *Event Recognizers* monitor the input for specific data patterns or conditions, and notify the *Mode Manager* when one is observed. The *Mode Manager* can then change the current mode for a device. The mode transition can either be immediate or "faded in", such that the mode is changed gradually, with the *Integrator* interpolating between the weights in the starting and finishing modes during the change.

This is similar to the control methods in the supervisory control configuration used for the research in this thesis, because it treats the human operator the same as any of the other input devices. The operator's input is combined with the input from the other sources to produce the action to execute.

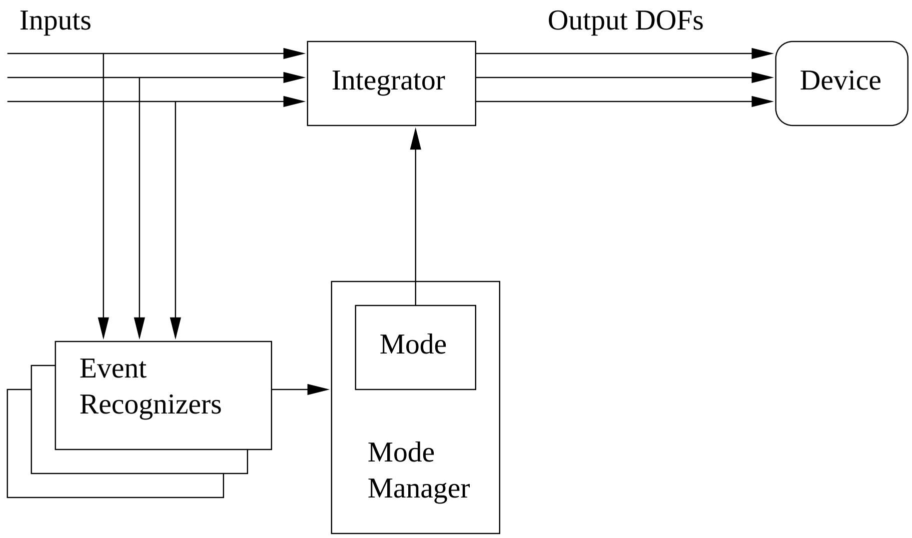


Figure 2.1: The fusion technique in ASIAGO. The operator provides one of the inputs. See the text for details.

2.2.3 Noreils's Man/Machine Interface

Noreils [42] emphasizes the need for both reflexive and planning capabilities for a mobile robot. He states that reactivity is dependent on the task, and that the configuration of the reactive component must be controlled by the planning component. He further emphasizes the need for a human in the decision process to guide the planning for the robot. Noreils feels that the man/machine interface should support the generation of meta-plans (which are high-level plans composed of lower-level missions), the creation of visual missions, which are plan steps created with a visual programming language, and altering information to be sent back to the reactive level (or functional level) of the robot.

Noreils notes that including a human in the decision process raises many questions (which are relevant to this thesis research), such as: the level of interaction between the operator and the robot; what type of information is relevant at the man/machine interface (MMI) level; and the nature of the interaction between the operator and the interface [43]. This dissertation's results provide guidelines for answering some of these questions for multiagent telerobotic systems in terms of the task.

2.2.4 Guo's Function-Based Control Sharing

Guo, *et al* [25], presents a method for combining the input from a human operator with the input from an event-based planner and applying it to single arm teleoperation. The goal is to develop a planning/control scheme such that the input from a human operator can be easily integrated without significantly disturbing its autonomous operation. When the operator is finished, the system resumes autonomous operation without any replanning.

The planning system is event-based and uses action reference parameters [25] rather than time. Instead of planning where the arm should be at a certain time, the planner creates parameters that specify goal locations where the arm should move to. When the arm achieves that goal, it proceeds to the next step in the plan. The human operator is allowed to influence the movement of the arm in a specified set of dimensions relative to the direction that the autonomous planner is moving it in: by stopping the arm, slowing down the arm, speeding up the arm, or inputing a force on the arm orthogonal to the direction the planner is moving it in. When the operator influences the arm in an orthogonal direction, it moves in a direction that is the sum of the velocity vectors of the planner and the operator. This is similar to the supervisory control component of this thesis's experimental testbed in that the operator's input is combined with the input from the robot's autonomous controller using vector addition to produce the action to execute.

Table 2.2: Related multiagent telerobotics systems. The table shows the MTSs in respect to the dimensions examined in this thesis.

Designer	System	Amount of human influence	Number of robots controlled at once	Reference
Asama, Yokota, *et al*	Interface for ACTRESS	strict supervisory control	individual, subgroups, entire group	[57]
Nakauchi	RT-Michele	not specified	individual, subgroups, entire group	[40]
Adams	MASC	strict supervisory control	individual	[1]
Dickson	AUTOMAN	strict supervisory control	entire group	[18]
Ohkawa	Ohkawa's work	supervisory control	NA	[44]
Ishikawa	Ishikawa's work	supervisory and direct manual control	individual	[28]

2.3 Multiagent Telerobotics

The following sections contain descriptions of the research conducted in multiagent telerobotics. Most of it differs from this research but is included for completeness. Table 2.2 denotes where these examples fit in the examined multiagent telerobotic system (MTS) dimensions.

ACTRESS [57], MASC [1], and AUTOMAN [18] are supervised autonomy systems. These systems range from MASC, which is almost autonomous, to AUTOMAN, which allows the user to specify the desired locations and orientations for manipulating an object, but not how to move between these locations.

MASC and Ishikawa's work [28] allow the user to control only one robot at a time, while AUTOMAN requires the user to instruct all the robots at once. ACTRESS and RT-Michele allow the operator to instruct one or more robots at a time.

2.3.1 Human Interface System for ACTRESS

ACTRESS (ACTor-based Robots and Equipment Synthesis System) [10] is a multi-agent robot architecture. A human interface system has been developed for ACTRESS [57, 54] that allows the human operator to command and monitor the status of the robots, and provides the robots with a means to contact the operator. The operator can give task-related commands, such as "push box" or "retreat", to the robots, by manipulating on-screen mechanisms, such as menus. The robots themselves coordinate how the tasks are carried out. The operator can direct one robot or a group of robots at a time to do these tasks.

The primary focus is to provide the operator with monitoring capabilities for the robot group, without requiring him to look at each robot individually. The interface provides a means for the operator to determine the status of individual robots, groups of robots, and the entire system. Status information can be gathered either by explicit or implicit communication with the robots [54]. With explicit communication, the agents are asked directly to give information about themselves. With implicit communication, the system gathers status information by eavesdropping on the messages passed between the agents.

Simulation tests were conducted to determine the communication load and reliability of information with the different communication strategies. The task required the robots to move from one point to another, while executing numerous turns. The following four monitoring methods were examined:

1. The operator queries each agent at fixed intervals.
2. Each agent reports to the operator at fixed intervals.
3. The agents report when they change direction.
4. The operator eavesdrops on the messages between agents.

Types 1-3 are explicit strategies, while type 4 is implicit. It was found that the explicit monitoring strategies are more reliable but place a higher communication load on the system. The implicit strategy did not increase this load, but it did not provide reliable information about the state of the system. Strategy 3 was found to be the best compromise between reliability and minimizing communication load [54].

This sort of evaluation is important to determine what types of MTSs perform best. Experiments conducted on real robots would have provided more reliable information, and additional tasks should be analyzed in the future. These tests, however, involving multiple systems, are necessary to provide future MTS developers with a basis to build upon.

More recently, this group has concentrated on methods for an operator to give instructions to a group of telerobots [52]. They identify four issues that a multiagent telerobotic system should address:

- Coordination of the robots
- Commanding level
- Operationality
- Cooperation strategies among the robots

Coordination means how many and which robots are controlled by the operator at a given time. They state that an MTS should allow the operator to command either individual robots, the entire group, or subgroups of the robots. Also, the operator should be able to choose which robots should be included in an action, or the human interface or the robots themselves may determine the allocation of the robots based on the operator's choice of coordination [52]. This thesis examines which of these sorts of coordination methods is most appropriate for which type of task.

The commanding level can be one of three levels [52]:

- Task level
- Action level
- Direct control level

The task level includes abstract commands such as "Execute Task A". The action level includes commands such as "Move straight 1m" or "Go to the position (x,y)". Direct control is the level at which the operator can control the robots' actuators and devices directly. One of the dimensions that is examined in this thesis, the level of autonomy of the robots, is related to the issue of the commanding level. The more autonomous the robots are, the higher the commanding level possible.

Their "principle of operationality" states that the operator should have several means of input for controlling the robots, such as buttons, menus, and command line input [53]. Regarding cooperation strategy among the robots, they say that the human interface and the robots should determine the formation of the robots based on the requirements of the human's task [52]. This acknowledgement of the task to the interface design is a central point of this dissertation.

Concentrating on one of these issues, Ishikawa *et al* [28], have developed a graphical user interface (GUI) for the ACTRESS system to allow an operator to choose which robot he wishes to control. The operator can choose a single robot to operate under direct manual control, while the other robots operate autonomously. The operator can easily change the control to another robot using a set of buttons, or he can allow all the robots to run autonomously.

2.3.2 RT-Michele

RT-Michele [40] is a multiagent interface architecture to support cooperative work among multiple humans and multiple robots. The system provides a protocol for allowing communication between any number of robots and humans. It classifies communication by whether it is synchronous or asynchronous, and by whether it is electronic or physical. Physical communication includes the transfer of some physical object to or from one agent to another. When a human or robot wants to communicate with another agent, it creates what is called a meeting-environment, which may or may not be an actual physical space. Then it asks those it wishes to communicate with to join the meeting. When the other agents "arrive" at the meeting-environment, they then communicate. This work is not concerned with how to communicate with and instruct the robots once all the parties are present at the meeting. The important thing seems to be the protocol for setting up these meetings.

2.3.3 MASC

MASC (Multiple Agent Supervisory Control) [1] is a supervisory control system for multiple robots that permits the human supervisor to interact at various levels in the perceptual processing. The human allows the robots to work autonomously, and only interferes when they are unable to carry out their task. The operator is provided with a display of the sensory input to a robot at all levels of abstraction, from the raw sensor data to processed data. The human can correct corrupted data or process decisions which could cause the robot to enter an incorrect state. The operator can inspect and correct data for individual robots, but not the group as a whole. In addition, a robot may ask for assistance from the human operator.

The operator can only work with one robot at a time. Since the operator's job is to monitor for incorrect data, then he must constantly or regularly inspect the data for each robot, repeatedly switching between robots. This type of constant monitoring of multiple data sets seems likely to cause cognitive overload for the supervisor.

2.3.4 AUTOMAN

AUTOMAN (AUtonomous robot-Team Object MANipulation) [18] is a control hierarchy for object-based task-level control of a team of robots. It is used for a team of two free-flying robots for the task of manipulating a free-flying object. AUTOMAN consists of three levels: User Interface, Strategic Control, and Dynamic Control. The Dynamic Control level first computes the desired accelerations and internal forces for the object to be manipulated. Then it computes the necessary external forces and accelerations on the object. These are passed to the robots' controllers. The Strategic Control level steps through the subtasks of a complex task (such as docking), making appropriate decisions and commanding the Dynamic Control level. The User Interface is a graphical iconic display of the object and workspace that allows the human operator to specify object-based task-level commands. The user uses a mouse to indicate a desired object to capture, transport, dock, or release, and the locations to do each. The object-oriented style of human control probably makes the robot group easier to control, since the operator does not need to focus on the actions of each robot. Evaluations should be conducted comparing this control style to other more conventional methods.

2.3.5 Ohkawa's *et al* Control Through Rewards

Ohkawa *et al* [44] have developed an interesting method for a human operator to control a group of robots. The method is meant to be used when the robots' task can be divided into subtasks, where each subtask can be completed by one of the robots. In this system, each robot selects which subtask it wants to do. The human then assigns rewards based on the selections that the robots make. The robots evaluate their previous choices based on these rewards, using Q-learning, and change their behavior. Thus, the human "controls" the robots by causing them to do reinforcement learning to improve their method for selecting subtasks. While this is an interesting technique, the operator's control of the robots, and the tasks for which it is appropriate, are rather limited.

2.4 Experimental Evaluation of Telerobotic Systems

Very little work [7, 54] analyzing the performance of multiagent telerobotic systems was encountered. Some research [12, 2] has considered performance analysis for multiagent (non-telerobotic) systems. The literature on telerobotic system evaluation, however, is more closely related to this research than the literature on the the evaluation of multiagent systems, and so the former is discussed here.

Most of the work on telerobotic system evaluation has focused on how well the human/machine system copes with time delay. Some researchers have considered other issues, such as the nature of the feedback to the operator [20] and the form of operator interaction with the interface [19].

None of the experimental evaluations of robotic systems that use real robots, however, have been conducted on as large a scale (in terms of the number of trials and participants) as the experiments conducted in this dissertation. While performance evaluations of this scale on real robots are very time consuming, they are crucial to advance the state of robotics to a science. Small scale evaluations do not provide good statistical guarantees of correctness, and studies in simulation cannot be guaranteed to produce results that are valid in the real world.

2.4.1 Hannaford

Hannaford presents performance measures for evaluating telerobotic systems, such as completion time, force performance, and error rate [26]. He advises conducting task segmentation analysis. This involves computing performance measures separately for the different phases of a task, because often different kinds of performance are important for the difference phases. He separates the tasks used for teleoperator evaluation into two classes: generic tasks and application tasks. Generic tasks are idealized, simplified tasks that are intended to test specific capabilities, while application tasks are designed to resemble real-world tasks as much as possible. The experiments conducted for this dissertation used generic tasks to focus on the specific task classes.

2.4.2 Bejczy

Bejczy notes that the training cycle greatly effects the performance of the operator. He suggests the following system for training operators for telerobotic evaluation experiments [14]. The first cycle should be used to familiarize the operator with the system and task, and for a novice operator, this cycle should be repeated at least twice. During the second training cycle, performance measurements are made so that the operator understands the measures against which the performance will be evaluated. The real performance evaluations can then be conducted. Each cycle and its repetitions should be separated by at least one day.

Unfortunately, training cycles of this length were not possible in the experiments conducted in this thesis. The operator, however, was informed of the performance measures, as Bejczy suggests. The training cycle that was used tried to insure that all participants had the same amount of prior experience with the system, and is described in Section 6.4.

2.4.3 Skubic

Skubic integrated a system for performance analysis into his Telerobotics Construction Set (TCS) [49]. The performance analysis system is based on the General Systems Performance Theory (GSPT) [30]. This allows comparison of the performance of various subsystems that are included in telerobotics systems constructed with TCS. The performance analysis of this dissertation's research is different from Skubic's analysis, in that Skubic was comparing different modules of one telerobotic system to each other, while this research focuses on MTSs as a whole when compared for different tasks.

2.4.4 Draper

Draper and others [19, 20, 31] at Oak Ridge National Laboratories have done experiments analyzing the performance of various manipulator systems under different conditions. One experiment [19] investigated options for camera controls on manipulator systems. The performance of two different systems was compared for a manipulation task in which a camera provided feedback. In the first system, the camera was controlled by conventional manual controls. For the second system, a combination of voice input and automation was used to control the camera. The results indicated that the manual control system had longer task completion times, yet required fewer camera position changes.

In another set of experiments [20], they compared the performance of three brands (Meidensha BILARM 83A, Central Research Laboratories Model M-2, and GCA PaR Systems Model 6000) of manipulator systems under differing forms of control. In particular, these experiments compared the differences between master/slave systems with and without force reflection, as well as the difference between master/slave systems and switchbox-controlled systems on three brands of manipulator systems [20]. They found no significant difference between the M-2 with and without force-reflection. The BILARM completed tasks faster without force-reflection, but produced more errors in this mode. Master/slave systems had lower task completion times than switchbox controlled systems, without a significant change in the error rate.

Another study [31] examined the performance of three manipulator systems (Central Research Laboratory's Model M-2, an advanced servomanipulator (ASM), and a Meidensha Prototype-2 (P-2)) as part of their in-cell maintenance systems for use in future nuclear fuel reprocessing facilities. The evaluation was based on the completion time for the task. The P-2 performed the worst, and the times for the M-2 and ASM were not significantly different from each other.

These experiments are similar in nature to the ones conducted in this thesis. Each of Draper's experiments, however, studied human control of manipulators rather than mobile robots. These evaluations were conducted on a smaller scale than this dissertation's experiments, but they are important nonetheless. This sort of formal evaluation is necessary to provide future developers with developmental guidelines.

2.4.5 Kirlik

Kirlik [29] compared the performance of two telerobotic systems that differed by dividing the operator's responsibility between different numbers of operators for a reconnaissance task. The task involved piloting a simulated helicopter and supervising four other simulated helicopters, while searching for various objects in the task environment. One system put all of the observation and control responsibilities on a single operator, while the other divided them between two operators. Novice users tried both systems, and their performace was compared with that of an expert one-person crew. The expert one-person crew and novice two-person crews performed comparably, and both performed better than novice one-person crews [29]. This evaluation differs from this dissertation's experiments, since it is more an evaluation of the effects of the number of operators on performance than the effects of differing system types.

Chapter 3

Methodology

There is no formal method for evaluating robot systems currently in widespread use. Typically, robot systems are evaluated by using a proof-of-concept technique, in which the system is shown to be capable of accomplishing a particular task. Sometimes, two different robot systems are compared against each other for a given task [19, 20]. Even with these tests, however, a more formal approach should be taken to choose which types of systems to compare and for what type of task. One notable exception, in which a formal approach is used for robot evaluation, is the work by Sukhatme [51], which concentrates on a method for deriving evaluation criteria for robots traversing rocky terrain and compares robots that differ in their physical characteristics.

Large-scale evaluations of different types of systems are necessary to advance the fields of robotic and telerobotics to a more scientific stage. A methodology that suits this sort of large-scale testing was used for this research. This approach combines elements from multiple fields, borrowing heavily from user-interface evaluation and human-factors studies. This section describes how to apply this methodology to robotics studies. The typical methodology for comparison of systems used in other fields such as user-interface evaluation has been changed to suit the nature of robotics research. The primary changes involve different approaches to choosing the types of systems and tasks to examine. A more formal method is presented for selecting the system and task types than the task analysis and functional analysis [27] typically used in user-interface evaluations, or the typically ad-hoc methods of robotics.

The following sections present step-by-step instructions for conducting evaluations of different types of telerobotic systems. While the experiments conducted as part of this research using this methodology compare multiagent mobile telerobotic systems, the approach is not specific to multiagent or mobile robot systems. It can be used just as easily for single-agent systems and manipulator arms. Furthermore, with a little modification, this procedure could be used for evaluating autonomous robot systems.

> In the remainder of this section, offset in text boxes, an example application presents an evaluation of possible Mars rover systems.

The overall procedure for comparing telerobotic systems can be broken into the following basic steps, each described in more detail in the following sections.

1. Determine the evaluation criteria.
2. Decide what types of systems to compare.
3. Formulate tasks for the systems.
4. Conduct experiments with real robots to compare the systems for the tasks.
5. Analyze the data collected from the experiments to determine how each system performed.

3.1 Determine Evaluation Criteria

The evaluation criteria provide a formal method for determining what data to collect during the experiments and how to use them to compare the systems. This step can be broken into three substeps:

1. Determine the criteria.

2. Decide what data/events represent those criteria.

3. Decide how to combine the data into a single quantitative value.

The first step for evaluating telerobot systems is to determine what sort of criteria to evaluate them by. For instance, is safety important, or should the systems be judged based on their effectiveness, ease of use, cost, or some other measure? This decision depends on what is important to the developer.

> **Evaluation Criteria:** For the Mars rover example, we will compare the systems by their power efficiency and safety. Safety is important to Mars rovers, because if the robot is damaged, it is difficult and often impossible to repair it. The greater the power efficiency, the fewer batteries that are required to power the robot. Since batteries are heavy, they contribute greatly to the cost of transporting the robot to Mars, which depends largely on the weight of the robot. Therefore, greater power efficiency means lower transportation costs.

Once the evaluation criteria are determined, methods for measuring them must be determined. In other words, for each criteria, it must be determined which events or other data can be collected during an experimental evaluation to determines how well a system meets that criterion. For instance, if one measure is effectiveness, is task completion time important or the distance the robots travel, or how well the robots accomplish the task, or some combination of multiple kinds of data? The types of data may be different based on the needs of the experimenter. In all cases, however, the metrics should be objective and measurable.

Events Representing the Criteria: To determine the safety in our example, we can count the number of collisions between all robots and obstacles and the number of times the robots unintentionally slip into pits. The power efficiency can be determined by summing the voltage change of the robots' batteries between the start and completion of the task.

Next, the designer must determine how the data for each judgment criteria will translate to a single quantitative representation of that criteria, so that different systems can be compared against each other. In some instances, this may be very simple. For example, if only one type of data, such as the number of collisions, is being used to determine the safety of the systems, then no transformation is necessary, and systems with fewer collisions can be considered safer.

A Single Quantitative Value: In the example, the voltage change from all batteries is the only measure used to determine power efficiency, so lower voltage changes directly indicate more efficient systems.

If, however, multiple data types are used to determine one criteria, a means for combining the data into a single quantitative measure must be determined. For example, if both task completion time and the distance traveled are used to determine the effectiveness of the systems, then there must be some way to combine the two measures into a single value. If the two measures share the same units, then they could just be added together (or one multiplied/divided by the other), or, more likely, it might be appropriate to weight and/or scale them before combining, since they might not be of the same importance or on the same scale. If, the measures use different and non-compatible units, such as seconds and feet in the previous example, it will be necessary to convert them to some dimensionless unit first. This conversion could be one-to-one, or it might involve scaling and weighting, depending on the units involved and their relative importance in the eyes of the experimenter.

Furthermore, high values might indicate good systems using one measure, while low values indicate good systems with another measure, making it necessary to invert and scale one of the metrics.

> **A Single Quantitative Value:** For Mars rovers, falling into holes is probably more dangerous (since the robot might not be able to climb back out of the hole) than colliding with obstacles, so we will determine the safety measure as follows:
>
> $$safety = (number\_of\_collisions) + 2 * (number\_of\_falls)$$
>
> Here, lower values represent safer systems. Ideally, this formulation should be based on a more reliable analysis of the relative importance of collisions and falls, but this analysis is sufficient for the example.

The appropriate method for combining multiple data types can only be determined by the experimenter. The important point is that whatever methods are used, they should produce a single quantitative value to represent each of the chosen criteria.

In summary, there are three steps in determining how to compare systems. First, criteria to judge the systems are determined. The experimenter then decides which measurable events and data best reflects that criteria for his purposes. Finally, if there are multiple data types for any single criterion, then a formula must be determined to combine the data into a single quantitative measure.

3.2 Determine Which Systems to Examine

The next step is to determine what types of systems to compare. If the purpose of the evaluation is to compare a set of already existing systems, then this step is complete. When the purpose is not to compare existing systems, however, but instead to determine the best type of system in general, a more formal approach is appropriate. In this case, these steps should be followed:

1. Decide the system dimensions.

2. Determine points along those dimensions.

3. If multiple dimensions are used, then cross[1] the dimensions to produce the system types.

4. Create representative systems for each system type.

First, decide on the system dimensions to be examined. There are many dimensions in which telerobot systems may differ, *e.g.*, homogeneity/heterogeneity, the amount of autonomy the robots have, the underlying control architecture, or the number of robots present. The ways in which telerobot systems may differ is potentially unlimited. Choose an appropriate number of these dimensions to examine, based on what is important to the nature of tasks to be examined. The number of chosen dimensions should preferably be small (one to three), because the number of system types to be tested increases exponentially with the number of dimensions. If a finite number of human subjects are available for testing, greater numbers of system types (due to more dimensions) reduce the number of subjects used for each system, which provides less of a statistical guarantee for the results of the experiments.

[1]Factors (or dimensions) are said to be crossed when every level of one factor appears with each level of every other factor [41].

System Dimensions: For the Mars rover example, we will consider two dimensions: the number of robots in the system and whether or not the robots are holonomic. Holonomic vehicles can move in any direction at any time, *i.e.*, they can start off in any direction and can change directions instantaneously. Greater numbers of robots might cost more to transport, but they might require less power overall. The system should also be more robust, since if one robot breaks, the others can continue working. Holonomic vehicles seem more likely to be safer, as they can maneuver better, but it is possible that the operator may be able to avoid dangerous situations without the extra maneuverability.

Next, for each of the chosen dimensions, determine at which points to examine them. For example, if the number of robots in the system is being evaluated, choose a few different points on this dimension, such as 1, 5, and 10 robots, or maybe 1, 2, 3, 4, and 5 robots. The actual choice of points must be based on the judgment of the system designer, but the following suggestions may be helpful. If only a few points can be examined due to time limitations on the experiments, then it is often useful to examine points that differ significantly, such as the extreme values for the dimension (if such extremes exist), to gain some understanding (possibly incorrect) of the range of possible results. At other times it may be more appropriate to examine points commonly used in existing systems. For example, if the type of underlying control system is being varied, then instances of well known control architectures might be used.

Points Along the System Dimensions: For the number of rovers, we will examine 1, 2, and 5 robots. One robot is the minimum value, as well as the commonly chosen point. Using two robots will indicate if any benefits can be gained by multiple robots, while still keeping low the number of robots that need to be transported to Mars. Examining five robots will provide some indication of how any performance change due to multiple robots scales as more robots are added. For the second dimension, the two points are non-holonomic vehicles and kinematically holonomic vehicles.

If multiple dimensions are being examined, then crossing them yields the different system types to be compared. Crossing the dimensions is effectively like taking their cross-product, thereby producing every possibly combination of the points along the different dimensions. In other words, if dimension A has points a_1 and a_2 and B has b_1 and b_2, then crossing dimensions A and B will yield points a_1b_1, a_1b_2, a_2b_1, and a_2b_2. So, if there are two dimensions, with n_1 points in the first and n_2 in the second, then there are $n_1 \times n_2$ different system types to compare.

Cross the Dimensions: There are six system types to examine in our Mars rover example, derived by crossing the two dimensions as shown in the following table:

		Holonomicity	
		Holonomic	Non-holonomic
Number of robots	1	H1	N1
	2	H2	N2
	5	H5	N5

This notation (H1, N1, etc.), will be used in the remainder of this example to denote the system types, *e.g.,* H1 indicates a system with one holonomic robot.

The designer must create a system or systems to represent the system types. Either a separate system must be developed for each type, or one flexible architecture can be created that is reconfigurable to represent each of the system types. The human-interfaces for the systems should be as easy to use as possible, to minimize the effects of the interface implementation on the experimental results. It is impossible to prevent the implementation of the system interfaces from affecting the results, just as it is impossible to know what type of human-interface is easiest to use. By following acknowledged human-computer interface principles, however, this effect can be minimized. Some examples of these principles, which are described in more detail in [27] and other human-factors textbooks, include:

- User-centered design: Produce what is best for the user, rather than what is easiest to implement.

- Providing a system model: Give the user a mental model regarding the functionality and sequencing of the system.
- Consistency: Similar things should be presented or done in similar ways.
- Simplicity: Break complex tasks into simpler subtasks.
- Feedback: Let the user know what happened when he/she performs an action.

User-interface studies can be used to improve the different possible interfaces for each system type. This is an iterative process in which human users try various interface styles for one or more tasks and performance measurements are taken. The user-interfaces are then refined based on the results to improve their ease-of-use.

In summary, there are four steps to determine which systems to compare. A reasonable number of dimensions along which the systems may differ should be determined first. Next, the designer should select a subset of points along those dimensions to examine. The systems to be considered include all of the possible combinations of the points in the dimensions, which are produced by crossing the dimensions. Finally, representative systems (or one reconfigurable system) for each of the system types need to be developed.

3.3 Establish Tasks for Comparing the Systems

Specific tasks should then be determined for the evaluations. If there is a particular set of real life tasks that are of interest to the experimenter, then these can be used. If, on the other hand, the designer does not have specific tasks in mind[2], then either

[2]This could be either because the experimenter is doing general research on systems and tasks, such as was done in this dissertation, or because the designer produced a solution and is now searching for a problem for it to solve, which, unfortunately, is not that uncommon of an approach to research.

commonly used telerobot tasks (such as inserting pegs into holes) can be chosen or a more formal approach can be taken to determine the tasks. The following discussion describes just such a formal approach. Note that this method is not necessary if the experimenter has specific tasks in mind that he wants to examine.

First, determine a taxonomy of robot tasks to use. Either use an existing taxonomy or create one. Not many taxonomies exist for robot tasks, and therefore it will probably be necessary to create a taxonomy of task types. The taxonomy should classify the tasks based on the qualities of the tasks that are important to the experimenter. For instance, if the systems being evaluated are multiagent mobile robotic systems, as in this research, then a taxonomy classifying the movement of the robots relative to each other might be most appropriate. A taxonomy of this sort is presented in Section 5.1, and can be used if mobile multiagent robot systems are being evaluated.

A subset (possibly the entire set if it is small) of the taxonomy should be chosen for examination in the evaluations. For each of these task categories in this subset, an experimental task must be chosen to represent the category. This is because the systems cannot be used for a task category, but must be used for an actual task. There are two types of experimental tasks that can be used: application and generic tasks [26]. Application tasks are designed to imitate real world tasks as closely as possible, while generic tasks are idealized and simplified to test specific capabilities and are usually subtasks of real world tasks. Repairing a broken machine, foraging, and scouting a potential battlezone are examples of application tasks, since they are entire applications that someone might actually want a telerobot to perform. Inserting a peg in a hole, moving from point A to point B, and climbing stairs are generic tasks, since they are not tasks that are useful in isolation, but are performed as part of other applications, and test a specific capability of the telerobot/human team. Generic tasks are more appropriate for evaluating the task classifications of a taxonomy, because application tasks are more likely to involve multiple task classifications, one for each subtask. Generic tasks, on the other hand, can be designed to fit only the desired classification.

In summary, if the experimenter has particular tasks that he wants to examine, those tasks can be used in the evaluations. Commonly used tasks from telerobotic research can be used if the designer wants to compare his system to others for a recognized benchmark task. If, however, a more general evaluation is desired, then the experimental tasks should be chosen to represent classes of a task taxonomy.

> **Experimental Tasks:** In our rover example, we will assume that there are two primary tasks that the system will be used for, and, therefore, the experimenter wants to examine these particular tasks. The first task is gathering Martian rock samples and depositing them at a base station. In the experimental task, the task is completed when the robots have gathered 20 rocks. The robots are initially at the base station, and the rocks are scattered around the environment. Additionally, there are small pits in the environment that the robots will have to avoid. The second task involves moving the base station. In the experimental task, the robots must move to the base station, and then push it through an obstacle field to a designated location.

3.4 Conduct Experiments

One experiment should be conducted for each of the task types being examined. Each experiment evaluates all of the different system types for one of the tasks. A *factor* is a predictor, or independent, variable to be studied in an investigation. For instance, in an investigation of the effect of price on product sales, price is the examined factor. A factor level is a particular value for that factor. If price is the factor, then \$50 would be one of many possible factor levels. In our situation, the factors are the system dimensions and the factor levels are the points along those dimensions. An experimental *treatment* is a combination of one level from each factor in an experiment. So, if one factor being examined is price (with levels \$50 and \$100) and another is size (with levels small and large), then each of the four combinations of price and size (\$50/small, \$50/large, \$100/small, and \$100/large) is a treatment. In a one-factor experiment, the factor levels are the treatments. The individual system types, produced by crossing the system dimensions, are the treatments for these experiments.

> **Factors, Factor Levels, and Treatments:** Since there are two tasks to be examined (gathering Martian rocks and moving the base station), two experiments will be conducted. Each experiment will compare the six systems for each task. The factors, factor levels, and corresponding treatments for both experiments are shown in the table on page 44.
> Each experiment has two factors (corresponding to the system dimensions), the number of robots and whether the robots are holonomic or not. The factor levels correspond to the examined points along the dimensions: 1,2, and 5 for the number of robots dimension, and holonomic and non-holonomic for the holonomicity dimension. The experimental treatments correspond to the possible combinations of levels: H1, H2, H5, N1, N2, N5.

A *replication* is a repeat trial for the same treatment, typically with different experimental units (human subjects in this case). For each of the treatments, the same number of human subjects should be used. The number of human subjects used for each treatment is the number of replications of the experiment. Unfortunately, it is not possible to know in advance how many replications should be conducted to guarantee that statistically significant differences will be found. This can be estimated by means of the power approach or estimation approach[3] [41], but both of these approaches require a good estimate of the sample variance, which, put simply, is the amount that the data values from each human subject differ. It is impossible to know what sort of sample variance to expect unless very similar experiments have been conducted before. If a prior experiment is being validated, then its variance can be used to estimate the number of replications to conduct. Since experiments of the sort conducted in this dissertation have not been conducted previously, it is not possible to know how many replications are needed to provide statistical guarantees. In this case, there are two reasonable courses of action. The first is to conduct as many replications as time permits. The greater the number of replications, the less likely it will be that two types of systems will be found to be statistically equivalent when they are not. The second strategy is to conduct a small number of initial replications, and then use the variance from this experiment to estimate the number of replications needed. This strategy is more time consuming, but provides better statistical guarantees.

The treatments should be tested with human participants and real robots. To prevent any bias due to the participant gaining experience or a preference to one particular system, each participant should be used in only one replication and for one treatment. Therefore, the number of participants needed will be equal to the product of the number of treatments and the number of replications.

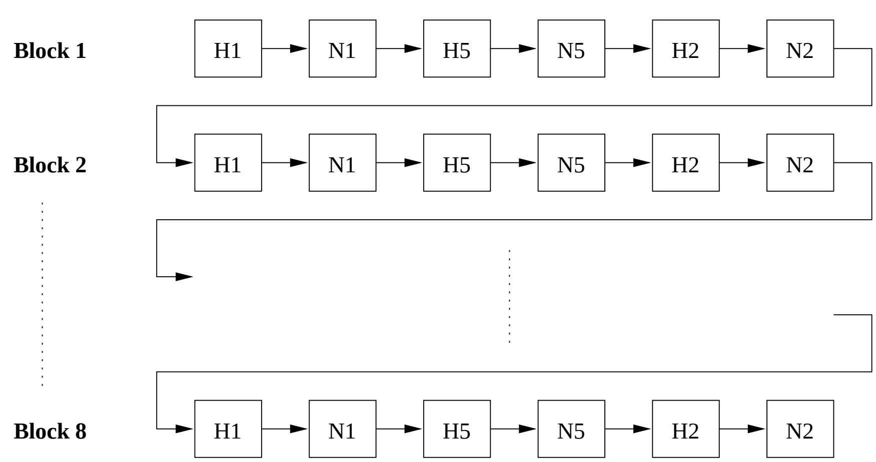
[3] A detailed explanation of these approaches can be found in most experimental design texts, and is beyond the scope of this dissertation.

Replications: For the rover example, we will conduct 8 replications. Since there are six treatments, this means that 48 human subjects are needed. Unless the variance of the data is high, this probably should provide enough replications to notice any differences between the performance of the systems, while still keeping the total number of participants tractable.

An experimental block is a grouping of experimental treatments, with every treatment occurring once. In temporal blocking, which is used in this thesis, one replication is conducted on all treatments in the block before the next replication is begun on any treatment. The number of replications of the experiment corresponds directly to the number of blocks. A block is simply a grouping of the treatments, specifying the order in which the treatments are examined. For these evaluations, the systems should be examined in blocks consisting of the set of all the system types. That is, one participant should attempt the task with the first system, then another participant should use the second system, and so on, until all of the systems have been used, as opposed to testing one system multiple times and then another system multiple times, and so forth. This comprises the first block, and then, the second block is begun. The figure in the boxed text of the example shows a blocking example.

The purpose of conducting the experiments in blocks is to prevent any bias towards one system due to any change in procedure by the experimenter. As an example, consider a comparison of two systems with 10 replications, where System-1 is first tested 10 times, and then System-2 is tested 10 times. If there was any slight change (intentional or not) in the experimental procedure between the start and finish, then most of the trials with System-1 would have been conducted with a different procedure than those for System-2. This could affect the results. Blocking the systems ([System-1, System-2], [System-1, System-2], etc.) helps to insure that any change in procedure will affect the results for each system similarly.

Blocking: The six treatments for the Mars robot experiments (one experiment for each of the two tasks) will be blocked as shown in the following chart. The squares and arrows show the temporal order in which the trials with each system will be conducted (see page 44 for an explanation of the notation for the system types). The entire chart represents one experiment. Each row represents a block. Since eight replications will be conducted, there are eight blocks in this experiment.



The same experimental procedure should be followed for each participant. The experimenter should use a fixed script when talking to the subjects, to make sure that each receives the same information. The purpose of the experiment should be explained to the participant, and he should be told that the system is being tested and not himself. The participant should be taught to use the system that he will be using for the experiment, both by an explanation and demonstration of the controls by the experimenter, and then by actually using each control himself. After the participant has had a fixed amount of time to experiment with the controls, he should be asked to attempt a sample task, which is different from the experimental task.

The amount of training and practice time that the participant receives can affect his performance with the system. Experts may use the system and perform differently than novices. If expert level users are desired, then Bejczy [14] recommends the following training procedure. Training should take place in two cycles, each with numerous repetitions. The first cycle is to familiarize the participant with the system and the task, and the second is to familiarize him with the nature of the performance measurements that will be taken. According to Bejczy [14], these cycles should be separated by at least one day. In many cases, with the large number of participants suggested for this evaluation method, time restrictions will prevent extensive training of the participants. Also, novice level performance may be an important consideration if the system might be used by non-experts. In such cases, it will likely be best to make sure that the participants are initially all novice users who have never used the system before. Then, each of them should be given a short training period to familiarize them with the use of the system. As long as the training period is the same for each participant, then they will have approximately the same expertise.

When the real task is explained to the participant, he should be informed of both the goals (what needs to be done to accomplish the task) and the performance guidelines (not hitting obstacles, finishing as fast as possible, etc.). These guidelines will typically correspond to either the metrics that will be used to compute values for the evaluation criteria, such as the number of collisions, or the requirements of the task, such as staying within certain boundaries. If there are multiple guidelines, then the participant should be told of their importance relative to each other.

Goals and Guidelines for the Human Subject: The goals for the rover example are to collect 20 rocks (for the first task) and to move the base station (for the second task). The guidelines for the task might be to avoid hitting obstacles and falling into crevices, and to complete the task as efficiently as possible with respect to power.

Finally, the participant should attempt the experimental task. During the task, the data which will be used to determine values for the evaluation criteria should be collected. If possible, then it is best to have these data values collected automatically by the system. When this is not possible, the experimenter will need to collect these values manually, such as observing the trial and counting the number of collisions.

Data: The data that should be collected for the rover experiments includes the number of times any robot bumps into an object or falls into a pit, and the change in voltage of all the robots' batteries from the beginning to the end of the task.

3.5 Analyze the Data from the Experiments

There are a few statistical techniques that investigate the relationship between predictor variables (system dimensions in this case) and a response variable (an evaluation criteria in this case), such as ANOVA (ANalysis Of VAriance), regression, and t-tests. ANOVA analysis, however, is the appropriate method to use for these evaluations. Regression analysis concentrates on predicting the value of the response variable from the predictor variables [41], while the ANOVA analysis concentrates on determining the relationships. Additionally, regression requires that the values of the predictor variables be quantitative, while ANOVA allows qualitative types (such as holonomic/non-holonomic), and the ANOVA model does not require any assumptions about the nature of the statistical relationship between the predictors and responses [41]. ANOVA analysis is therefore more appropriate than regression for this type of evaluation. T-tests can be used if only two systems are being compared. If more than two systems are being compared, which may be the case with this sort of evaluation, then the ANOVA technique is most appropriate [32]. Since the ANOVA method can compare two systems as well as greater numbers, it is as a general approach for this methodology.

To determine how different systems compare to each other based on the judgment criteria, perform a single-factor ANOVA analysis on the data gathered for that criteria. This allows the systems to be ranked from best to worst for that particular criteria. If more than one criteria is being used, then a separate analysis should be performed for each, producing one ranking for each criteria. Remember, that since each experiment considered only one task, this ranking is also only for that task.

A brief description for how to perform the single-factor ANOVA analysis is given in Section 7.1, and Figure 3.1 shows the process used to produce the system rankings. Further details can be found in most statistical textbooks, such as [41]. The Box-Cox and Tukey multiple-comparison procedures are described in more detail in Chapters 7.1.1 and 7.1.3 respectively. In short, the Box-Cox procedure insures that each data set follows a normal distribution and that the variance of all the data sets is approximately the same. The Tukey multiple-comparison procedure insures that an entire group of comparisons has the level of statistical significance that is desired, rather than just each individual comparison.

The factor for this analysis will be the type of system, so all the different system types will be the factor levels, disregarding the fact that they may be from different system dimensions. In other words, rather than considering the two dimensions as separate factors (as they are later in a multi-factor ANOVA analysis on the data), this method considers all the treatments as levels of one factor. The single-factor

ANOVA analysis indicates whether there is any actual difference in the means of the data values, or if any apparent difference is due to noise. The ANOVA analysis can then be used to produce confidence intervals[4] around the means, thereby indicating what the system rankings are.

[4]A confidence intervals indicates, with a particular certainty, that the true mean value for that evaluation criteria is somewhere in the interval. More details on confidence intervals are found in Section 7.1.3.

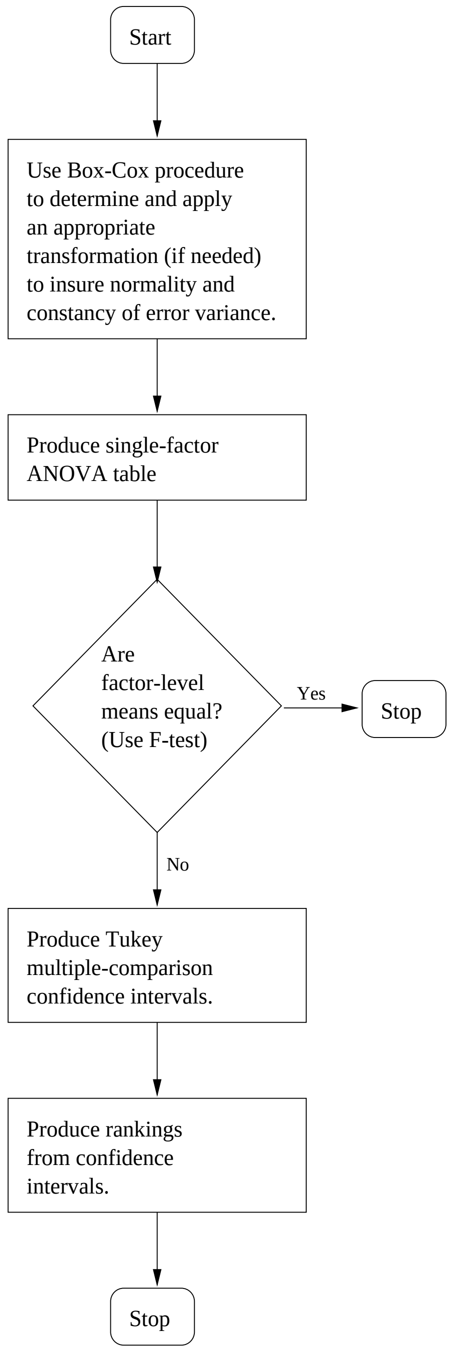


Figure 3.1: Flowchart of the process to produce the system rankings.

Collected Data: The following values represent the number of collisions and falls for the Mars rock collecting task of the rover experiment. The system types are listed in the left column, and the data values for the eight replications are listed across each row. These values are arbitrary, and were not collected from a real experiment, but they will be used to demonstrate the analysis procedure.

Number of collisions								
H1	1	0	2	5	0	0	1	1
H2	0	1	2	1	2	2	0	2
H5	2	2	3	1	0	0	3	1
N1	1	1	1	0	1	2	2	2
N2	0	0	5	1	5	4	2	2
N5	5	4	6	1	2	2	1	2
Number of falls								
H1	0	1	1	1	1	1	2	0
H2	1	1	0	0	1	2	1	0
H5	3	2	1	0	2	1	0	2
N1	1	1	1	1	2	1	1	0
N2	2	2	1	0	2	2	3	1
N5	1	1	2	1	2	1	1	1

Transforming the Data: Since the safety measure is determined by

$$safety = (number\_of\_collisions) + 2 * (number\_of\_falls)$$

the resulting safety values are shown in the following table.

Safety Data								
H1	1	3	4	7	2	2	5	1
H2	2	3	2	1	4	6	2	2
H5	8	6	5	1	4	2	3	5
N1	3	3	3	3	5	4	4	2
N2	4	4	7	1	9	8	8	4
N5	7	6	10	3	6	4	3	4

Using this data, the single-factor ANOVA analysis will be demonstrated for the safety criteria for the Mars rock collecting task. The safety analysis for the task of moving the base station and the power efficiency analysis for both tasks is performed similarly.

The Box-Cox procedure (Chapter 7.1.1) indicates that a transformation of $Y^{0.4}$ is needed to insure normality and constant variance of the data sets. After applying this transformation, the data is as follows:

Transformed Safety Data								
H1	0	2.9016	3.8967	6.1934	1.6800	1.6800	4.7514	0
H2	1.6800	2.9016	1.6800	0	3.8967	5.5086	1.6800	1.6800
H5	6.8217	5.5086	4.7514	0	3.8967	1.6800	2.9016	4.7514
N1	2.9016	2.9016	2.9016	2.9016	4.7514	3.8967	3.8967	1.6800
N2	3.8967	3.8967	6.1934	0	7.4044	6.8217	6.8217	3.8967
N5	6.1934	5.5086	7.9494	2.9016	5.5086	3.8967	2.9016	3.8967

Single-Factor ANOVA Table and F-Test: The single-factor in this analysis is the system type, with levels H1, H2, H5, N1, N2, and N5. The following table is the single-factor ANOVA table for the transformed data.

Source	Sum of Squares (SS)	degrees of freedom (df)	Mean Square (MS)	F^*
Factor A	45.93	5	9.185	2.439
Error	158.2	42	3.766	
Total	204.1	47		

The procedure for determining the appropriate test for equivalence of sample means is described in Chapter 7.1.2. The decision rule (using a 95% level of confidence) that is generated by this procedure is as follows:

> If $F^* \leq 2.50$,
> then conclude that all treatment means are equal,
> else conclude that all treatment means are not equal.

Using this rule, we can determine if the treatment means are all equal or not. F^* is obtained from the ANOVA table. In this instance, $F^* = 2.439$, so we conclude that the treatment means are not all equal.

System Ranking for Safety: The next step is to determine the confidence intervals for the means, in the manner explained in Chapter 7.1.3. The confidence intervals are as follows:

System	Mean	Lower Boundary	Upper Boundary
N1	2.6379	1.4052	3.8706
N2	2.3783	1.1456	3.6110
N5	3.7889	2.5562	5.0216
H1	3.2289	1.9962	4.4616
H2	4.8664	3.6337	6.0991
H5	4.8466	3.6128	6.0782

The following ranking can be determined from these intervals, by noting that systems with overlapping confidence intervals are statistically equivalent (as explained in Chapter 7.1.3).

1. N2
   N1
   H1
   N5
2. N1
   H1
   N5
   H5
   H2

This indicates that the results for systems N2, N1, H1, and N5 are statistically equivalent, and that N1, H1, N5, H5, and H2 are statistically equivalent, but N2 performed better than both H5 and H2.

If more than one system dimension was examined, a multiple-factor ANOVA analysis can also be performed to detect any main effects, which would indicate more general findings related to one level of a system dimension. In other words, a main effect would indicate that one particular level of a system dimension is better or worse than the other levels for this dimension, regardless of what levels are chosen along the other dimensions. In this case, the system dimensions are used as the multiple factors for the ANOVA analysis. Figure 3.2 shows the process to find main effects, and Section 7.2 describes how to conduct this analysis, and explains what main effects and interactions are. In short, an interaction is a varying influence on the data values for one factor by the different levels of another factor. A main effect indicates that data values of one or more levels of a factor are always greater (or lesser) than that factor's other levels, regardless of the settings of the other factors. More details, and background information can be found in [41].

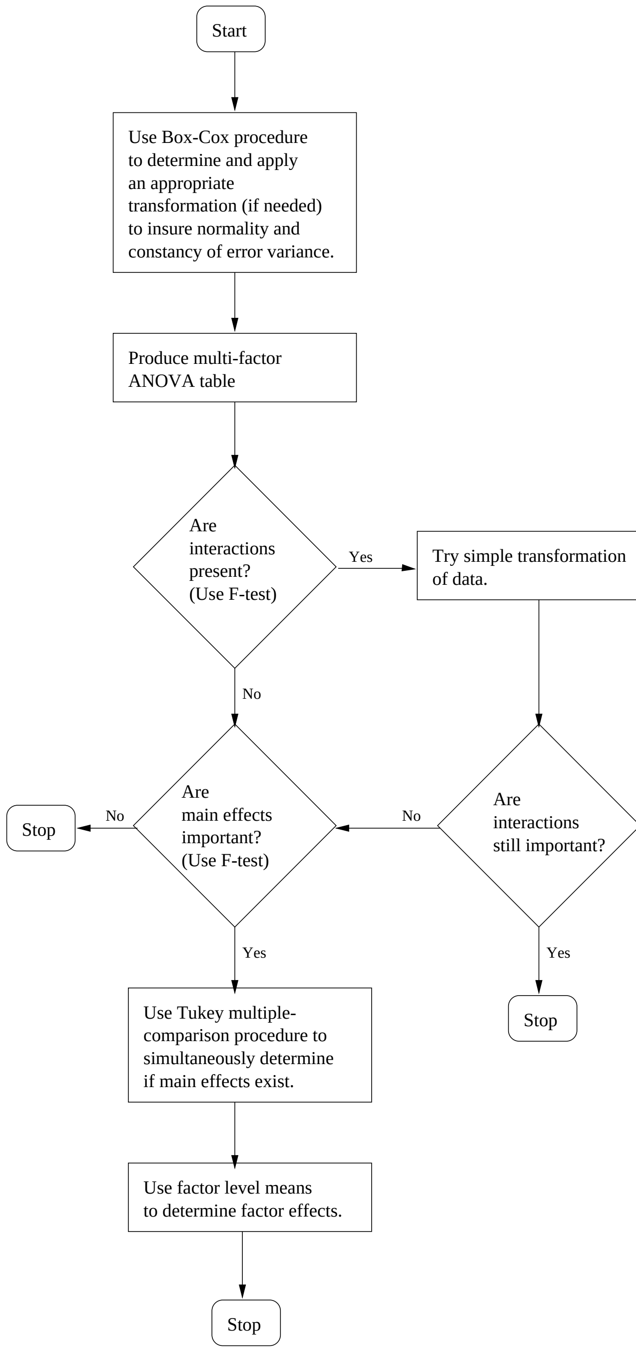


Figure 3.2: Flowchart of the process to produce the general findings.

Two-Factor ANOVA Table and Test for Interactions: The Box-Cox procedure has already been performed on the safety data values (page 58), and the following two-factor ANOVA table is generated from the transformed data. The two factors in this analysis are the number of robots and the holonomicity of the robots.

Source	Sum of Squares (SS)	degrees of freedom (df)	Mean Square (MS)	F^*
Factor A	22.79	1	22.79	6.052
Factor B	15.31	2	7.655	2.032
Interaction	7.822	2	3.911	1.038
Error	158.2	42	3.766	
Total	204.1	47		

If there are no interactions between the two factors, then we can search for main effects (explained in Chapter 7.2) The test for interactions is

> If $F^* \leq 3.24$,
> then conclude that no interactions are present,
> else conclude that interactions are present.

The F^* value for this test is obtained from the "F^*" column and "Interaction" row of the two-factor ANOVA table, and is 1.038. This indicates that no interactions are present, and we can look for main effects.

General Findings for Safety: The process for determining the tests for factor main effects is described in Chapter 7.2. The tests are:

If $F^* \leq 3.24$,

then conclude that no Factor A (number of robots) main effect exists,

else conclude that a Factor A main effect exists.

and

If $F^* \leq 4.09$,

then conclude that no Factor B (holonomicity) main effect exists,

else conclude that a Factor B main effect exists.

The F^* value for the Factor A and Factor B tests are obtained from the two-factor ANOVA table. These values are 2.032 and 6.052, indicating that no main effects are present due to the number of robots used, but a main effect due to the holonomicity may be present.

A multiple-comparison procedure must be used to insure that the entire family of comparisons conducted to determine main effects retains the 95% level of confidence. The Tukey multiple-comparison procedure described in Chapter 7.2.2 indicates that Factor B effects do in fact exist with a 95% family confidence level. Since there are only two levels for this factor, we can examine the means to find the main effect, and deduce the general finding that *holonomic vehicles were significantly safer than non-holonomic vehicles, regardless of the number of robots used.*

3.6 Summary

Formal evaluations are needed to further transform the field of telerobotics into a science. Currently, few researchers perform extensive evaluations on their systems. This chapter presented an evaluation methodology for comparing telerobotic systems for various tasks.

There are five main steps to evaluate telerobot systems. First, evaluation criteria by which to compare the systems are determined. This is composed of three substeps: deciding which traits of a system (such as safety and effectiveness) are most important to the design, determining which events or data influence those criteria, and determining a means to produce a single quantitative value for that criteria. These decisions are dependent on the purpose of the telerobotic systems to be developed.

The second step involves determining which types of systems to evaluate. The evaluations should decide upon a few system dimensions and specific points along those dimensions. The different system types to examine are produced by crossing the different dimensions and considering all the possible combinations.

Tasks to use the systems for are then created. If there are no specific applications that the experimenter wants to test, then a more formal approach involves examining the classifications of a task taxonomy (or some subset of it). If the latter method is used, then experimental tasks must be developed to represent each of the taxonomic classifications.

A set of experiments is then conducted, with one experiment per task. In each experiment, multiple human subjects use each of the systems for a task with real robots. The same procedure should be used with each human subject, including identical instruction and training.

Finally, the data collected from the subjects' attempts is used to determine the systems' performance. System rankings can be produced with a single-factor ANOVA analysis. General findings related to one or more levels of a system dimension may be found with a two-factor ANOVA analysis if more than one system dimension was examined.

Chapter 4

Multiagent Telerobotic System Descriptions

Multiagent telerobotic systems (MTSs) can differ in many ways. For instance, they can differ in the amount of autonomy the telerobots are provided with, the hardware, the number of telerobots controlled at one time, the form of control, and the nature of the interaction between the human operator and the telerobots. In addition, each of these dimensions can have many different levels. For example, the level of autonomy of the telerobots can be direct manual control or one of numerous levels of supervisory control. There is a potentially infinite number of different MTSs that can be created.

For this research, MTSs that differ along two dimensions were examined. These dimensions are the amount of autonomy that the telerobots are provided with and the number of robots that the human operator is controlling at a time. Four systems are examined in this thesis, differing in those two dimensions. Section 4.1 establishes the dimensions and the respective points on those dimensions. Section 4.2 describes the four systems. The underlying robot architecture of the systems is described in Section 4.3.

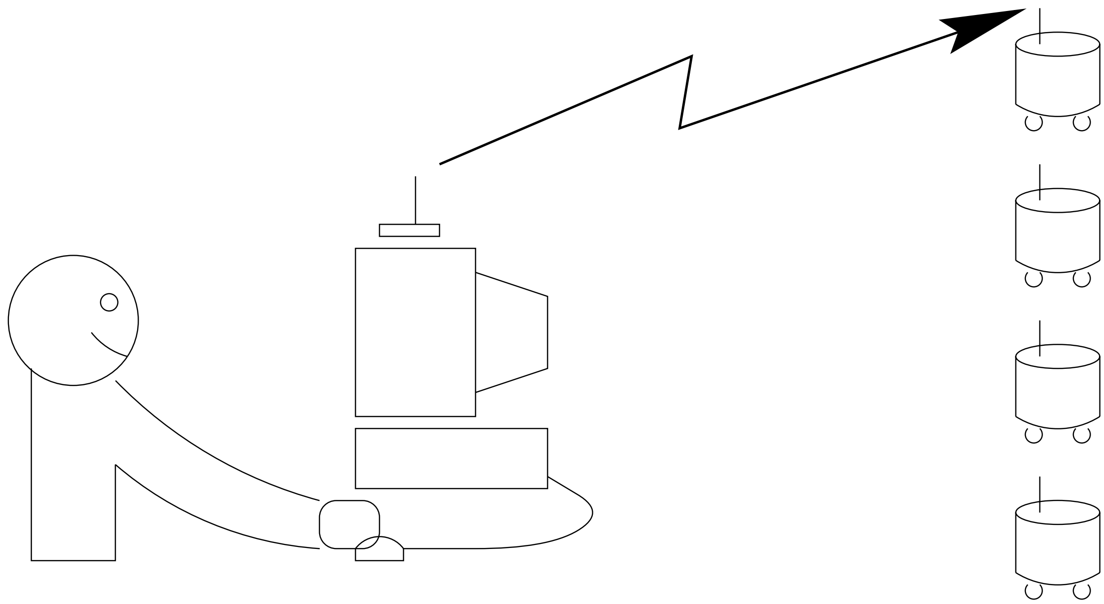


Figure 4.1: **Individual** control. The operator controls one robot at a time, switching control between the robots.

4.1 System classifications

One difference between the multiagent telerobotic systems examined is the number of robots controlled by the operator at one time. Individual control and entire group control were tested. Another possibility that was not examined is subgroup control, where the operator chooses a subset of robots to send commands to. Table 4.1 shows the dimensions and their corresponding points that were examined.

With **Individual** control, the human operator controls one robot at a time, switching between robots, as shown in Figure 4.1. This does not mean that the robots cannot execute any previous instructions while the operator is controlling one of the other robots. It simply means that the operator can issue instructions to only one robot at any one time.

For **Group** control, the operator gives instructions to the entire group of robots at all times, as shown in Figure 4.2. Any instruction given to the robots is received

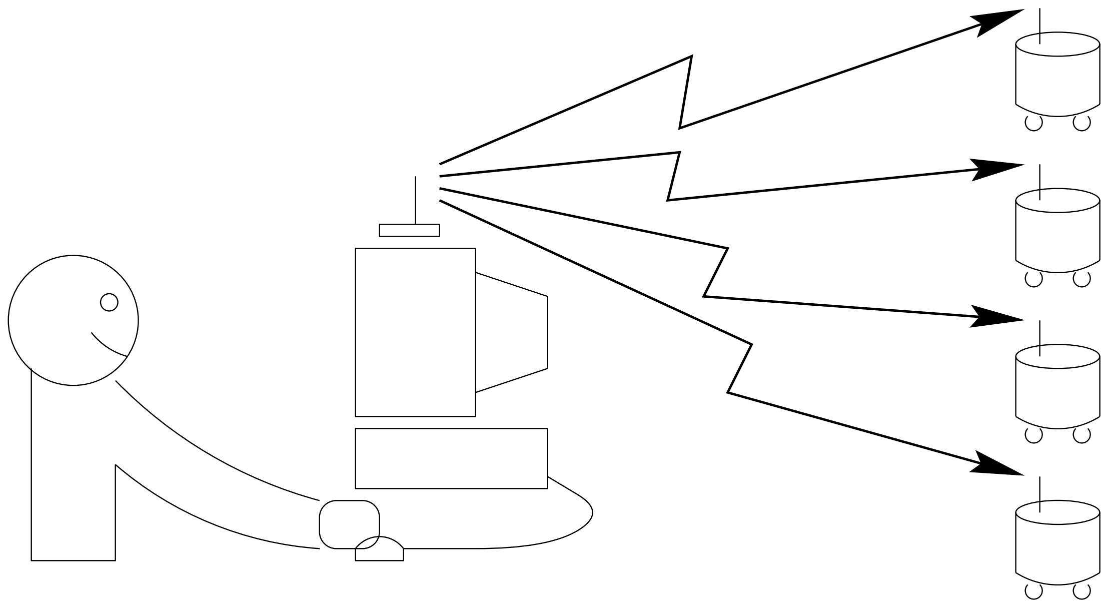


Figure 4.2: **Group** control. The operator controls all of the robots at once.

and executed by all of the robots. Therefore, the operator cannot single out a particular robot or subgroup of the robots for special instructions.

The other difference between MTSs that was considered is the level of autonomy of the telerobots. Direct manual control and supervisory control systems were compared. With **direct manual** control, the human specifies all robot motion through continuous input [11]. The operator must continually send motion commands (or set the controls to continually send them) to make the robot move, since there is no additional influence to the telerobots motion caused by a computer. An example of a mobile telerobot using direct manual control is a radio-controlled toy car.

Sheridan [47] defines **supervisory** control in two ways. In the strictest sense, supervisory control means that one or more human operators are intermittently programming and continually receiving information from a computer that itself closes an autonomous control loop through artificial effectors and sensors to the controlled process or task environment. Sheridan states, in a less strict sense, supervisory

Table 4.1: The Four Systems and the Dimensions They are Derived From

# of robots controlled at a time		Level of Autonomy: Direct Manual Control	Level of Autonomy: Supervisory Control
	Individual	Direct Manual Individual	Supervisory Individual
	Group	Direct Manual Group	Supervisory Group

control means that one or more human operators are continually programming and receiving information from a computer that interconnects, through artificial effectors and sensors, to the controlled process or task environment. The primary difference between these two differences is when the human gives instructions to the telerobots. In the strict sense, the human instructs the robots, then lets them execute the instructions autonomously, and then repeats the cycle. In the less strict sense, the human commands the robots continuously, and these instructions are combined with the output from a computer. The **Supervisory** control systems used in the experiments conducted for this research allowed the user to choose whether or not to intermittently or continuously instruct the robots. Therefore, they fit somewhere between these two definitions.

Backes [48] gives a simpler and more inclusive definition of supervisory control that is robot specific. He defines supervisory control by saying that robot motion may be caused by either human inputs or computer generated inputs.

For this dissertation, the term **supervisory control** means that a human operator can affect the motion of a robot, either by continuous or intermittent programming, while the computer also influences the motion of the robot. Furthermore, the human continuously monitors the feedback from the computer.

These two factors, the level of autonomy and the number of controlled robots, were crossed to produce four types of systems for examination. Therefore, as shown in Table 4.1, the systems used in these experiments include:

- **Direct Manual Individual** control
- **Direct Manual Group** control
- **Supervisory Individual** control
- **Supervisory Group** control

4.2 Representative systems for each classification

For each of the four classes of systems considered, a representative system was developed. This section describes the control techniques and the details of the human interface for each of those systems, and the following section describes the underlying architectural details.

4.2.1 Direct Manual Group Control

The **Direct Manual Group** control system allows the user to control the group of mobile robots in a strict teleoperative sense. That is, the robots are not running any low-level behaviors, and thus, they do not contribute to determining their motion. The operator can give instructions to the robots in terms of a compass direction and speed for travel. Any instructions are sent to all of the robots in the group at the same time. The user cannot single out an individual robot or subgroup of the robots for special instructions.

In the experimental testbed, an on-screen joystick, depicted in Figure 4.3, is used to input the direction and speed (the formation toggle buttons shown in this figure are discussed in Section 4.2.3). The joystick is marked with the compass directions N, E, S, and W. The operator uses a mouse to position this joystick. Clicking anywhere in the white circle of the joystick with the left mouse button sets the joystick, drawing a line from the center of the joystick to where the user clicked, and sending a corresponding movement command to the robots based on the direction and distance from the center of the joystick to the location clicked (shown in Figure 4.4). The farther the distance from the center, the greater the magnitude of speed command that is sent to the robots. For instance, if the user clicks in the section of the circle between the North and East markers and close to the edge of the circle, the system will send a command to the robots to move northeasterly at close to the maximum speed allowed.

Once the joystick is set, it will remain at that setting indefinitely and continue to send the same movement command to the robots until the user clicks again in the joystick window. To clear the joystick, the user can click with the middle mouse button anywhere in the white circle of the joystick. This erases the line on the joystick and stops sending movement commands to the robots. With the **Direct Manual** control systems, this causes the robots to stop.

4.2.2 Direct Manual Individual Control

The **Direct Manual Individual** control system is similar to the **Direct Manual Group** control system, except that the operator gives commands to only one of the robots at a time instead of the whole group, manually switching between the robots. Just as in the **Direct Manual Group** system, the robots are not running any autonomous behaviors, so only the operator's commands contribute to the motion

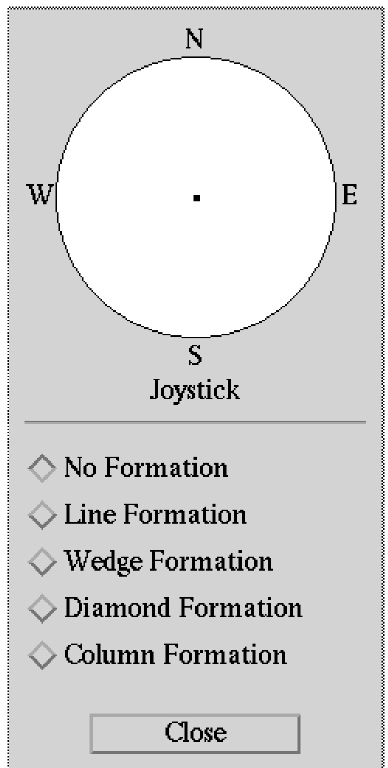


Figure 4.3: The on-screen joystick and the formation toggle buttons. The joystick is used for giving the robots directional and speed information in terms of compass directions. The formation toggle buttons are used for switching between different formations. The joystick is available for use in all four of the systems tested, while the formation toggle buttons are available only in the **Supervisory Group** control system.

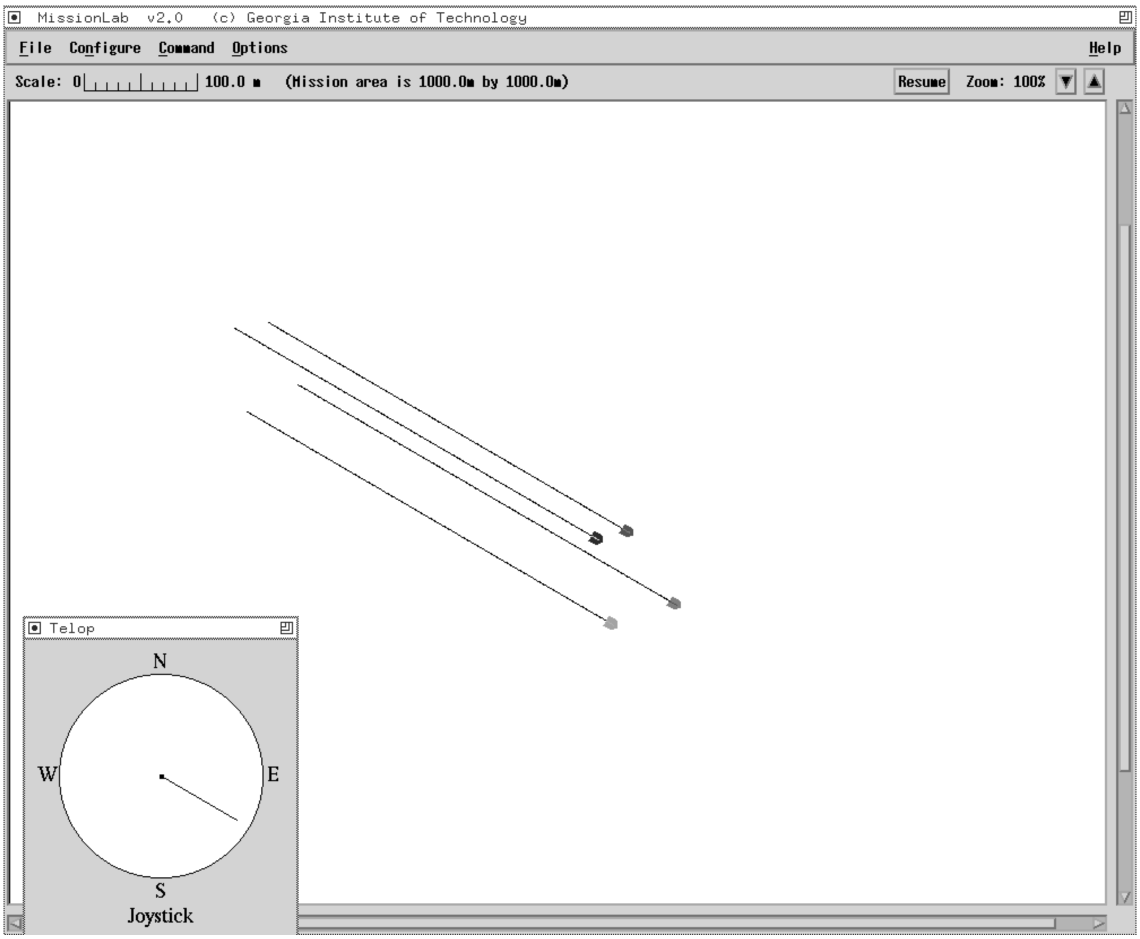


Figure 4.4: The on-screen joystick allows the operator to the instruct the robots to move at a particular velocity (both direction and speed).

of the robots. The operator gives instructions in terms of a compass direction and speed, just as in **Direct Manual Group** control. A robot will continue to execute previously given direction and speed commands even after the operator has switched the control to another robot, but it cannot receive commands unless it is the operator's specified focus of attention.

The operator uses the same on-screen joystick that was used in the **Direct Manual Group** control system to give instructions to the robots. He also has a set of toggle buttons (Figure 4.5) that is used to tell the system which robot should receive the current instructions. Whichever toggle button is currently depressed indicates which robot is currently the focus of attention. Each toggle button is labeled with the number of the robot as well as a color. The icons representing the robots on the screen are color coded for these colors. Similarly, the real robots used in the experiments each had a colored object on its antenna, similarly color coded. When the user switches control from one robot to another, the line on the joystick showing the current movement command changes to the last command given for the robot that has been selected.

4.2.3 Supervisory Group Control

With the **Supervisory Group** control system, the robots are running low-level behaviors to handle obstacle avoidance and to avoid collisions between robots. The details of these behaviors are given in Section 4.3. The user has three methods for controlling the motion of the robots, and as with **Direct Manual Group** control, any instructions given to the robots go to all the robots in the group. No robot or subgroup of robots can be singled out for different instructions than the others.

The first method for controlling the robots' motion is the on-screen joystick described above, that allows the operator to give direction and speed commands.

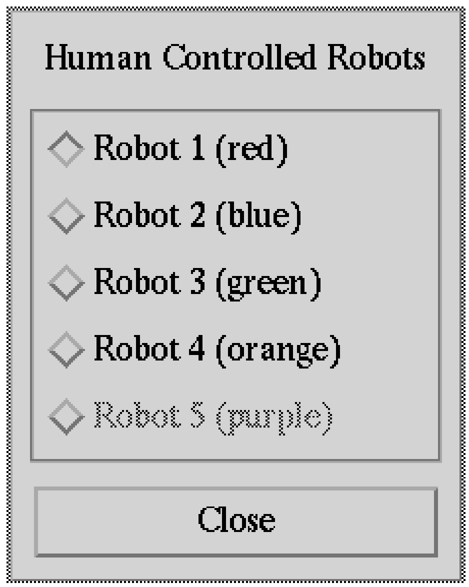


Figure 4.5: Robot Selection Box. The human operator uses this box to indicate which robot he is currently sending commands to. Robot selection is used during both of the **Individual** control systems (**Direct Manual Individual** and **Supervisory Individual**). In this figure, there are 4 robots that are active (highlighted), and Robot 1 is currently the focus of attention (button depressed), and will therefore receive any new commands given by the user.

With the **Supervisory** control systems, however, the speed instructions do not translate directly to the speed of the robots, but instead represent a gain value, or the amount of influence that the joystick's directional command will contribute to the actual direction that the robots move. This is explained in more detail in Section 4.3.

The second method for controlling the robots is by setting a waypoint or a path of waypoints for the robot to follow. During the robot execution, a map, displayed on the operator's workstation monitor, shows the task environment in low detail, as well as the locations of the robots within that task environment. An example of this layout is shown in Figure 4.6. As the robots move, their changes in position are marked on this layout. This representation is not guaranteed to show the exact locations of the robots, but shows the best estimate of their locations based on dead-reckoning performed by each robot. The operator can use the mouse to point and click on the layout to set a waypoint. The robots will then try to move to the corresponding location of that waypoint in the real world. The operator can also set additional points, one after the other, to create a path. The robots will move from waypoint to waypoint in order, as shown in Figure 4.7. The robots follow the path as a group. Therefore, if any robot reaches a waypoint before the other robots, then that robot waits until all of the robots have arrived before continuing to the next. Because all of the robots cannot actually be in the same location at the same time, a waypoint is considered achieved when the center of mass of the group of robots is within a certain threshold distance of it.

Clicking in the layout with the left mouse button sets an initial waypoint at the real world location corresponding to the location in the map where the operator clicked. If a waypoint path has already been set, then clicking with the left mouse button clears the earlier path and sets a new initial point. After the initial waypoint has been set, clicking with the middle mouse button sets additional ones, thereby creating a path of waypoints. If an initial point has not been set, then clicking with the middle button will create one. Finally, clicking with the right mouse button will clear the entire path.

The third method for controlling the robots is to declare spatial formations that the robots should try to maintain while they are moving. There are four different formations that the robots can be instructed to assume. These are line, column, diamond, and wedge. Additionally, the robots can be commanded to assume no formation. The underlying formation control code was developed as part of the UGV Demo II project [13]. When moving in formation, if one robot slows down for some reason, such as when an obstacle is in its way, the others will slow down to wait for it, so as not to cause the group to stray too far from the desired formation. Figure 4.8 shows an example formation. The human operator uses the set of toggle buttons below the joystick (Figure 4.3) to instruct the robots to assume a desired formation.

4.2.4 Supervisory Individual Control

In the **Supervisory Individual** control system, the robots run low-level behaviors to avoid obstacles and other robots, just as in the **Supervisory Group** control system. With the **Supervisory Individual** control system, however, the robots are controlled on an individual basis, just as in the **Direct Manual Individual** control system. The operator uses the same toggle buttons (Figure 4.5) to indicate which robot is currently the focus of attention. Similarly, the operator can use the on-screen joystick (Figure 4.3) to give the robots direction and speed instructions, although only to one robot at a time. The human can also set waypoints and paths of waypoints for the robots to follow in the same manner as with the **Supervisory**

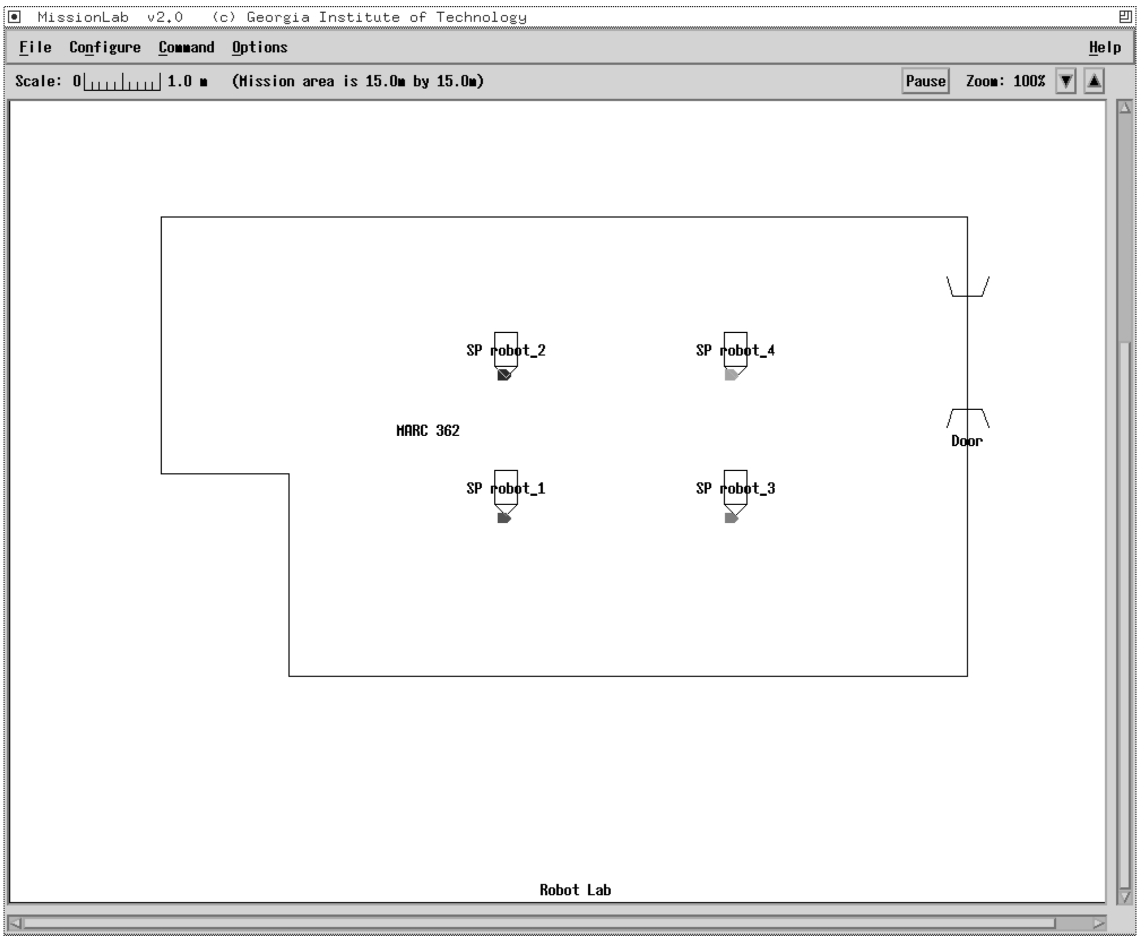
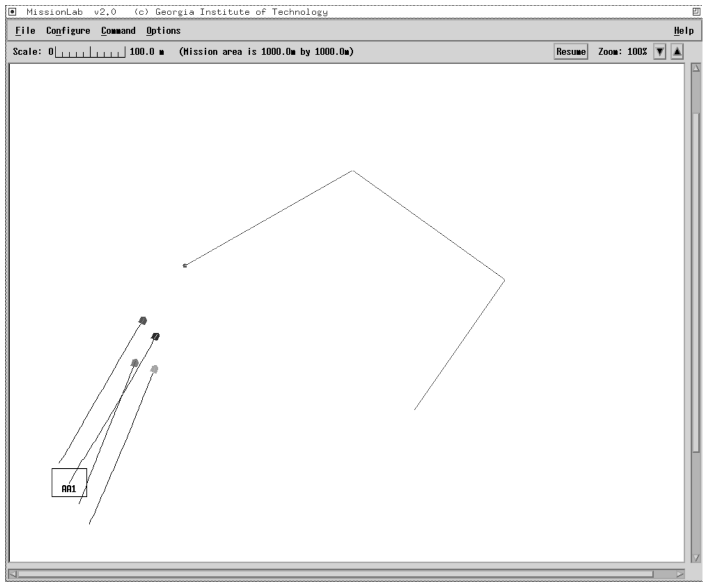
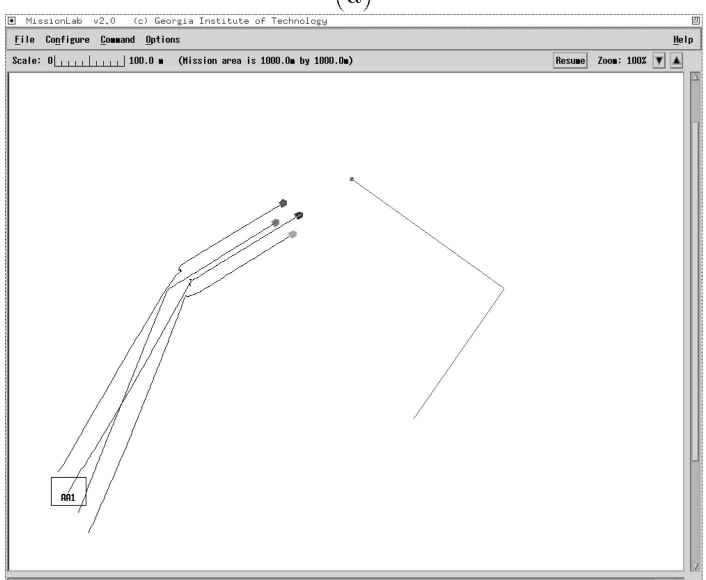


Figure 4.6: On-screen depiction of the task environment. The layout shows a low-detailed two-dimensional line-diagram of the task environment. This example from one of the tasks used in the experiments shows the walls of the room, the starting positions of the robots, and the current positions of the robots. The operator uses this layout when giving waypoint instructions.



(a)



(b)

Figure 4.7: Waypoint control. In (a), the operator has set a path of waypoints, and the robots are moving towards the first waypoint. In (b), the robots have achieved the first waypoint in the path, and are moving to the second.

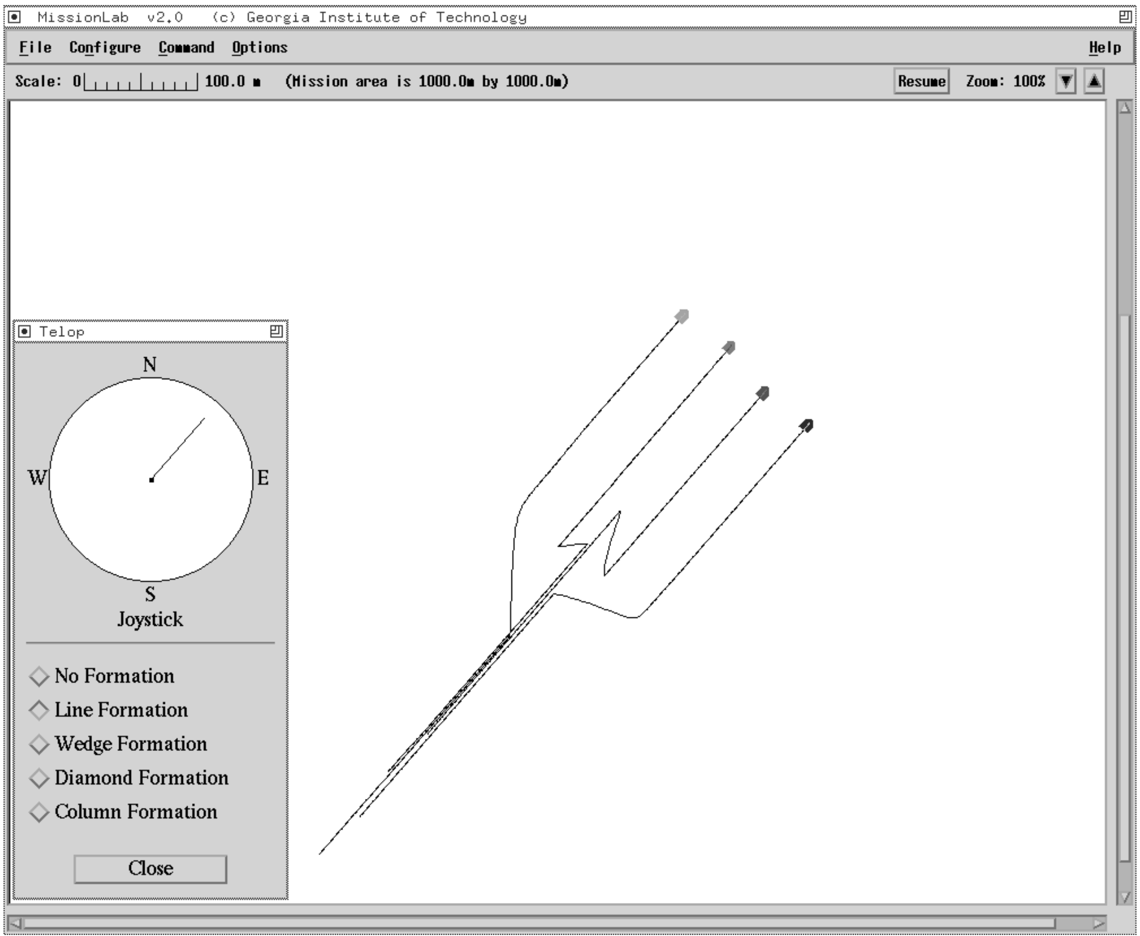


Figure 4.8: Formation control. The robots try to maintain a spatial formation specified by the operator. The operator has just switched the robots from a column formation to a line formation.

Group control system, except that each robot has its own individual waypoints, and its path is shown on the layout in the same color as the robot (each robot has its own separate color in the layout).

4.3 Underlying Robot Architecture

These multiagent telerobotic interfaces have been incorporated into the *MissionLab* system [33]. *MissionLab* is a system for specifying, simulating, and executing multiagent mobile robot missions. *MissionLab* takes high-level specifications and executes them with teams of real or simulated robot vehicles. It provides tools for quickly creating behavior-based robot programs, which can then be run either in simulation or on hardware without altering the control software. The architecture for *MissionLab* is shown in Figure 4.9. Those components shown in gray already existed as part of the *MissionLab* system, while the non-gray components were added as part of this research.

The underlying control architecture for the robots uses a schema-based reactive architecture [3]. In the schema-based approach, each reactive behavior, or motor schema, tries to influence the behavior of the robot by producing a vector in the direction consistent with the behavior's goals and with a magnitude that reflects the importance of going in that direction. The vectors of all the active motor schemas are summed and normalized, and the resulting vector is sent to the robot for execution. Each schema has a specific behavioral function. For example, the **avoid-static-obstacle** schema tries to move the robot away from obstacles by producing an output vector pointed directly away from the obstacle and with a magnitude based on the current distance of the robot from the obstacle, with smaller distances producing greater magnitudes. More than one schema is usually used at once. The influence from each of these specific behaviors combines to create a more complex

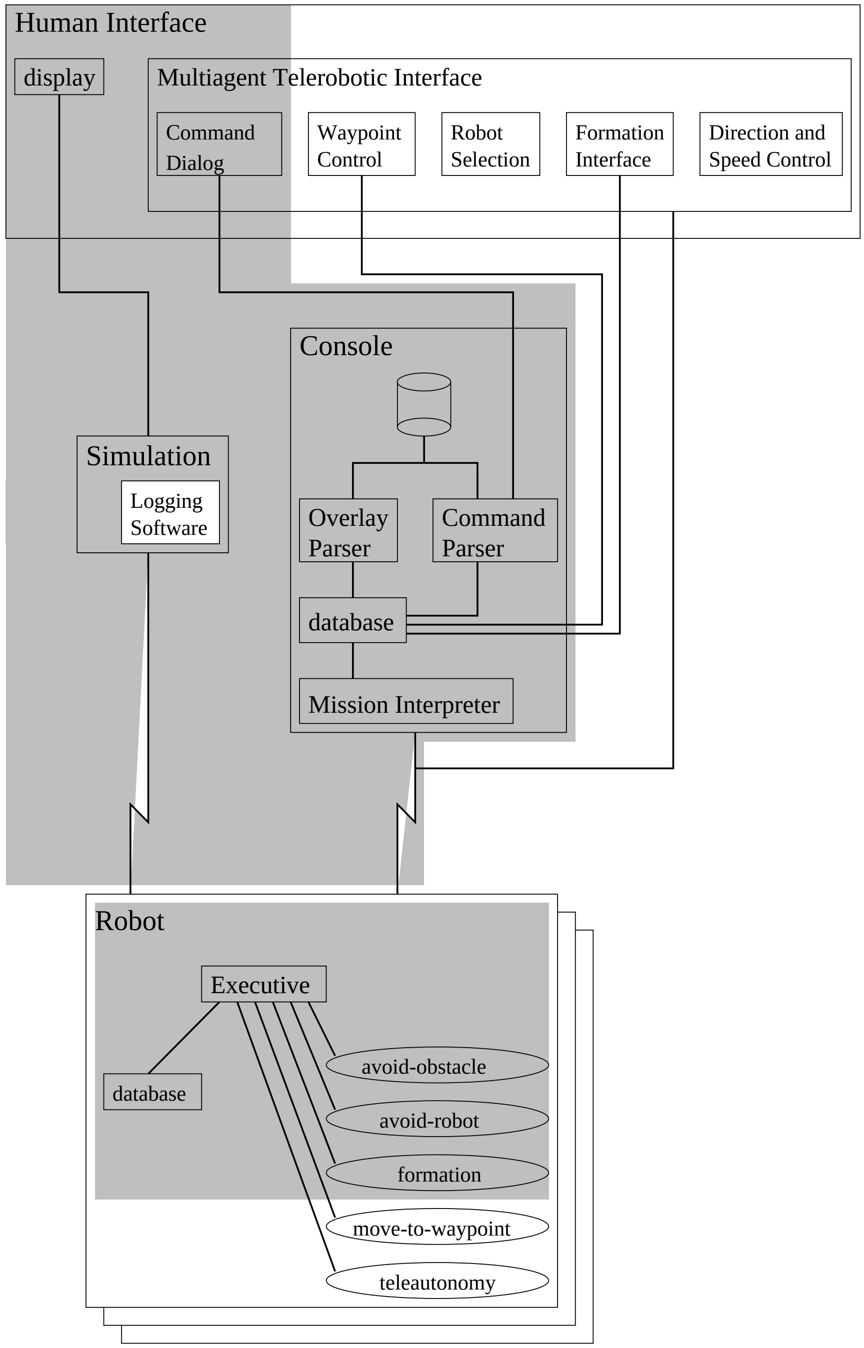


Figure 4.9: System architecture for *MissionLab* including the added multiagent telerobotic interface. The components shown in gray already existed in *MissionLab*, and the non-grayed components were added to accommodate this research.

behavior.

The telerobotic control systems used in this research use **avoid-static-obstacle**, **avoid-robot**, **move-to-waypoint** [3], **maintain-formation** [13], and **teleautonomy** [7] schemas. The **avoid-static-obstacle** schema was described above. The **avoid-robot** schema functions the same as the **avoid-static-obstacle** schema, except that it only produces vectors pointing away from other robots. The **move-to-waypoint** schema produces a vector with a fixed magnitude equal to a preset gain gain value and a direction toward the next waypoint. The **maintain-formation** schema tries to keep the robots in a specified formation. The **teleautonomy** schema takes an input in the form of a compass direction and speed from the on-screen joystick and produces an output vector to move the robot in that direction and with that speed. The details on how each schema works are given in Appendix B.

Schemas are often grouped together, and the outputs from each schema in the group are summed and normalized to produce a motion vector for the robot. These groupings are called assemblages. A short description of each assemblage of schemas, and how each schema is used in the assemblage, is given here. In the **Direct Manual** control systems, both **Individual** and **Group** control, the **teleautonomy** schema is the only active schema. Therefore, the vector generated by this behavior is passed directly to the robot for execution. In reality, it goes through the summing and normalization step, but there are no other vectors to sum it with. Each robot is running its own **teleautonomy** schema. With the **Direct Manual Group** control system, the same direction and speed information that the operator last entered into the on-screen joystick is passed to the **teleautonomy** schema on each robot. With the **Individual** control system, each robot's **teleautonomy** schema receives a different and individual direction and speed from the joystick, based on what the user last entered for that robot.

In the two **Supervisory** control systems, there are multiple schemas active that are common to both **Individual** and **Group** control. **Avoid-static-obstacle** and **avoid-robot** schemas keep the robots from colliding with each other and objects in the environment. The **teleautonomy** schema is active, allowing the user to direct the robots to move in a particular compass direction, either individually or as a group, depending on the form of control. A **move-to-waypoint** schema is also active. If the user gives waypoint commands, then this schema uses the waypoint (or the first waypoint in a series of waypoints) as the goal location and outputs vectors normally. When the robot reaches this waypoint, the goal location is changed to the next waypoint in the path. When the operator has not set any waypoints for the robots to follow, then the gain for the **move-to-waypoint** schema is set to zero, so the output vector does not influence the motion of the robots. In the **Supervisory Group** control system, a **maintain-formation** schema is also used. This produces output normally if the user has set a formation for the robots to use. When the user has chosen "No Formation", then the gain for this schema is set to zero.

Regardless of whether **Individual** or **Group** supervisory control is being used, each of the robots has their own set of the schemas running. The **avoid-static-obstacle** and **avoid-robot** schemas for a robot receive their input from the perceptual schemas running on that robot. The **move-to-waypoint**, **maintain-formation**, and **teleautonomy** schemas receive input both from the individual robot that it is running on and from the user interface. When using **Individual** control, however, these last three schemas each receive different user input, while they receive the same user input under **Group** control.

Although the schema-based architecture is used by the robots in the experiments, the results should be applicable to all behavior-based telerobots. Nothing about the telerobotic system configurations or the tasks used in the experiments depends strictly upon the schema-based approach. Any system in which similar behaviors can be generated should be usable for this research.

4.4 Summary

This dissertation examines four multiagent telerobotic systems. These systems differ in terms of the amount of autonomy that the telerobots have and the number of robots that the human controls at a time. For the amount of autonomy, **Direct Manual** and **Supervisory** control were considered. **Individual** and **Group** control were examined for the number of robots controlled at a time. Therefore, the four systems that were compared are **Direct Manual Individual** control, **Direct Manual Group** control, **Supervisory Individual** control, and **Supervisory Group** control.

Human interfaces and control systems were developed for each of these systems. The robot control systems were created as part of the *MissionLab* system and use a schema-based reactive architecture.

Chapter 5

Experimental Tasks

The goal of the experiments conducted for this dissertation is to determine which system types perform best for which kinds of tasks. To do this, one must know what types of tasks exist and choose tasks to represent each of these classes. A literature search did not find any existing mobile multiagent task taxonomies. Therefore, one has been developed. This taxonomy is presented first, and then the experimental tasks are described.

5.1 Taxonomy of Mobile Multiagent Tasks

Examining multiagent mobile robot (and animal) tasks led to the discovery of commonalities that serve as the basis for this new taxonomy, which categorizes tasks in terms of the relative motion of the agents. It should be noted that the taxonomy makes no attempt to separate the tasks by other differences, such as the reasons the robots move to particular locations, manipulation or surveillance of the world that the robots do (either while moving or at the goal locations), or the decision making process that leads to a particular movement. The examined Multiagent Telerobotic Systems (MTSs) differ in the way they allow the human operator to specify the movement of the robots. Therefore, the task classes were chosen to differ in terms of the type of movement of the robots relative to each other. The identification of these task classes was strongly influenced by the work of Gage [23] on the identification of formations for moving large groups of mobile robots (Section 2.1.2). It is quite possible for other taxonomies, yet to be developed, to be used with the testing methodology described in Chapter 3.

Figure 5.1 shows the taxonomy. There are three ways of classifying the relative movement of the agents. The first classification is whether the task is a *coverage* or *convergence* task. *Coverage* (*Cov*)[1] tasks require that the robots spread out to cover an area evenly or to cover as much area as possible, as shown in Figure 5.2(a) and Figure 5.2(c). These tasks usually require that the robots maintain a uniform distribution over the area covered. Often the purpose of this coverage is to maximize the sensor capabilities of the entire group. Some examples of *coverage* tasks include:

- foraging/search
- surveillance/reconnaissance
- grazing/cleaning
- communication relaying
- barrier/sweep tasks

Convergence (*Conv*) tasks require the robots to gather together or move while grouped together. In these tasks, the robots often converge to help each other with a difficult job that is easier to do in numbers. Figures 5.2(b) and (d) are examples of *convergence* tasks. Some examples of this class of tasks are box pushing and multi-arm manipulation, in which multiple robots may be able to push or handle the object more easily than one. Similar examples from nature include retrieving

[1]The abbreviations for each classification method are appended together to denote a task category, as shown in Table 5.1.

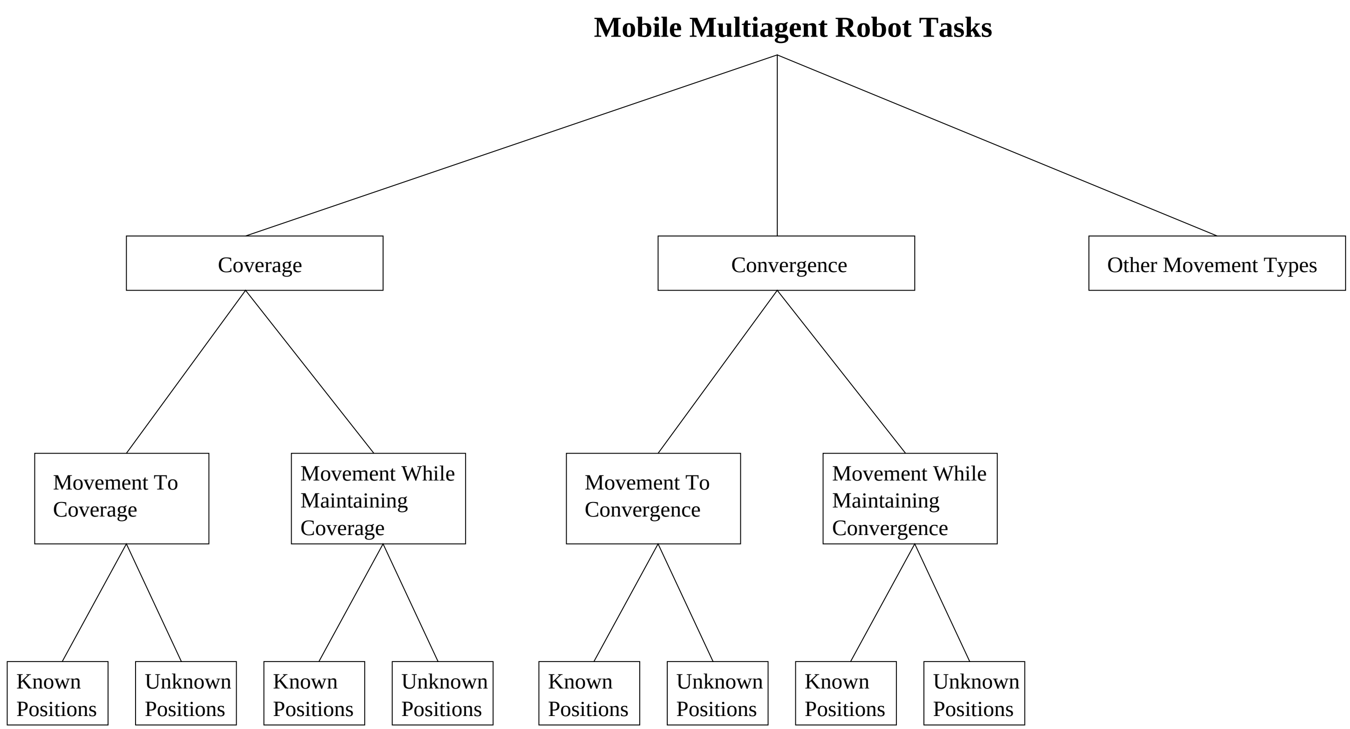


Figure 5.1: Mobile Multiagent Task Taxonomy.

large prey [21], and gathering for protection, both in amorphous groups such as flocks of birds or fish, and in defensive circles, as demonstrated by elephants, bison, and quail [17].

The second classification type considered is whether the agents are moving to positions of coverage/convergence or moving while maintaining these positions relative to each other. In *movement-to* (*Mt*) tasks, the robots are either in the process of spreading out to cover an area or gathering in for convergence, as shown in Figures 5.2(a) and (b). In *movement-while-maintaining* (*Mw*) tasks, the robots are moving while trying to stay spread out or grouped together, such as in Figures 5.2(c) and (d). A multiagent box-pushing task is an example that demonstrates both a *movement-to* subtask and a *movement-while-maintaining* subtask. In the first part of box-pushing, the robots gather on one side of the box from various locations so that they will be able to push it (*movement-to-convergence*). After the robots are gathered on one side of the box, then they move while staying grouped together to actually push the box (*movement-while-maintaining-convergence*).

The third classification type in the taxonomy is whether or not the agents have known or predefined positions relative to each other that they should try to move to or maintain. With *known-positions* (K), the robots try to move to or maintain a predefined location relative to the other robots. For instance, in some military formations, each soldier or vehicle has a particular location within the formation. With *unknown-positions* (U), a particular robot can be anywhere in the world, so long as the group as a whole satisfies the other movement restrictions (such as coverage or convergence).

Certain types of coordinated group movement do not fit neatly into any encompassing category. In this taxonomy, these types of movement are categorized simply

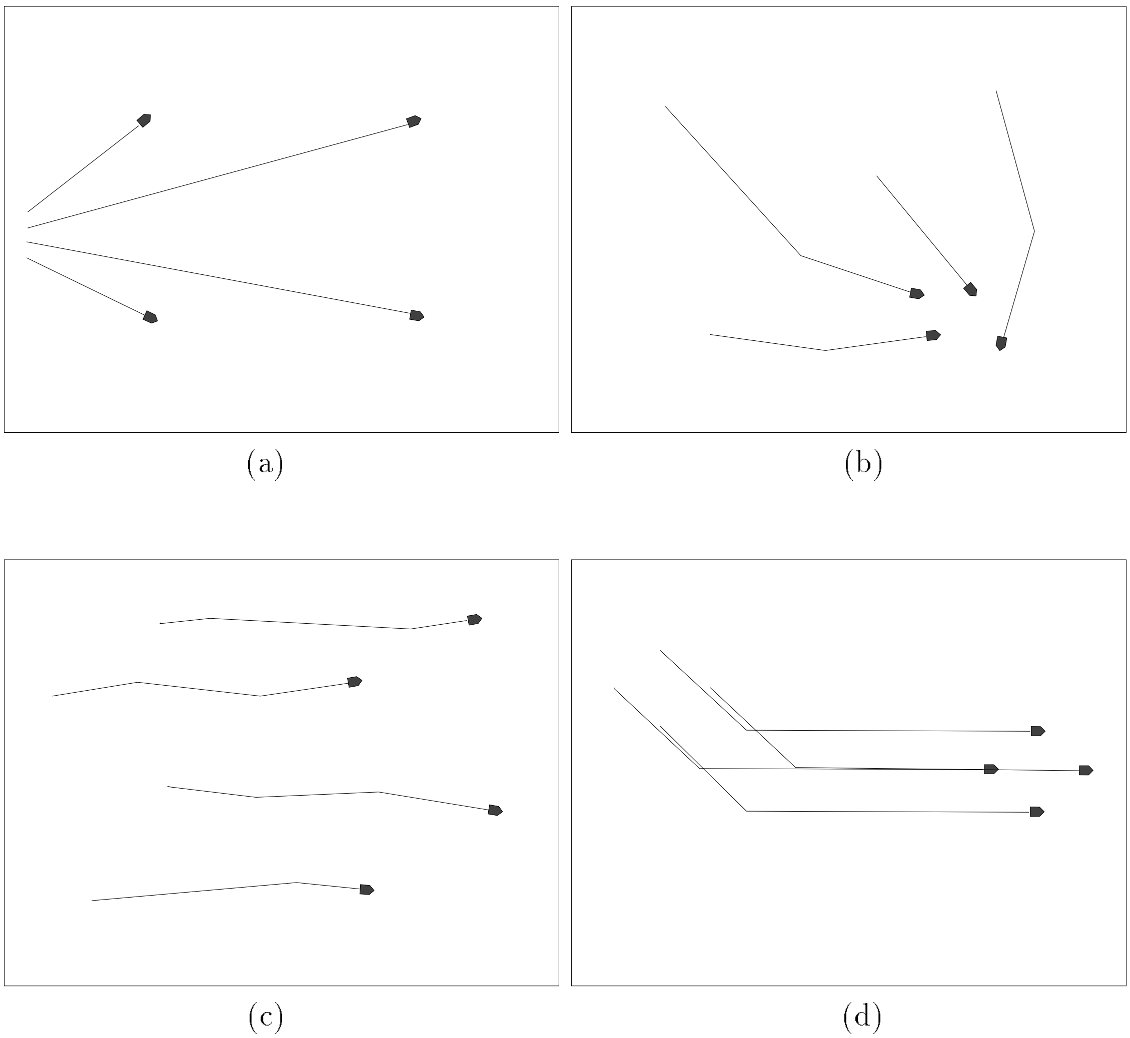


Figure 5.2: Classes of Tasks. (a) *Movement-to-coverage*, (b) *Movement-to-convergence*, (c) *Movement-while-maintaining-coverage*, (d) *Movement-while-maintaining-convergence.*

Table 5.1: Categories in the Mobile Multiagent Task Taxonomy.

Task Category	Abbreviation
movement-to-coverage-with-known-positions	*MtCovK*
movement-to-coverage-with-unknown-positions	*MtCovU*
movement-to-convergence-with-known-positions	*MtConvK*
movement-to-convergence-with-unknown-positions	*MtConvU*
movement-while-maintaining-coverage-with-known-positions	*MwCovK*
movement-while-maintaining-coverage-with-unknown-positions	*MwCovU*
movement-while-maintaining-convergence-with-known-positions	*MwConvK*
movement-while-maintaining-convergence-with-unknown-positions	*MwConvU*
other-movement-types	*O*

as *other movement types* (O). An example is movement that is scripted, both spatially and temporally, such as a football play. Not only do the separate players each have a particular path they are supposed to run in relation to the other players, which could be explained by *known positions*, but they have temporal restrictions about when they should be at each position.

Table 5.1 shows the nine task categories in the taxonomy and abbreviations for each. These abbreviations can be used to indicate which movement type a group of robots are executing, where:

$$Group\_name = Category\_abbreviation(robot\_list)$$

For instance, A = *MtCovU*(1,2,3,4) denotes that group A consists of robots 1 through 4, which are *moving-to-coverage-with-unknown-positions*. The utility of this notation is demonstrated below.

Some applications do not fit neatly into a single class, but are composed of sequences of subtasks that fit into these classes. For instance, foraging by ants (or robots) may involve subtasks of several classes. While searching, the ants spread out, thus *moving-to-coverage* (*MtCovU*). Once a large food source is found by one ant, more ants follow the chemical trail left by the first (*MtConvU*). Carrying a large piece of food back to the nest may require the help of several ants moving together as a group ($MwConvU$).

Some tasks require a large group of robots to split into subgroups, with each subgroup doing something different. In this case, the task classifications in this taxonomy should be applied in a recursive manner. Each subgroup should be considered a single agent, and the relative motion of the collection of subgroups should be classified using the taxonomy. Then the relative motion of the robots within a subgroup is examined and classified. Figure 5.3 shows an example in which the subgroups (when considered as individual entities) are *moving-to-coverage-with-known-positions*, yet the robots within each subgroup are *moving-while-maintaining-convergence-with-known-positions*. This behavior is applicable to soldiers (or robots serving as soldiers), with the soldiers in each subgroup staying close to protect each other, and the two subgroups spreading apart to maximize sensor capabilities. We can denote this behavior as follows:

$$Group = MtCovK(A, B)$$

$$A = MwConvK(1, 2, 3, 4)$$

$$B = MwConvK(5, 6, 7, 8)$$

It is possible to carry this process further, considering this entire robot group (Group) as a subgroup in a larger group of robots (not shown in the figure), and these larger subgroups might be exhibiting some other type of movement in the taxonomy.

Only the first two classification methods were examined in this study, *i.e.*, the differences between *coverage* and *convergence*, and *movement-to* and *movement-while-maintaining*. All of the experimental tasks are in the *unknown-positions* category[2].

[2]The experimental methodology presented in Section 3 is just as appropriate for examining these four task types in the *known positions* category. Time restrictions did not allow this, however.

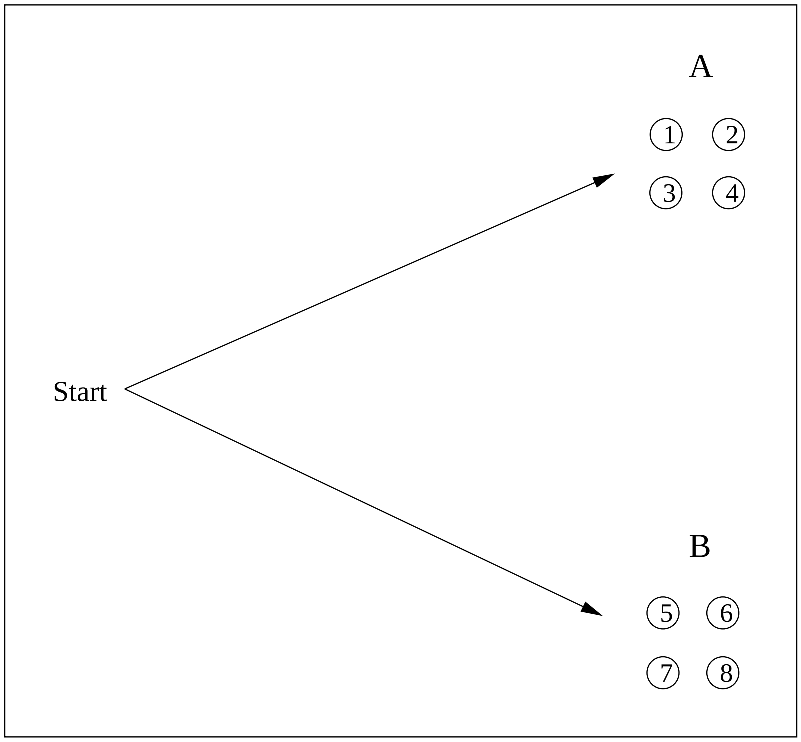


Figure 5.3: An example of the taxonomy applied recursively. Subgroups A and B, are moving apart from each other (*moving-to-coverage*). The robots in subgroups A and B exhibit both exhibit *movement-while-maintaining-convergence* within that subgroup.

Therefore, the experiments consider four different classes of tasks, including:

- *movement-to-coverage*[3]
- *movement-to-convergence*
- *movement-while-maintaining-coverage*
- *movement-while-maintaining-convergence*

Examples of these four classes of tasks are shown in Figure 5.2.

[3]Henceforward, the suffix *with-unknown-positions* will be omitted from the task classifications for brevity, since all of the experimental tasks had *unknown-positions*.

Table 5.2: Representative Tasks for Each Taxonomic Class.

Task Class	**Representative Task**
Movement-to-Coverage	Sentry Positioning
Movement-to-Convergence	Gathering to Perform Work
Movement-While-Maintaining-Coverage	Dragging a River Bottom
Movement-While-Maintaining-Convergence	Patrolling

5.2 Experimental Tasks

In order to compare the different MTSs for each taxonomic task class, experimental tasks must be chosen to represent each one. Table 5.2 lists the four tasks which were chosen. These experimental tasks are generic by nature (*i.e.*, they are idealized, simplified tasks designed to test a specific capability [26]), with each requiring movement that closely represents its class. They have been given names and descriptions, however, imitating real-life applications to make them more interesting to the participants, namely **Sentry Positioning**, **Gathering to Perform Work**, **Dragging a River Bottom**, and **Patrolling**. In all instances tested, there are four robots in the group. Likewise, in all cases, the human operator is given the same two guidelines: to complete the task as quickly as possible and with as few collisions as possible.

The **Sentry Positioning** task represents the *movement-to-coverage* class of tasks. Here, the human operator pretends that the robots are sentries, and he must move them from a starting location to positions spread out across an area to be guarded. The floor of the laboratory is divided into four quadrants, as shown in Figure 5.4. Initially, all four robots are in one of the quadrants. The operator's job is to move the robots so that there is exactly one robot in each of the four quadrants. Three obstacles are placed at predetermined locations to provide greater difficulty.

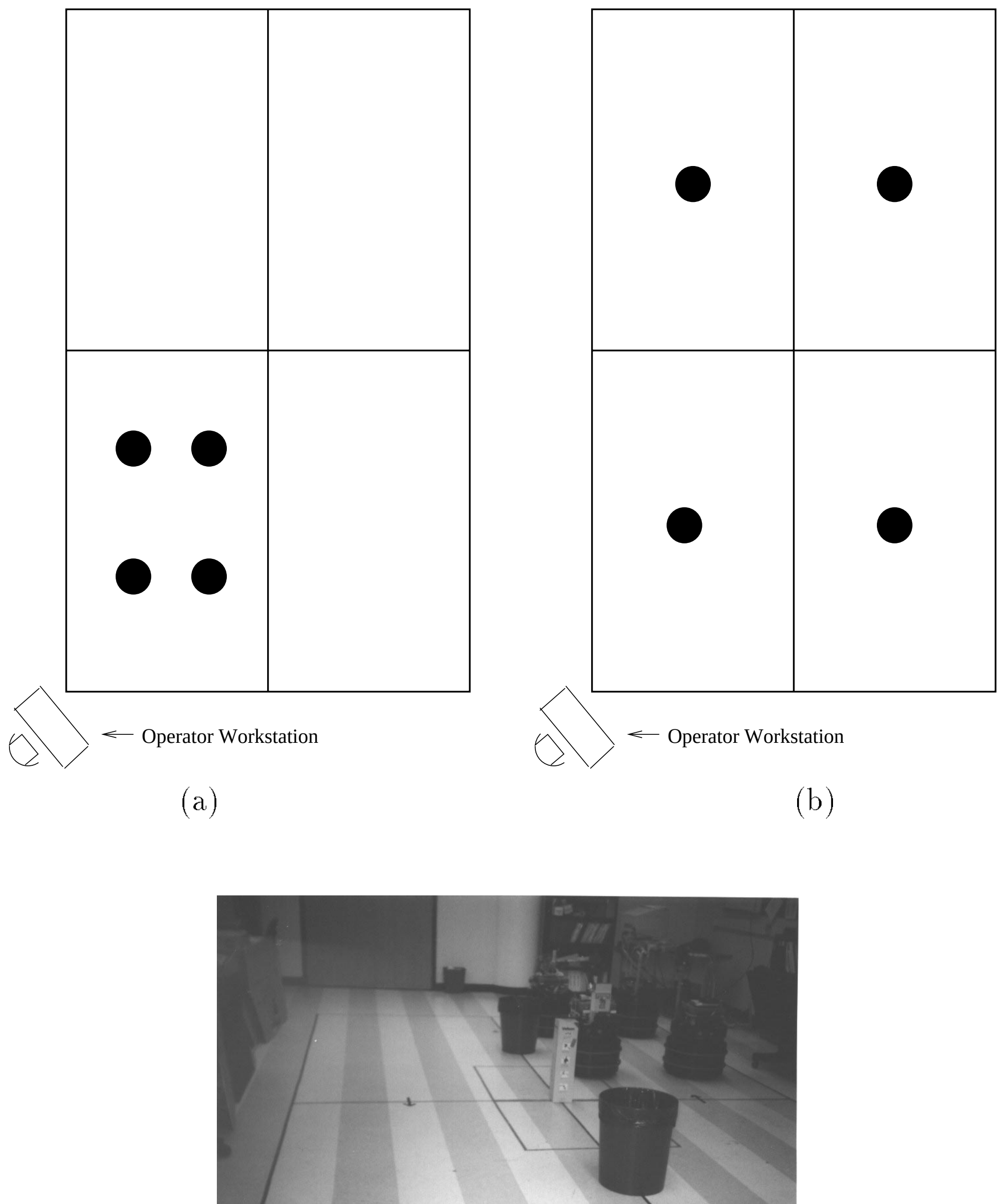
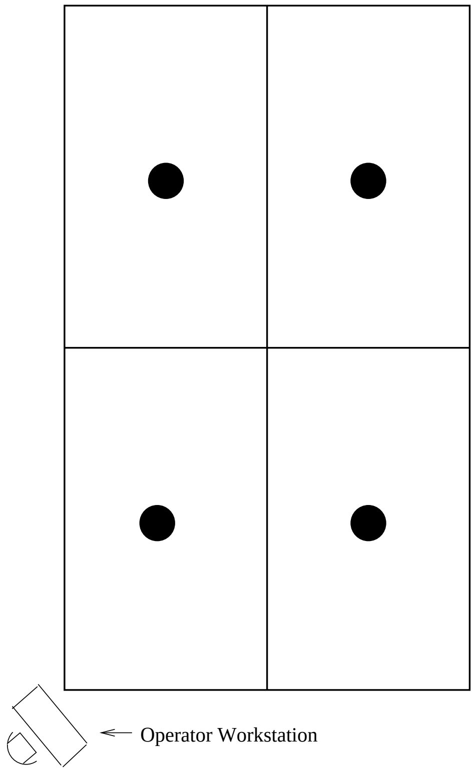


Figure 5.4: **Sentry Positioning** Task. The floor is divided and marked into quadrants for this task. (a) and (b) show approximate locations for the robots at the start and end of the task, respectively. The black dots represent robots. The photograph, (c), shows the initial setup of the robots and obstacles.

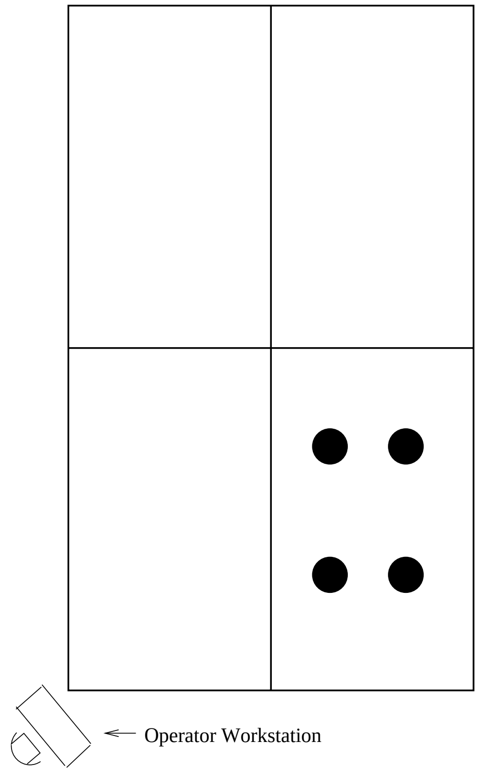
The **Gathering to Perform Work** task represents the *movement-to-convergence* class. This task is the opposite of the **Sentry Positioning** task, but uses the same quadrants and obstacles. Initially, there is a robot in the center of each of the four quadrants, each supposedly performing some work or monitoring machines in its area. The operator is told that one of the robots has discovered a malfunctioning machine at its location and has requested the help of the other three robots in fixing the machine. The operator's job is to move the robots such that all four robots are within the broken machine's quadrant at the same time, so that they can proceed to repair the machine (Figure 5.5).

The **Dragging a River Bottom** task represents the *movement-while-maintaining-coverage* class of tasks. In this task, the operator pretends that he is directing a group of boats down a river as they drag the bottom in search of something. The rectangular laboratory is used to simulate the river. Initially, the robots are spread out evenly in a line across the width of the "river". Each of the robots is in a lane marked on the floor that travels downstream, as shown in Figure 5.6. There are obstacles scattered about the "river". The human's job is to move the robots downstream (from one side of the laboratory to the other), and across a finish line at the far end of the "river", without letting any robots stray from their respective lanes.

The **Patrolling** task represents the *movement-while-maintaining-convergence* task class. The operator pretends that the robots are military scouting vehicles, and his job is to direct them through the path of their patrol route. The lab is again divided into four quadrants, as shown in Figure 5.7. Initially, all four of the robots are inside one of the quadrants. The human operator has to move them, as a group, through the other three quadrants in a particular order, and then back to the starting area. The order in which the quadrants should be visited causes the



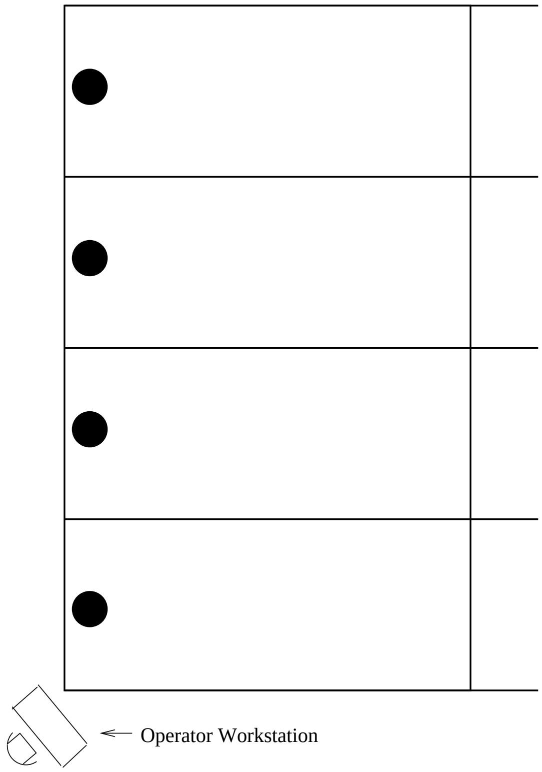
(a)



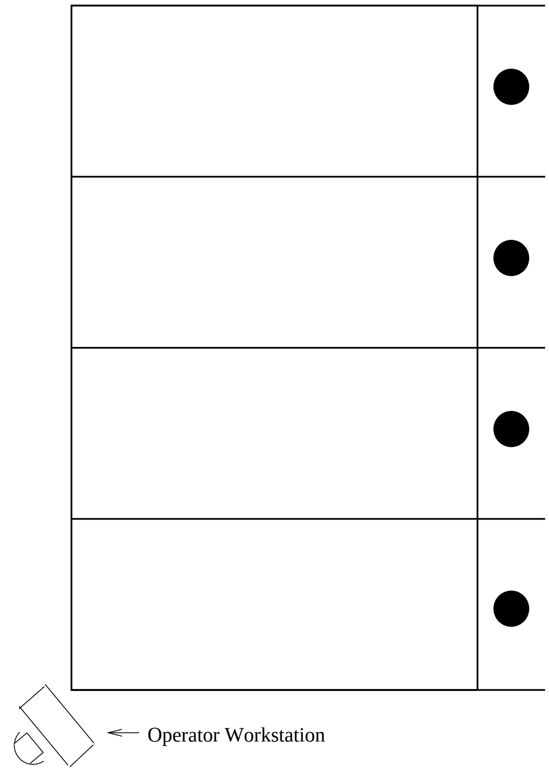
(b)

(c)

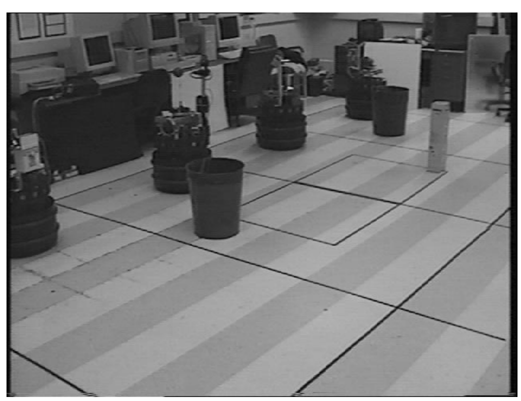
Figure 5.5: **Gathering to Perform Work** Task. The floor is divided and marked into quadrants for this task. (a) and (b) show approximate locations for the robots at the start and end of the task, respectively. The photograph, (c), shows the initial setup of the robots and obstacles.



(a)



(b)



(c)

Figure 5.6: **Dragging a River Bottom** Task. The floor is divided and marked into lanes for this task. (a) and (b) show approximate locations for the robots at the start and end of the task, respectively. The photograph, (c), shows the initial setup of the robots and obstacles.

robots to take a square path around the laboratory. Because the robots represent military vehicles, they have to stay close together during the patrol so that they can protect each other. In terms of the actual task, this means that as the operator is moving the robot group into a quadrant, none of the robots can proceed on to the next quadrant until all four are within this quadrant. As in the other tasks, there are obstacles placed in the task area to provide an extra challenge.

5.3 Summary

A taxonomy of mobile multiagent tasks was developed. This allows researchers to know what type of task they are evaluating their system for and if it is the same task type that other researchers have used. Additionally, a taxonomy helps experimenters to formally choose which tasks to evaluate and makes sure that no type is unintentionally ignored.

This dissertation's experiments examine four categories from the developed taxonomy, namely *movement-to-coverage*, *movement-to-convergence*, *movement-while-maintaining-coverage*, and *movement-while-maintaining-convergence*, all with *unknown-positions*. Experimental tasks were chosen to represent each of these classifications. These tasks are **Sentry Positioning**, **Gathering to Perform Work**, **Dragging a River Bottom**, and **Patrolling**, respectively.

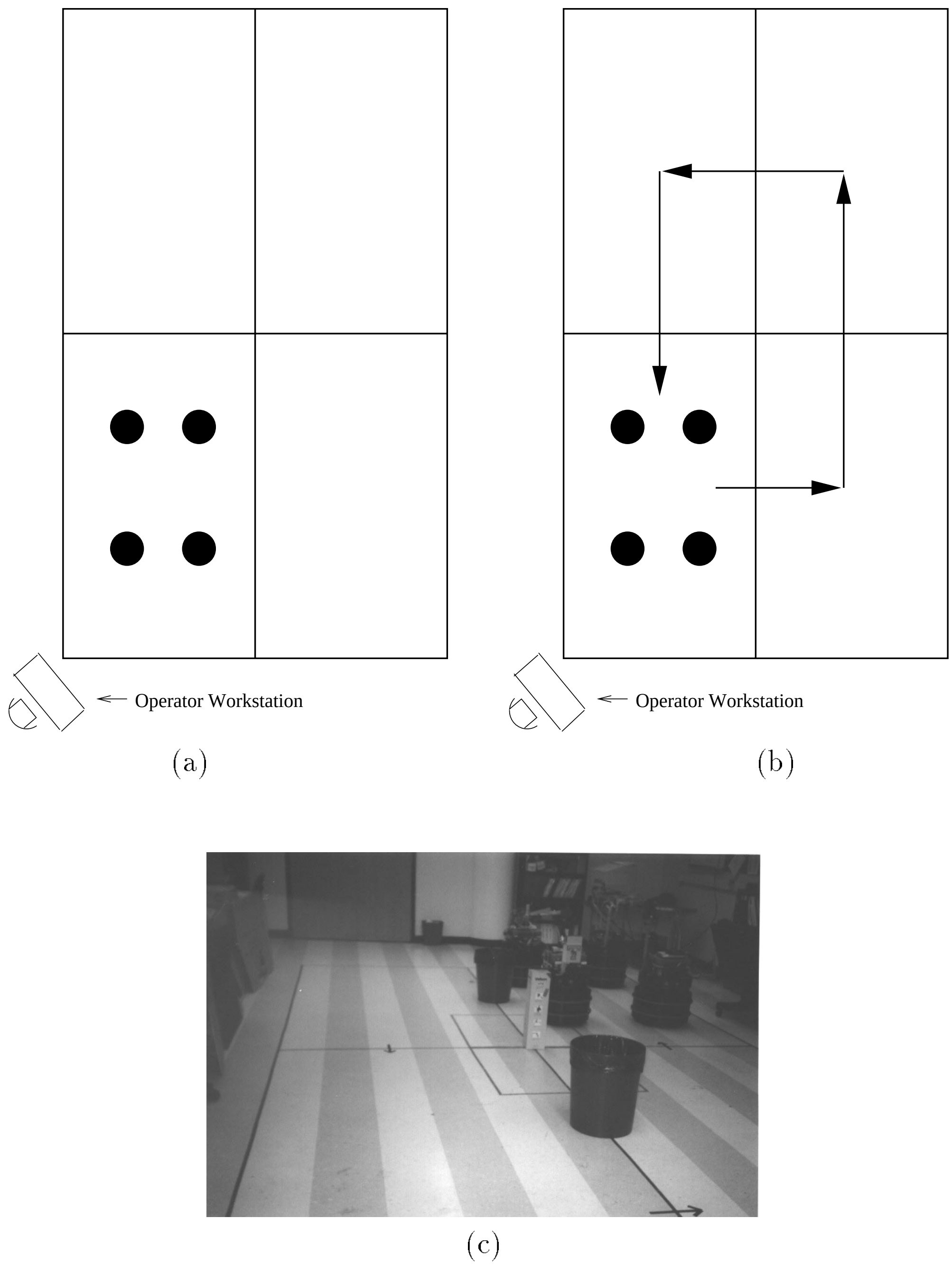


Figure 5.7: **Patrolling** Task. The floor is divided and marked into quadrants for this task. (a) shows the locations of the robots at the start. The robots take the path shown in (b) around through the quadrants, to end in the starting area again. The photograph, (c), shows the initial setup of the robots and obstacles.

Chapter 6

Experimental Design and Procedure

The goal of the experiments was to determine, for each task, which multiagent telerobotic system is safest, most effective, and easiest to use. There were two desired results from each of the experiments:

- system rankings for each task
- general relationships between system dimensions and tasks

The first result is rankings of the systems in terms of the performance criteria (safety, effectiveness, and ease-of-use) for each task. That is, for each task, there are three separate rankings, one for each of the three criteria. The second desired result includes more general findings relating the nature of the system to the performance for a task class. For example, one such finding is that individual control is more effective than group control for the *movement to coverage* class of task, regardless of whether supervisory or direct manual control is used. Chapter 7 explains how these results were generated from the experimental data. This chapter describes the experiments themselves, including the nature of the data and how it was collected.

6.1 Factors, Treatments, and Replications

The tests consisted of one two-factor experiment (see Chapter 7) for each of the four classes of tasks. The independent variables (*i.e.* those which can be manipulated by

```
Factors (independent variables)
   A: Level of autonomy
        a1: direct manual control
        a2: strict supervisory control
   N: Number of robots controlled at one time
        n1: individual control
        n2: group control

Responses (dependent variables)
   Safety:
      number of collisions
   Effectiveness:
      completion of task
      task completion time
   Ease of use:
      number of user actions
```

Figure 6.1: Experimental factors and responses. Four experiments were conducted, with one experiment for each of the examined tasks.

the experimenter) for each experiment are the level of autonomy of the robot system and the number of robots controlled at one time by the human operator (Figure 6.1). Two factor levels were examined for each of the factors. Thus, there were four treatments (the four system types) examined during each experiment. For each of the four treatments, six replications were conducted. That is, six participants, unique to that treatment, used each of the four systems. Therefore, 24 participants (four treatments by six replications) were used for each experiment. Since four experiments were conducted (one for each class of task), a total of 96 different participants were used. For each test, the subject attempted the task with one of the systems, and certain response values (described in the next section) were measured.

6.2 Response Variables for the Experiments

To determine how well the criteria (or response variables) of safety, effectiveness, and ease-of-use are met, certain values were measured during the experiments. Data values which could be gathered automatically by the operator interface and robot architectures were collected by that means. Other values, which could not be obtained automatically, were gathered by observation of the actual robots by the experimenter. Figure 6.1 shows the response variables and the corresponding data types that were gathered to determine the responses.

To determine the safety of a system configuration, the number of collisions (actual contact) between robots and other robots or obstacles was counted. Safer systems correspond to systems with fewer collisions.

To determine the effectiveness of a system configuration, the task completion time was recorded, as well as whether the operator was able to successfully complete the task. There are two ways in which the operator may have failed to complete the task:

- the task was not completed before the timeout time
- the robots failed to obey the rules of the task (such as staying within their lanes during the **Dragging the River Bottom** task)

Systems with faster completion times are regarded as more effective systems. For those systems in which the task was not completed, the timeout period is used as the task completion time for that trial when determining the mean completion time for the system. This permits the use of a quantitative value during the analysis, allowing for a more objective analysis. The drawback to this decision is that it does not differentiate between a task that was completed in just under the timeout period and one that was not completed. No better alternative was determined, however.

Another possibility considered was using twice the timeout period for computing mean completion time when the task was not completed. This distinguishes between tasks that were completed and ones that were not. It is possible, however, that the user may have been able to complete the task if given a little more time. This method may therefore exaggerate the mean completion time. Since the first method (using the timeout period for failed tasks) will keep the mean times closer together, it is less likely to indicate a difference between systems when there really is none. This more conservative method was chosen.

The number of user actions is used to determine the ease-of-use of a system configuration. User actions are mouse-clicks on any of the control windows. Systems with fewer user actions were deemed to be easier-to-use systems, because those systems required the human operator to do less work. At the extremes, a system that did not require the operator to do anything would be easiest to use, while a system that required the operator to continuously give instructions to the robots would be the most difficult to use. As with each of the criteria, other measures could be used to determine this value. User-interface studies often use a combination of subjective and objective measures to determine a system's ease-of-use. The subjective measures, such as asking the user how difficult it was to use the system, are valid techniques, but a more objective and scientific method was desired for these evaluations. Other examples of objective measures that are typically used include the task completion time and the number of user errors. In these experiments, measures such as task completion time were considered more appropriate for determining other criteria than ease-of-use. Here, the number of mouse clicks was chosen for the reasons stated previously (*i.e.*, systems requiring the least work by the human are considered easier to use).

6.3 Participant Selection and Resulting Population

The participants were gathered primarily through the use of postings on electronic bulletin boards and signs posted in a multidisciplinary student laboratory at the Georgia Institute of Technology. Additionally, some participants were gathered through word of mouth. The only two requirements for the participants were that they be at least sixteen years old and that they have experience using a computer mouse. Aside from these requirements, anyone who responded was used as a participant. Only one potential participant had to be rejected, and that was because the person had never used a computer mouse.

The resulting participant population was mostly male undergraduate students. Appendix E contains two tables showing the distribution of subjects across the tasks and systems in terms of sex, age, education, and experience with mobile robots. There were relatively few subjects who had prior experience with mobile robots. Those that had experience were evenly distributed across the four system types, and nearly evenly distributed across the four tasks. There was no attempt to assign particular participants to particular systems or tasks. The process of assigning participants to systems was random, and is described in the following section.

6.4 Experimental Procedure

The same experimental procedure was followed for each subject. Each experiment, corresponding to a task class, was conducted in its entirety before the next experiment was begun. This is due to the need to change the layout of the environment for each task. Within each task, however, the tests were conducted in six blocks of four, with a block consisting of one replication of each of the four systems. This was done to minimize any bias due to possible changes in the experimental procedure over time, as described in Section 3.4. Within each block, the **Direct Manual Individual** system was tested first, then **Direct Manual Group**, **Supervisory Individual**, and lastly **Supervisory Group** control. Participants were assigned to one of the four systems in the order they signed up for the tests, which was random.

The procedure followed during the testing is as follows. A checklist of this procedure appears in Appendix F.

1. The experimenter explained the purpose of the experiments and told the participant that the system is being tested and not himself. When the experimenter talked to each of the participants, he used a memorized speech (Appendix F) to eliminate differences in what each participant had been told. The participants, however, were allowed to ask questions, and the answers to these questions were obviously not from a memorized script. Also, the participants were allowed to ask questions at any time during the testing. All questions were answered to the best of the experimenter's ability, except for questions asking whether a robot was going to hit an obstacle or not. These questions were answered to make sure that the participant understood how the system worked. It is possible that the more inquisitive participants may have had an advantage over the others, due to a greater understanding of the system. Since the participants were pseudo-randomly assigned to the systems, however, based on the order in which they showed up to use the systems, in principle each system was tried by an adequate mix of inquisitive and non-inquisitive participants.

2. The participant read and signed a consent form (included in Appendix F) that has been approved by the Georgia Institute of Technology Institutional

Review Board.

3. The participant filled out a survey (Appendix F), which asked general questions regarding their sex, age, level of education, and experience using a computer mouse and mobile robots.

4. The participant was taught, by explanation and demonstration, to use the controls for the system that they would be using to control the robots. Each participant was taught to use the controls for only one of the four types of systems being tested, since each participant only used one system. The participant practiced using the controls on simulated robots for ten minutes. He was allowed more time at the end of the ten minutes if he still did not feel comfortable using the system, but no participant wanted to continue practicing.

5. The user tried to complete a sample task, involving navigating a group of robots from one point to another around a box-canyon, using simulated robots.

6. The real task was explained to the participant. This explanation included the goal of the task, the timeout period, and the guidelines for the task, as shown in the script in Appendix F. Each task has a timeout period. If the human operator did not complete the task within the specified time, then the robots were stopped, and the task was counted as incomplete. The timeout period for the tasks was ten minutes, except for the **Patrolling** task, which had a twenty minute timeout due to the longer distance that the robots must travel to complete the task. The guidelines were the same for each task. The operator was asked to complete the task as quickly as possible and with as few collisions as possible.

Figure 6.2: The four robots used in the experiments.

7. The participant attempted the task with four real Nomad 150 robots (Figure 6.2), produced by Nomadic Technologies, Inc. The four robots are homogeneous, with the same hardware and control software. The robots are three wheeled and near-holonomic, since all three wheels turn together. Each has a ring of sixteen ultrasonic sensors, which were used by the **Supervisory** control systems for obstacle avoidance. The robots also have a ring of tactile sensors, which were unused during these experiments. The control software is described in Section 4.3.

The experimenter monitored the task to determine when the task had been completed, whether the timeout period had been exceeded, and to count the number of collisions between the robots and obstacles or other robots. After the task had been completed, or timed out, the experimenter informed the participant and shut down the system.

The participants had to depend on their own sight and the feedback from the robots and computer to determine the robots' positions. The participants were allowed to stand up behind the operator workstation table, but they were not allowed to come out from behind the table. The obstacles were small enough that the operator could see the tops of all robots at all times.

Chapter 7

Data Analysis Techniques

Two methods were used to analyze the data gathered during the experiments. To produce the system rankings, a one-factor ANOVA (ANalysis Of VAriance) analysis was conducted. This type of analysis compares the mean values of data from multiple sources which differ in one way (in this case the type of system). To determine if more general results existed, a two-factor ANOVA analysis was conducted. This analysis technique compares the mean values from sources that differ in two ways (in this case, the level of autonomy and the number of robots controlled). The following sections explain these two analysis techniques more fully, and explain how they apply to the conducted experiments. An example analysis is then presented.

7.1 Single-Factor Analysis to Produce Rankings

One of the goals of each experiment was to determine how the different types of systems compare for the task in terms of each of the three judgment criteria: safety, effectiveness, and ease-of-use. Therefore, three rankings of the four types of systems are needed for each task, one for each of the criteria. To produce these rankings, we can compare the means of the data values collected for all of the replications. For example, to determine the safety of a particular system for a task, the number of collisions between robots and obstacles or other robots was counted. Since, there were six replications for each system/task combination (*i.e.*, six different people used each system for a particular task), there are six values representing the safety for each system/task combination. The mean of the six values for one particular system can be used to compare it to other systems. In this way, rankings can be produced.

It is necessary, however, to determine if the differences between the mean values representing the safety of each system are statistically significant. That is, we must determine if the probability that these differences could have been a result of random noise is greater than some predetermined ratio.

As discussed in Chapter 3.5, the single-factor ANOVA model is an appropriate model [41] for studying the relationship between the predictor variable (in this case, the type of system) and the response variable (*i.e.*, the safety, effectiveness, and ease-of-use ratings of the systems). In other words, we can use the single-factor ANOVA model to determine if there are any statistically significant differences between the means for each system, or if the means are probabilistically equivalent, with any differences detected due to noise. If any differences between systems are detected, the single-factor ANOVA model will also allow us to determine what those differences are. Figure 7.1 shows the process used to produce the rankings of the systems for each criteria and class of task. The procedures and tests mentioned in this figure are explained in the following sections, as well as in [41].

7.1.1 Insuring normality of error values and constancy of error variance

The ANOVA model assumes normality of error values for each sample (*i.e.*, the frequency of the differences between the measured values and the true mean fit a normal distribution), and constancy of error variance between samples (*i.e.*, the magnitude of the differences is approximately the same between data sets). If this

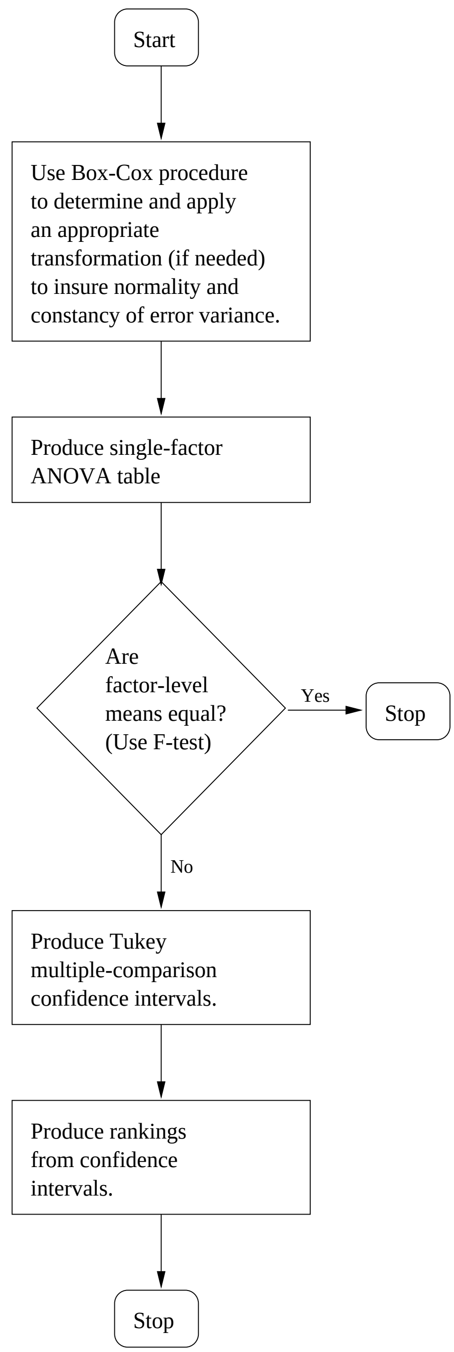


Figure 7.1: Flowchart of the process to produce the system rankings.

is not the case with the actual data values, then a transformation must be used to insure these conditions are met. The Box-Cox [41] procedure identifies a power transformation of the form Y^λ (where Y is the data set) to correct for both lack of normality and non-constancy of error variance. If no transformation is needed to meet either of these conditions, then the Box-Cox procedure indicates that $\lambda = 1$.

The Box-Cox procedure was applied to the data from each experiment. If the procedure indicated that a transformation was needed, then that transformation was used on the data. Each of the λ values determined by the Box-Cox procedure for the transformation of the experimental data sets is presented in Appendix D.

7.1.2 ANOVA table and test for equality of factor-level means

For each experiment, Matlab [38] was used to produce a single-factor ANOVA table using the data. The F^* value shown in this table can be used to determine whether the factor level means are equal or not. F^* is defined as

$$F^* = \frac{MSTR}{MSE}$$

where $MSTR$ is the treatment mean square and MSE is the error mean square [41][1].

F^* follows the F distribution (a standard probabilistic distribution) when H_0 holds (H_0 is the hypothesis that the factor level means are equal), and does not follow the F distribution when H_a holds (H_a is the hypothesis that the factor level means are not equal) [41]. More specifically, F^* is distributed as $F(r-1, n_T - r)$

[1]A discussion of the derivation of $MSTR$ and MSE is beyond the scope of this dissertation. Furthermore, it is not necessary to know how these values are derived to conduct the analysis for these experiments, as most statistical software packages will generate these values from the data. Readers who desire a further explanation of these values are referred to [41].

when H_0 holds, where r is the number of factor levels and n_T is the total number of replications conducted across all factor levels. In these experiments, the factor levels were the four types of systems, so $r = 4$, and there were 6 replications conducted for each of the factor levels, so $n_T = 24$.

The appropriate decision rule [41] to control the level of significance[2] at α is:

If $F^* \leq F(1-\alpha; r-1, n_T - r)$,
then conclude H_0,
else conclude H_a.

where $F(1-\alpha; r-1, n_T - r)$ is the $(1-\alpha)100$ percentile of the appropriate F distribution.

For the tests conducted on the data from these experiments, α was chosen to be 0.05, thereby ensuring that if we conclude H_a, then we can be 95% certain that the factor level means are really not equal. By consulting a table of F distributions, we find that $F(0.95; 3, 20) = 3.10$. So, if the F^* value indicated in the ANOVA table is less than or equal to 3.10, then we conclude that the factor level means are not significantly different. This result would indicate that the systems all performed equivalently for that particular class of task. If the F^* value is greater than 3.10, then we conclude that the factor level means are significantly different. This would mean that at least one of the systems performed differently than the others for that task class. The F^* values derived from each experimental data set are presented in Appendix D.

7.1.3 Determining the confidence intervals

If the F-test indicates that the systems do not all perform equivalently for the task, then the next step is to determine how each system ranked relative to the others.

[2] α is the level of significance. If an α value of 0.05 is used in the decision rule, and H_a was concluded, then we can state, with 95% certainty (1-α), that H_a actually is true.

This involves conducting multiple comparisons between the factor level means. The Tukey multiple comparison procedure [41] was used to insure that the entire family of comparisons retained the 95% confidence, rather than just each individual comparison. This procedure is used to produce family confidence intervals, which are the same around each sample factor level mean. The confidence interval for each mean indicates that, with 95% certainty, the true factor level mean lies within that interval.

The confidence intervals for each estimated treatment mean (*i.e.* the sample mean for each treatment), $\overline{Y}_{i\cdot}$[3], has the limits

$$\overline{Y}_{i\cdot} \pm \frac{1}{2} T s\{\hat{D}\} \tag{7.1}$$

where

$$T = \frac{1}{\sqrt{2}} q(1-\alpha; r, n_T - r)$$

with q being the studentized range distribution (a standard probabilistic distribution), and α, r, and n_T defined the same as in the decision rule in Section 7.1.2. When the number of replications for each treatment are the same, then

$$s\{\hat{D}\} = \sqrt{\frac{2}{n} MSE}$$

with MSE being the error mean square (which can be found in the ANOVA table) and n being the number of replications per treatment [41]. For this research,

$$\begin{aligned} T &= \frac{1}{\sqrt{2}} q(1-\alpha; r, n_T - r) \\ &= \frac{1}{\sqrt{2}} q(1-0.05; 4, 24-4) \\ &= \frac{1}{\sqrt{2}} q(0.95; 4, 20) \\ &= 2.800 \end{aligned}$$

[3]The dot notation (*e.g.* $\mu_{1\cdot}$) indicates a summation of all the values of the variable along that index.

and

$$
\begin{aligned}
s\{\hat{D}\} &= \sqrt{\frac{2}{n} MSE} \\
&= \sqrt{\frac{2}{6} MSE} \\
&= \sqrt{\frac{1}{3} MSE}
\end{aligned}
$$

Therefore the confidence intervals for each experiment are

$$\overline{Y}_{i\cdot} \pm \frac{1}{2}(2.8)\sqrt{\frac{1}{3} MSE}$$

or

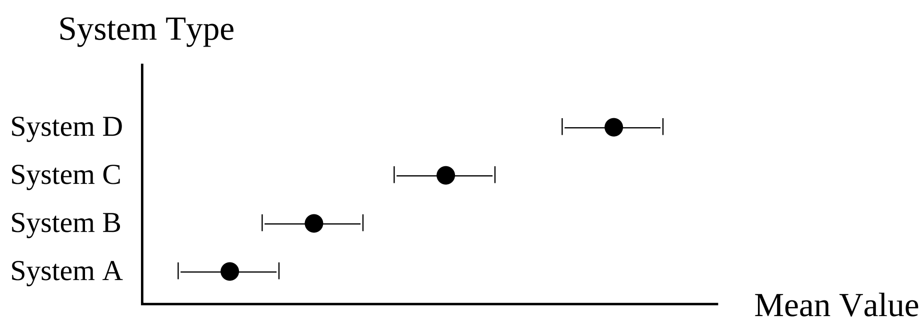
$$\overline{Y}_{i\cdot} \pm 0.808\sqrt{MSE} \tag{7.2}$$

where the MSE value is different for each set of experimental data, and can be obtained from the ANOVA table for that data set.

The rankings of the four types of systems can be generated from these confidence intervals. If and only if the confidence intervals for two or more factor levels do not overlap, then those factor levels are considered different. Figure 7.2 shows an example set of confidence intervals and the resulting ranking of the systems. Appendix D presents the confidence intervals for the experimental data sets in which the F test did not find all the means equivalent.

7.2 Two-Factor Analysis to Identify Principles

A second goal of the experiments was to determine, where possible, general results relating a system dimension to a class of task. These findings would relate the levels along one of the two dimensions (amount of autonomy and number of robots controlled at a time) with the task class, regardless of the level of the other dimension.



(a)

Ranking

1. System A
   System B
2. System C
3. System D

(b)

Figure 7.2: Confidence intervals and rankings. (a) shows a set of possible confidence intervals, and (b) shows the ranking of systems that would be derived from these intervals.

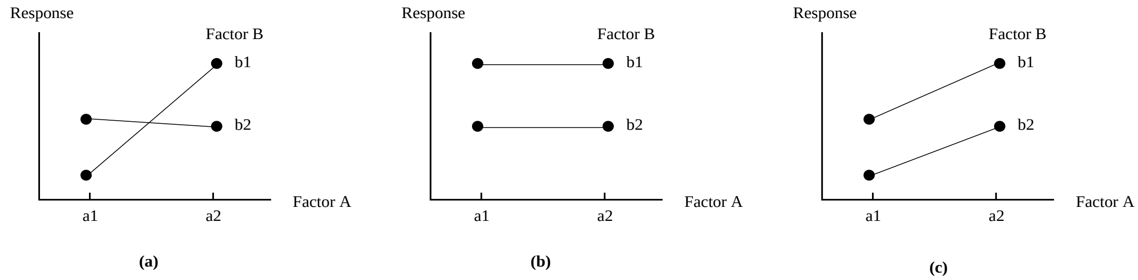


Figure 7.3: Factor Main Effects. The vertical axis shows the response value, the horizontal axis shows the level of Factor A, and the two different lines represent the level of Factor B. (a) shows a situation in which there are no main effects. In (b), factor B shows a main effect, and in (c), both factor A and B show main effects.

This corresponds to searching for factor main effects in a two factor analysis. A main effect is the difference between a factor level mean and the overall mean for all levels of that factor [41].

Thus we look for situations where all of the factor level means for one factor are higher than the corresponding factor level means for the other dimension. For instance, Figure 7.3 shows three examples, two of which demonstrate main effects. Figure 7.3(a) does not exhibit any main effect, since there is no factor level that has a higher response than the other level for the same factor, regardless of the level of the other factor. Figure 7.3(b), however, shows a main effect in factor B, because, regardless of what level factor A is at, factor level b1 shows a higher response than factor level b2. Figure 7.3(c) shows a main effect in both the A and B factor, since factor level b1 shows a higher response than factor b2, regardless of the setting of factor A, and factor level a2 shows a higher response than level a1, regardless of the setting of factor B.

The procedure [41] used to determine if factor main effects were present is described in the following sections. Figure 7.4 shows the decision process that was used.

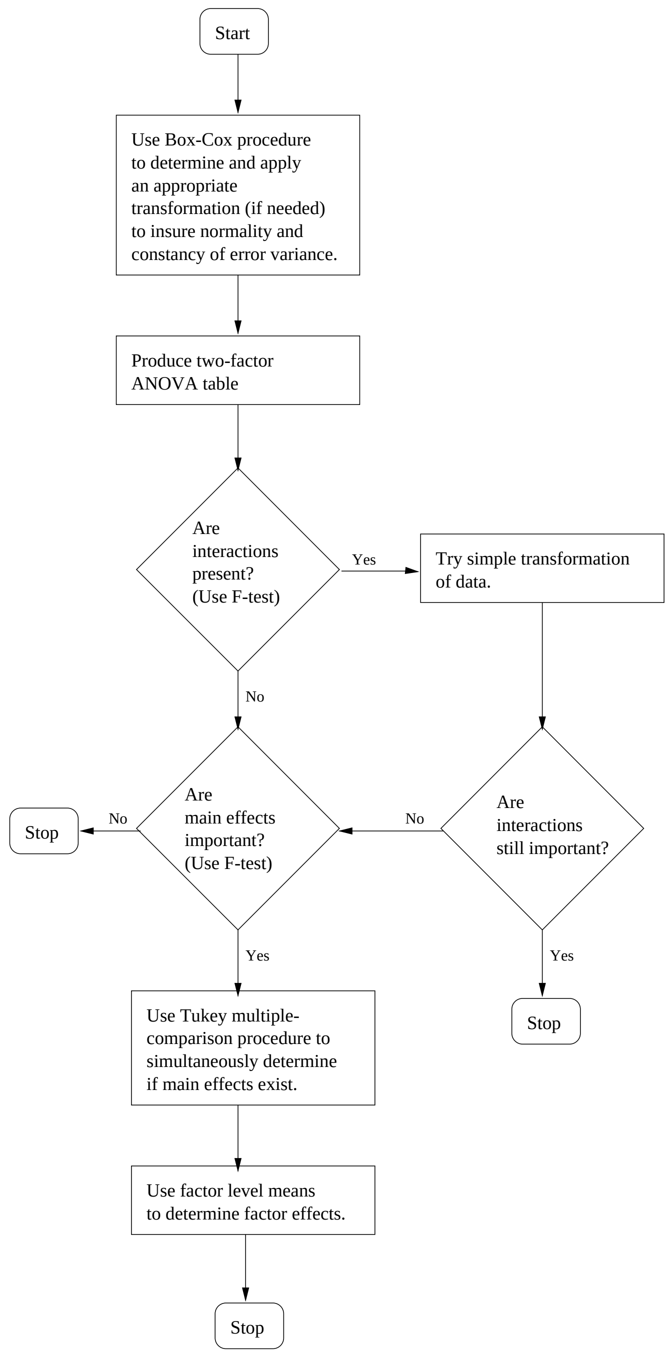


Figure 7.4: Flowchart of the process to produce the general findings.

The Box-Cox procedure is used first on the data to determine if any transformation was needed, and then determine the appropriate transformation (of the form Y^{λ}, where Y is the data set, and λ is determined by the Box-Cox procedure), to ensure normality of the data for each treatment and constancy of error variance between treatments (see Section 7.1.1).

Matlab [38] was then used to produce a two-factor ANOVA table for each of the three criteria, and for each task type. This table has three F^* values. The first is used to determine if there is a column main effect. This would indicate a main effect due to the number of robots being controlled at a time. The second F^* value is for the row main effect, indicating whether there is a main effect due to the level of autonomy of the robots. And the third F^* value indicates whether there are interactions between the factor effects.

Interacting factor effects exist when the factor effects are not additive. Figure 7.5(a) shows factor effects that do not interact, and Figure 7.5(b) shows factor effects that do interact. In Figure 7.5(b), we see that Factor B has no effect on the response when Factor A is set at a1, but it does have an effect when Factor A is set at a2. This varying influence of Factor B at its different levels indicates that Factors A and B interact, which is called an AB interaction. Interactions can be detected visually when the lines in the graph are not parallel.

More specifically, two factors interact when not all treatment means, μ_{ij}, can be expressed according to

$$\mu_{ij} = \mu_{..} + \alpha_i + \beta_j$$

where $\mu_{..}$ is the overall mean response value for all treatments, α_i is the main effect for factor A when at level i, and β_j is the main effect for factor B when at level j. When there are important interactions between factor effects, one should not ordinarily examine the effects of each factor separately in terms of the factor level

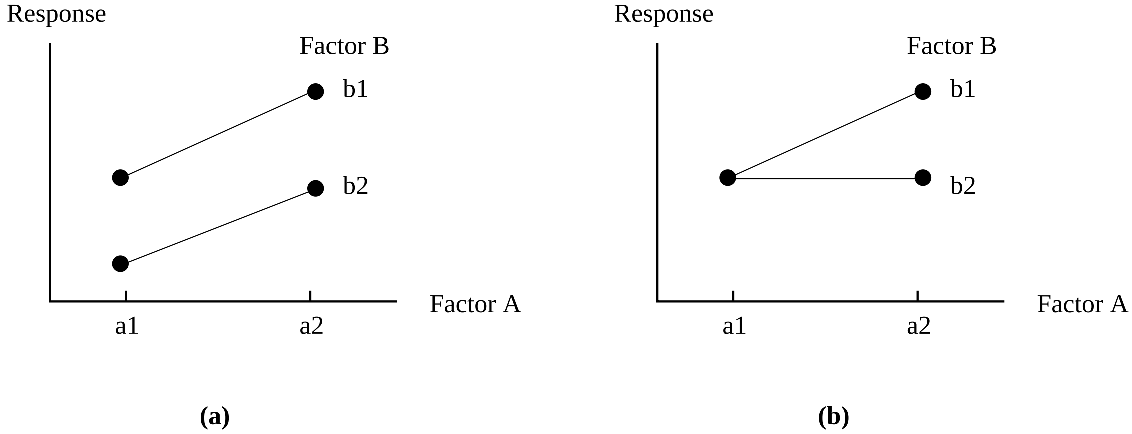


Figure 7.5: Demonstration of Interactions between Factor Effects. (a) does not show any interaction, while (b) does.

means [41]. Therefore, if there are interactions, no general results can be identified.

In certain cases when interactions are identified, a simple transformation may be used to remove or reduce the effect of the interaction until it is negligible. An example is factor effects that act multiplicatively, rather than additively:

$$\mu_{ij} = \mu_{..}\alpha_i\beta_j$$

instead of

$$\mu_{ij} = \mu_{..} + \alpha_i + \beta_j$$

This interaction can be removed by applying a logarithmic transformation:

$$\mu'_{ij} = \log \mu_{ij}$$

$$\mu'_{..} = \log \mu_{..}$$

$$\alpha'_i = \log \alpha_i$$

$$\beta'_j = \log \beta_j$$

This results in the following:

$$\mu'_{ij} = \mu'_{..} + \alpha'_i + \beta'_j$$

The interaction is transformable and, thus, unimportant. In this case, the analysis can proceed with the transformed data.

7.2.1 F-Tests for interactions and main effects

The F^* value corresponding to interactions in the ANOVA table is derived from

$$F^* = \frac{MSAB}{MSE}$$

where $MSAB$ is the AB interaction mean square and MSE is the error mean square. Large values of this F^* indicate the existence of interactions. The appropriate decision rule [41] to control the level of significance at α is

$$\boxed{\begin{array}{l}\text{If } F^* \leq F[1-\alpha; (a-1)(b-1), (n-1)ab],\\ \text{then conclude } H_0,\\ \text{else conclude } H_a.\end{array}}$$

where a is the number of levels of factor A, b is the number of levels of factor B, n is the number of replications conducted per treatment, H_0 indicates that there are no interactions, and H_a indicates that interactions exist.

Since $\alpha = 0.5$ was used in this analysis,

$$F[1-\alpha; (a-1)(b-1), (n-1)ab] = F[0.95; 1, 20] = 4.35.$$

So, if the F^* value for interactions in the ANOVA table is greater than 4.35, then there are interactions present, and the search for general findings relating factors to the task class cannot proceed unless a simple transformation can be used to remove those interactions.

If there are no important interactions present, then the next step is to use a pair of F-tests to determine if any main effects are important. The F^* value corresponding to columns in the ANOVA table indicates whether there is a main effect due to factor A (the number of robots controlled at a time). This F^* value is derived from

$$F^* = \frac{MSA}{MSE} \tag{7.3}$$

where MSA is the factor A mean square. Once again, large values of F^* indicate the presence of factor A main effects. The decision rule for controlling the level of significance at α is

If $F^* \leq F[1-\alpha; a-1, (n-1)ab]$,
then conclude H_0,
else conclude H_a.

where H_0 indicates that the factor A main effect is not important, and H_a indicates that the factor A main effect is important.

Similarly, the F^* value corresponding to rows in the ANOVA table indicates whether there is a main effect due to factor B (the level of autonomy of the robots). This F^* value is derived from

$$F^* = \frac{MSB}{MSE} \tag{7.4}$$

similar to Equation 7.3, except substituting MSB (factor B mean square) for MSA. The decision rule is

If $F^* \leq F[1-\alpha; b-1, (n-1)ab]$,
then conclude H_0,
else conclude H_a.

Since both a and b were 2 in this research, the F-test values are identical. So

$$F[1-\alpha; a-1, (n-1)ab] = F[1-\alpha; b-1, (n-1)ab] = F[0.95; 1, 20] = 4.35.$$

Appendix D presents the F^* values for the AB interaction, factor A effect (column effect), and the factor B effect (row effect) for each set of experimental data.

7.2.2 Tukey multiple comparison procedure to test for main effects

Since more than one comparison is being made for factor main effects, a multiple comparison procedure must be used to determine whether the family of comparisons maintains the α level of significance. So, the F-tests for main effects simply serve to indicate whether further testing is needed.

If the F-tests indicated that the main effects were important, then the Tukey multiple comparison procedure is used to conduct the simultaneous tests. This procedure is similar to the Tukey procedure described for single-factor ANOVA tests. In this research, however, there is no need to determine the confidence intervals. Since there are only two levels for each factor, if a difference is indicated, then the factor level means indicate which is greater. The test procedure [41] is described below.

The test statistic, q^* will be used in the decision rule to determine main effects. To test for a factor A main effect,

$$q^* = \frac{\sqrt{2}\hat{D}}{s\{\hat{D}\}} \tag{7.5}$$

$\hat{D}$ and $s\{\hat{D}\}$ are defined as follows:

$$\hat{D} = \overline{Y}_{i..} - \overline{Y}_{i'..}$$

where $\overline{Y}_{i..}$ and $\overline{Y}_{i'..}$ are the total calculated mean values for the two different levels of factor A, namely **Individual** and **Group** control.

$$s\{\hat{D}\} = \sqrt{\frac{2MSE}{bn}} \tag{7.6}$$

The decision rule for determining factor A main effects is

$$\boxed{\begin{array}{l} \text{If } |q^*| \leq q[1-\alpha; a, (n-1)ab], \\ \text{then conclude } H_0, \\ \text{else conclude } H_a. \end{array}}$$

where H_0 indicates that no factor A main effect is present, and H_a indicates that one is present.

Similarly, to test for a factor B main effect, compute q^* with Equation 7.5, except use the following $\hat{D}$ and $s\{\hat{D}\}$.

$$\hat{D} = \overline{Y}_{.j.} - \overline{Y}_{.j'.} \tag{7.7}$$

where $\overline{Y}_{.j.}$ and $\overline{Y}_{.j'.}$ are the total calculated mean values for the two different levels of factor B, namely **Direct Manual** and **Supervisory** control.

$$s\{\hat{D}\} = \sqrt{\frac{2MSE}{an}} \tag{7.8}$$

just as in Equation 7.6, except replacing b with a. Similarly, with the decision rule, replace a with b:

$$\boxed{\begin{array}{l} \text{If } |q^*| \leq q[1-\alpha; b, (n-1)ab], \\ \text{then conclude } H_0, \\ \text{else conclude } H_a. \end{array}}$$

As mentioned earlier, in this research, both a and b, the number of levels of factors A and B respectively, are 2. Therefore, the q value to be tested against is the same in both cases, namely:

$$q[1-\alpha; a, (n-1)ab] = q[1-\alpha; b, (n-1)ab] = q[0.95; 2, 20] = 2.95$$

If a factor A main effect is present, then we can look at the means for the factor levels, **Individual** and **Group** control, to determine which performs best for this task, based on the particular criteria (safety, effectiveness, or ease-of-use) currently being examined. If a factor B main effect is present, we determine, in the same way, whether **Direct Manual** or **Supervisory** control performs best based on the examined criteria. For example, one result might be that **Supervisory** control is safer than **Direct Manual** control for one type of task, regardless of whether **Individual** or **Group** control is being used.

Some of the F^* values for particular data sets in this research indicated that a main effect should be checked for. In this case, the q^* value for that test is presented in Appendix D.

7.3 Example

The following example shows how this process was used to analyze the data for the *movement-to-coverage* experiment conducted for this research. As described in Chapter 5.2, **Sentry Positioning** was the experimental task for this evaluation. As explained in Chapter 6, this experiment compared four systems (**Direct Manual Individual** (DI), **Direct Manual Group** (DG), **Supervisory Individual** (SI), and **Supervisory Group** (SG)) which differed along two dimensions (the amount of autonomy and the number of robots controlled at a time). The collected data for this experiment is listed in tabular form in Appendix C and is duplicated in Table 7.1 for convenience.

Unlike some of the other tasks, the only way to fail to complete this task was to exceed the time limit. Therefore, the task completion time is already set to the timeout period in every instance where the subject did not complete the task. So, the task completion time can be used directly as a measure of the effectiveness, with no extra processing required to combine the two measures into one quantitative measure, as described in Chapter 6.2. The number of collisions is used as the safety rating, and the number of user actions is used as the ease-of-use rating.

Table 7.1: *Movement-to-Coverage* Data Values

Number of collisions						
Direct Manual, Individual	0	0	0	0	0	0
Direct Manual, Group	12	7	6	5	12	6
Supervisory, Individual	0	0	0	0	0	0
Supervisory, Group	0	0	1	0	0	0
Completion of the task (Y = Completed, N = Incomplete)						
Direct Manual, Individual	Y	Y	Y	Y	Y	Y
Direct Manual, Group	N	N	N	N	N	N
Supervisory, Individual	Y	Y	Y	Y	Y	Y
Supervisory, Group	Y	N	N	Y	Y	N
Task completion time (seconds)						
Direct Manual, Individual	212	181	237	253	196	179
Direct Manual, Group	600	600	600	600	600	600
Supervisory, Individual	198	308	500	534	221	239
Supervisory, Group	552	600	600	599	539	600
Number of user actions						
Direct Manual, Individual	47	26	32	51	42	36
Direct Manual, Group	45	40	108	49	55	77
Supervisory, Individual	32	58	80	82	56	30
Supervisory, Group	45	65	67	53	80	73

Table 7.2: Transformed *Movement-to-Coverage* Data Values

Safety						
Direct Manual, Individual	0	0	0	0	0	0
Direct Manual, Group	12	7	6	5	12	6
Supervisory, Individual	0	0	0	0	0	0
Supervisory, Group	0	0	1	0	0	0
Effectiveness)						
Direct Manual, Individual	212	181	237	253	196	179
Direct Manual, Group	600	600	600	600	600	600
Supervisory, Individual	198	308	500	534	221	239
Supervisory, Group	552	600	600	599	539	600
Ease-of-use						
Direct Manual, Individual	0.6814	0.7219	0.7071	0.6749	0.6881	0.6988
Direct Manual, Group	0.6834	0.6915	0.6261	0.6776	0.6698	0.6477
Supervisory, Individual	0.7071	0.6663	0.6452	0.6436	0.6686	0.7117
Supervisory, Group	0.6834	0.6587	0.6567	0.6723	0.6452	0.6511

We will consider the four systems to be levels of a single factor and perform a one-factor ANOVA analysis. The first step is to insure that the error values of each data set fit a normal probability distribution and that the error variance is constant across data sets using the Box-Cox procedure (Section 7.1.1). The Box-Cox[4] procedure indicated that $\lambda = 1$ for both the safety and the effectiveness ratings, and that $\lambda = -0.1$ for the ease-of-use rating. This indicates that no transformation is needed for the first two ratings, and a transform of $Y^{-0.1}$ is needed for the ease-of-use rating. Table 7.2 shows the data after the transformation.

The next step is to produce a single-factor ANOVA table for each criteria (Tables 7.3, 7.4, and 7.5). Most statistical software can automatically create this table from the data (Matlab [38] produced these tables). The F^* values in the tables are used in the test to determine if all the factor level means are equal. In Section 7.1.2, the

[4]Consult [41] for details on how the Box-Cox procedure is performed.

Table 7.3: *Movement-to-Coverage* Single-factor ANOVA Table for Safety. The only values important to us are the F^* value and the MSE value (in the Mean Square column and Error row).

Source	Sum of Squares (SS)	degrees of freedom (df)	Mean Square (MS)	F^*
Factor A	284.1	3	94.71	37.26
Error	50.83	20	2.542	
Total	335	23		

Table 7.4: *Movement-to-Coverage* Single-factor ANOVA Table for Effectiveness. The only values important to us are the F^* value and the MSE value (in the Mean Square column and Error row).

Source	Sum of Squares (SS)	degrees of freedom (df)	Mean Square (MS)	F^*
Columns	6.587e+05	3	2.196e+05	37.5
Error	1.171e+05	20	5855	
Total	7.758e+05	23		

following decision rule was determined for this test:

If $F^* \leq 3.10$,
then conclude H_0,
else conclude H_a.

Since $F^* = 37.26$ for the safety and $F^* = 37.5$ for effectiveness, we conclude that not all the means are equivalent for these two criteria. $F^* = 2.724$ for the ease-of-use, however, indicating that all four systems performed equivalently for this criteria. Therefore, no system ranking can be determined for the ease-of-use.

Because the means for safety and effectiveness are not all equivalent, multiple-comparison confidence intervals are used to determine what the differences are. In Section 7.1.3, the confidence intervals were determined to be $\overline{Y}_{i\cdot} \pm 0.808\sqrt{MSE}$

Table 7.5: *Movement-to-Coverage* Single-factor ANOVA Table for Ease-of-use. The only values important to us are the F^* value and the MSE value (in the Mean Square column and Error row).

Source	Sum of Squares (SS)	degrees of freedom (df)	Mean Square (MS)	F^*
Columns	0.004061	3	0.001354	2.724
Error	0.009937	20	0.0004969	
Total	0.0014	23		

Table 7.6: *Movement-to-Coverage* Safety Confidence Intervals.

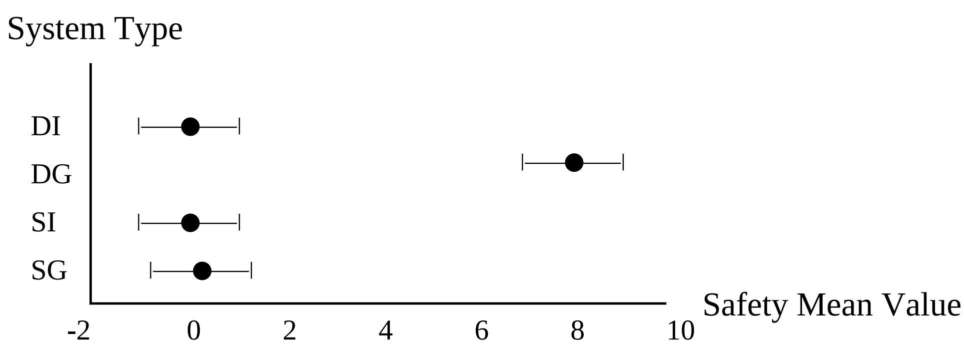
System	Mean	Lower Boundary	Upper Boundary
DI	0.0	-1.2887	1.2887
DG	8.0	6.7113	9.2887
SI	0.0	-1.2887	1.2887
SG	0.2	-1.1220	1.4554

(Equation 7.2). The MSE value can be obtained from the ANOVA tables (Tables 7.3 and 7.4), from the MS column and the Error row. This produces the confidence intervals in Tables 7.6 and 7.7 (depicted visually in Figures 7.6 and 7.7). The system rankings can be determined from the confidence intervals and means. Lower means indicate better systems. Any confidence intervals that overlap are considered equivalent and are grouped together in the ranking. There will be at least two (and at most four) non-overlapping sets of systems, since this step is not performed if the F-test indicated that all systems were equivalent.

The next step is to determine if there are any general results relating a point in a system dimension to a task class. To do this we conduct a two-factor ANOVA analysis, considering each of the system dimensions, the level of autonomy and the number of robots controlled at a time, as factors A and B respectively. We have

Table 7.7: *Movement-to-Coverage* Effectiveness Confidence Intervals.

System	Mean	Lower Boundary	Upper Boundary
DI	209.7	147.8157	271.5176
DG	600.0	538.1491	661.8509
SI	333.3	271.4824	395.1843
SG	581.7	519.8157	643.5176



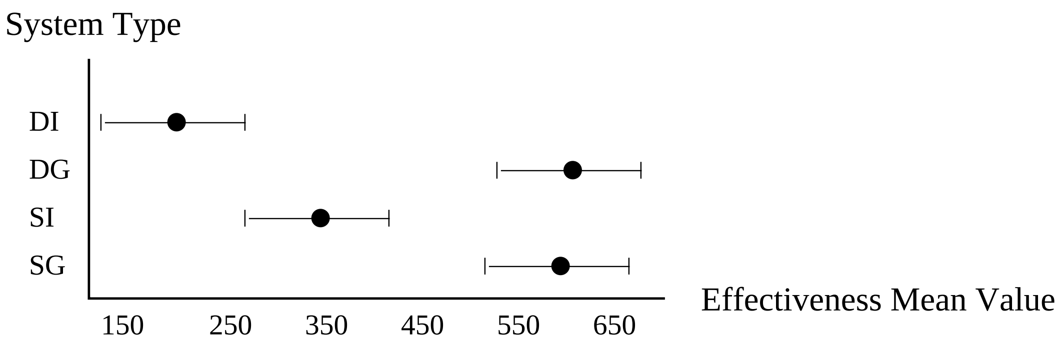
(a)

Ranking

1. Direct Manual Individual (DI)
   Supervisory Individual (SI)
   Supervisory Group (SG)
2. Direct Manual Group (DG)

(b)

Figure 7.6: *Movement-to-Coverage* Safety Confidence Intervals and Ranking. (a) depicts a graph of the confidence intervals, and (b) shows the corresponding safety ranking for this task. The three systems ranked as (1) performed better than the one ranked (2).



(a)

Ranking

1. Direct Manual Individual (DI)
   Supervisory Individual (SI)
2. Supervisory Group (SG)
   Direct Manual Group (DG)

(b)

Figure 7.7: *Movement-to-Coverage* Effectiveness Confidence Intervals and Ranking. (a) depicts a graph of the confidence intervals, and (b) shows the corresponding effectiveness ranking for this task. The two systems ranked as (1) performed better than those ranked (2).

Table 7.8: *Movement-to-Coverage* Two-factor ANOVA Table for Safety. The values of importance to us are the F^* values and the MSE value (in the Mean Square column and Error row). The row labeled "Factor B" pertains to the level of autonomy dimension, and the row labeled "Factor A" pertains to the number of robots controlled dimension.

Source	Sum of Squares (SS)	degrees of freedom (df)	Mean Square (MS)	F^*
Factor B	100	1	100	39.36
Factor A	92.04	1	92.04	36.21
Interaction	92.04	1	92.04	36.21
Error	50.83	20	2.542	
Total	335	23		

already performed the Box-Cox procedure on the data in the single-factor analysis, so we do not need to do so again. The transformed data is listed in Table 7.2. If we had not done this already, then this procedure would be needed to insure normality of the data sets and constant variance across data sets (Section 7.1.1).

A two-factor ANOVA table is now produced for each of the three criteria. Tables 7.8, 7.9, and 7.10 are the tables produced by Matlab [38] for the safety, effectiveness, and ease-of-use, respectively.

The next step is to test for interactions between the factor effects. In Section 7.2.1, the decision rule for this test was determined to be:

If $F^* \leq 4.35$,
then conclude H_0 (no interactions),
else conclude H_a (interactions present).

The F^* value used in this test is obtained from the ANOVA table (from the "F^*" column and the "Interaction" row). The test indicates that interactions are present for the safety criteria, but not for the effectiveness or ease-of-use criteria. Simple transformations do not remove the interactions from the safety data, so no general

Table 7.9: *Movement-to-Coverage* Two-factor ANOVA Table for Effectiveness. The values of importance to us are the F^* values and the MSE value (in the Mean Square column and Error row). The row labeled "Factor B" pertains to the level of autonomy dimension, and the row labeled "Factor A" pertains to the number of robots controlled dimension.

Source	Sum of Squares (SS)	degrees of freedom (df)	Mean Square (MS)	F^*
Factor B	6.118e+05	1	6.118e+05	104.5
Factor A	1.664e+04	1	1.664e+04	2.842
Interaction	3.025e+04	1	3.025e+04	4.165
Error	1.171e+05	20	5855	
Total	7.758e+05	23		

Table 7.10: *Movement-to-Coverage* Two-factor ANOVA Table for Ease-of-use. The values of importance to us are the F^* values and the MSE value (in the Mean Square column and Error row). The row labeled "Factor B" pertains to the level of autonomy dimension, and the row labeled "Factor A" pertains to the number of robots controlled dimension.

Source	Sum of Squares (SS)	degrees of freedom (df)	Mean Square (MS)	F^*
Factor B	0.002608	1	0.002608	5.25
Factor A	0.001033	1	0.001033	2.08
Interaction	0.0004187	1	0.0004187	0.8427
Error	0.009937	20	0.0004969	
Total	0.014	23		

results can be determined. The search for general results continues, however, for the other two criteria.

To determine if any general results are present for effectiveness and ease-of-use, the data is tested for factor main effects. In Section 7.2.1, the decision rule for both factor A (row) and B (column) main effects was determined to be:

If $F^* \leq 4.35$,
then conclude H_0 (no main effect),
else conclude H_a (main effect present).

The test for factor A main effects uses the F^* value in the "Factor A" row of the ANOVA table, and the test for factor B main effects uses the one in the "Factor B" row. These tests indicate that there is no effect due to the level of autonomy (factor A) for either effectiveness or ease-of-use, but there is an effect due to the number of robots controlled at a time (factor B) for both criteria.

Since more than one comparison was made for each criteria, a multiple comparison procedure must be used to determine if the main effects are significant within the 95% family confidence level (Section 7.2.2). From Equations 7.8 and 7.7, $s\{\hat{D}\} = 31.2383$ and $\hat{D} = -319.3333$ for the effectiveness factor B, and $s\{\hat{D}\} = 0.0091$ and $\hat{D} = 0.0084$ for the ease-of-use factor B. From these values and Equation 7.5, $q^* = -14.4568$ for effectiveness, and $q^* = 1.2982$ for ease-of-use. In Section 7.2.2, the decision rule for factor B main effects was determined to be:

If $\lvert q^* \rvert \leq 2.95$,
then conclude H_0 (no main effect),
else conclude H_a (main effect present).

Therefore, the main effect due to the number of robots controlled at a time is statistically significant for effectiveness, but not ease-of-use, when considered within the family of comparisons. So, there is a main effect for effectiveness, but the main effect for ease-of-use is rejected. Looking at the actual means for the effectiveness data sets, and knowing that a main effect exists concerning the number of robots controlled at a time, we can see that **Individual** control is more effective than **Group** control for this task.

Using the analysis methods presented, the rankings in Figures 7.6 and 7.7 were determined for the safety and effectiveness, and all of the systems were equivalent in terms of ease-of-use. No general findings (corresponding to main effects) were found for safety or ease-of-use, but one was found for effectiveness. This analysis method was repeated for each of the other three tasks.

7.4 Summary

Standard statistical analysis techniques are applied to determine the rankings and identify any general relationships between one system dimension and the task. A one-factor ANOVA analysis is used to determine the rankings of the individual systems and a two-factor ANOVA analysis is used for the more general findings. Three rankings (one for each of the evaluation criteria) were produced for each task class. The two-factor analysis is used to identify main effects, which correspond to the general findings.

In both the single and multi-factor analyses, the Box-Cox procedure insures that the assumptions of the ANOVA model are met. The Tukey multiple-comparison procedure insures that the 95% certainty level is maintained in both cases, even though several comparisons are made. For more details on this analysis procedure, see [41].

Chapter 8

Experimental Results

The data from the experiments was analyzed as described in Chapter 7. The results of this analysis are discussed in this Chapter. These results are divided into the three rankings that were developed for each of the tasks (Section 8.1), one for each of the three evaluation criteria, and the more general results (Section 8.2) that were found for some of the system/task combinations. A discussion of the generalizability of the results is provided in Section 8.3. The actual data values that were collected are shown in tabular form in Appendix C.

8.1 System Rankings

The four systems were ranked (in terms of safety, effectiveness, and ease-of-use) for each of the four tasks. These rankings indicate which system types are best for each type of task. A system developer, or a human operator with a system that allows choosing the type of the control, can use these rankings to determine which system to develop or use based on the criteria that are important to him.

Tables 8.1, 8.2, 8.3, and 8.4 show the rankings of the four types of systems for each task class. These were computed as described in Section 7.1, by a single-factor ANOVA analysis of the data. Among each task, there is a separate ranking for each of the three criteria of safety, effectiveness, and ease-of-use. Those systems listed first in the rankings are the best systems (out of the system types examined) for that criteria, indicated by a ranking of one.

In many cases, there is more than one system listed under the same number in the ranking, because there was no statistically significant difference between them. In a few instances, such as the ease-of-use ranking for the *movement-to-convergence* task (Table 8.2), one system is listed in more than one position in the ranking. In this case, **Supervisory Individual** control appears in both the number one and two positions, because that system's performance was not significant different than the performance of the other two systems in the number one ranking, nor was it significantly different than the other system in the number two position. The other two systems in the number one position, however, showed a statistically significantly difference than the other system in the number two position. In this case, the confidence interval for **Supervisory Individual** control overlaps those for all the other systems, but the confidence intervals for **Supervisory Group** and **Direct Manual Individual** control do not overlap the interval for **Direct Manual Group** control. Also, in a few cases (Table 8.1 Ease-of-Use, Table 8.3 Effectiveness, and Table 8.4 Safety), none of the systems distinguished themselves from the others for a particular performance criteria, and the ranking is replaced by the sentence "All systems equivalent." This was determined by the F-test (Chapter 7.1.2), which indicated that all the systems had equivalent sample means. The actual means and confidence intervals that these rankings were derived from are presented in Appendix D.

As an example, look at the Effectiveness ranking for the *movement-to-coverage* task:

1. Direct Manual Individual
   Supervisory Individual
2. Supervisory Group
   Direct Manual Group

Table 8.1: System rankings for *Movement-to-Coverage* task.

	Safety		Effectiveness	Ease of Use
1.	Direct Manual Individual	1.	Direct Manual Individual	All systems equivalent.
	Supervisory Individual		Supervisory Individual	
	Supervisory Group			
		2.	Supervisory Group	
2.	Direct Manual Group		Direct Manual Group	

Table 8.2: System rankings for *Movement-to-Convergence* task.

	Safety		Effectiveness		Ease of Use
1.	Direct Manual Individual	1.	Direct Manual Individual	1.	Supervisory Group
	Supervisory Individual		Supervisory Group		Direct Manual Individual
	Supervisory Group		Supervisory Individual		Supervisory Individual
2.	Direct Manual Group	2.	Direct Manual Group	2.	Supervisory Individual
					Direct Manual Group

This ranking indicates that **Direct Manual Individual** (DI) and **Supervisory Individual** (SI) control performed equally well. Likewise, **Supervisory Group** (SG) and **Direct Manual Group** (DG) control performed equivalently. Since the DI/SI group is ranked 1, this indicates that the two systems in it were more effective than the systems in the SG/DG group, which ranked 2.

8.2 General Findings

For some of the tasks, the data indicated more general results relating a type of system to the task class. This was determined by a two-factor ANOVA analysis

Table 8.3: System rankings for *Movement-While-Maintaining-Coverage* task.

	Safety	Effectiveness		Ease of Use
1.	Supervisory Individual	All systems equivalent.	1.	Supervisory Group
	Supervisory Group			Direct Manual Group
	Direct Manual Individual			
			2.	Direct Manual Individual
2.	Direct Manual Individual			Supervisory Individual
	Direct Manual Group			

Table 8.4: System rankings for *Movement-While-Maintaining-Convergence* task.

Safety		Effectiveness		Ease of Use
All systems equivalent.	1.	Direct Manual Group	1.	Direct Manual Group
		Supervisory Group		Supervisory Group
		Direct Manual Individual		
			2.	Direct Manual Individual
	2.	Supervisory Group		Supervisory Individual
		Direct Manual Individual		
		Supervisory Individual		

(Section 7.2). For these tasks, it was possible to determine that a certain factor level is better, based on one of the judgment criteria, than the other levels for that factor, regardless of the settings for the other factors. For instance, it might be possible to determine that, for a particular class of task, **Group** control is safer than **Individual** control, regardless of whether the system uses **Supervisory** or **Direct Manual** control. The following results are those indicated to be statistically significant by the data collected. The results are first presented as a group, and then they are discussed individually.

- *Movement to coverage* task
  - **Individual** control is more effective than **Group** control.
- *Movement to convergence* task
  - No general findings identified.
- *Movement while maintaining coverage* task
  - **Supervisory** control is safer than **Direct Manual** control.
  - **Group** control is easier to use than **Individual** control.
- *Movement while maintaining convergence* task
  - **Supervisory** control is safer than **Direct Manual** control.
  - **Group** control is more effective than **Individual** control.
  - **Direct Manual** control is more effective than **Supervisory** control.
  - **Group** control is easier to use than **Individual** control.

1. For the *movement-to-coverage* task:

Individual control is more effective than **Group** control.

Regardless of whether **Supervisory** or **Direct Manual** control is used, it is more effective to use **Individual** control when moving the robots to cover an area. This indicates that the operator will be able to complete the task significantly faster with an **Individual** control system. This is probably because *movement-to-coverage* tasks require each of the robots to move away from each other, and therefore, in different directions. **Group** control only allows the operator to give the same directional commands to the entire group. Therefore, getting the robots to move in different directions is difficult or impossible. **Individual** control, however, allows the operator to give different commands to each robot, which is what is needed for this type of task. The results seem to confirm this intuition.

2. For the *movement-to-convergence* task, no general principles were identified. This is because there were not any statistically significant main effects (Chapter 7.2) present in the experimental data. Because the two **Group** control systems only allow the user to give the same instructions to all the robots, it might seem that the same result identified for *movement-to-coverage* tasks would be found here. The waypoint commands for the **Supervisory Group** control system, however, allow the user to easily gather the robots together. This is probably the reason why no differences were found between the **Group** and the **Individual** control systems.

3. For the *movement-while-maintaining-coverage* task:

- **Supervisory** control is safer than **Direct Manual** control.
- **Group** control is easier to use than **Individual** control.

These results indicate that both **Supervisory** control systems produce less collisions than either **Direct Manual** control system. This is the expected result, since the **Supervisory** control system takes an active role in trying to avoid obstacles. It was thought that this result would be found for all the classes of tasks, although it was not. For the other tasks, the human was able to compensate for the lack of automated obstacle avoidance, although this result may not have appeared if the number of robots had been increased, producing a greater cognitive load on the operator.

The **Group** control systems also required the human to issue less commands than the **Individual** control systems while carrying out the task. *Movement-while-maintaining-coverage* tasks require that the group of robots move while maintaining positions that are spread out from each other. Since the robots are already dispersed in this task, if they each receive the same movement commands, they will tend to stay spread out, while moving in the direction indicated. Therefore, **Group** control is best suited for this task class, and requires less of the operator than **Individual** control.

4. For the *movement while maintaining convergence* task:

- **Supervisory** control is safer than **Direct Manual** control.
- **Group** control is more effective than **Individual** control.
- **Direct Manual** control is more effective than **Supervisory** control.
- **Group** control is easier to use than **Individual** control.

These results indicate that **Supervisory** control produces significantly less collisions than **Direct Manual** control, but that **Direct Manual** control results in significantly faster completion times than **Supervisory** control. It is intuitive that **Supervisory** control is safer than **Direct Manual** control. The apparent reason why **Direct Manual** control is more effective than **Supervisory** control is less apparent. This was an unexpected result, although it seems obvious in hindsight. With the **Direct Manual** control system, the operator can command the robots to take shorter paths, passing very close to obstacles, rather than skirting wide around the obstacles. With **Supervisory** control, the robots automatically go wide around the obstacles, often causing them to take longer paths, and giving the human less control of their trajectories. When attempting a *movement-while-maintaining-convergence* task, the operator or system designer will need to decide whether safety or effectiveness is more important based on the particular situation.

Group control is shown to be both more effective and easier to use than **Individual** control for this task, regardless of whether **Supervisory** or **Direct Manual** control is used. Therefore, whether the operator chooses the safer **Supervisory** control system or the more effective **Direct Manual** control system, he should use **Group** control when attempting a *movement-while-maintaining-convergence* task.

8.3 Generalizability of the Results

It is important to realize that these results are somewhat dependent on the actual setup of the experiments. In order to conduct the experiments, it was necessary to choose a particular type of supervisory control, as well as a particular method for the human to interface with this control system. Likewise, the number of robots used for the tasks had to be set at a fixed number (four in this case). It is possible, and even probable, that some of these decisions affected the results. For instance, while **Direct Manual Individual** control was found to be as safe as **Supervisory Individual** control for the *movement-to-coverage* task, when using four robots, this result may not have been the same if 100 robots were used. Likewise, a change in the human interface, even for the same control technique, can affect the results. For instance, if the human interface for the waypoint technique for giving the robots instructions was difficult to use, then the **Supervisory** control systems might not have performed as well as it might have if this interface was easy to use. Ideally, every combination of control system, number of robots, and all other possible differences should have been examined in the experiments, but there is an infinite number of possible combinations, so a complete examination is not possible.

The results will not necessarily be valid when considered outside the scope of the experiments, which is limited to similar control systems, human interfaces, robots, distribution of subject backgrounds, and experimental task setups (*e.g.*, the obstacle density in the task environment, amount of time delay present in the teleoperation, etc.). This effect of the experimental setup on the results does not mean that the results cannot be used in other situations. One should be careful when making generalizations from these results, however, especially the more general findings listed in Section 8.2. Instead, they should be used as guidelines for other situations, as well as indications of where further research should be focused. While predictions can be made, they should be verified through further experimentation.

For example, in both of the *movement-while-maintaining* tasks that were examined, **Group** control was easier to use than **Individual** control. This points out something that seems to be commonsense in afterthought. Because *movement-while-maintaining* tasks require the human operator to keep the robots from deviating from their current formation, it should be easier for the operator to give all of the robots the same movement commands at the same time, which is what group control provides. Therefore, since the two examples in these experiments point out a finding that seems logical, it is reasonable to think that the finding may apply in other situations for the same classes of tasks.

Ideally, at this point, the Multiagent Telerobotic System (MTS) designers should conduct a small set of similar experiments to test this finding with their particular MTS setup (control system, number of robots, etc.). Even if further experimentation is not conducted, however, using the results of these experiments, as well as commonsense, is better than just trying to guess what type of control system to provide for a task, which is, unfortunately, the technique that is currently used most often in practice.

8.4 Summary

Two types of experimental results were presented. First, the systems were ranked in terms of safety, effectiveness, and ease-of-use for each of the four task classes. The rankings were determined by the procedure described in Chapter 7.1. Second, some general findings were identified, relating the types of systems to the task classes, although these results were not found for all the task classes. These findings were determined by the procedure described in Chapter 7.2.

The experimental results can be used to help MTS developers create safe, effective, and easy-to-use systems for particular tasks. Likewise, if an existing system provides multiple methods of control, then a human operator can utilize these results to choose a control method for the current task. Care should be taken when generalizing the results beyond the scope of the experiments. While they can provide insights and guidelines, they should not be taken at face value in other situations.

Chapter 9

Evaluation Through Predictive Study

The rankings and principles that are the results of this study are intended to be utilized when designing a MTS for a particular task. Therefore, some sort of predictive study was needed to determine if the results are actually useful for this purpose. A predictive study is an experiment that utilizes the previous findings to predict the new results.

A different task than those used in the earlier experiments was chosen and classified using the task taxonomy. The task was chosen **prior** to the analysis of the data from the first set of experiments. The chosen task is the first subtask of a box-pushing task, that is, moving the robots such that they are all gathered close to one side of a large box, that they will later push. This fits into the *movement-to-convergence* task classification, because the robots must move from dispersed positions to gather in a group near the box.

The idea was to choose one MTS that should perform well for this task and one system that should perform poorly, based on the earlier analysis. These two systems would then be tested against each other, using the identical experimental procedure as for the initial tests. If the system that *should* perform well did better than the system that *should not*, then this indicates some validity to the earlier results, and thus the rankings would have demonstrated their utility.

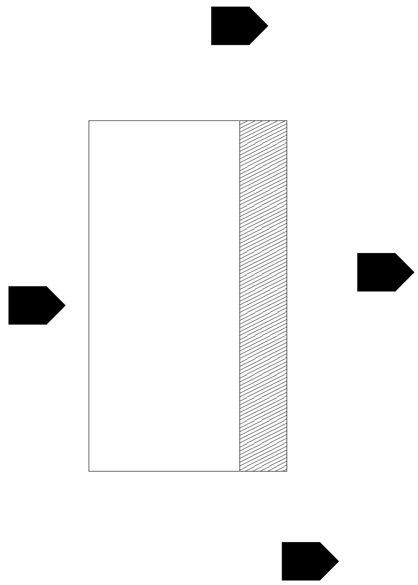
It was not possible, however, to determine which system should perform best for this class of task, since three of the systems performed equally well while the fourth performed poorly. Therefore, only one of the three good systems was chosen and compared to the lone system that should perform poorly. The initial study's results indicate that the **Direct Manual Group** (DG) control system should perform the worst for this class of task over all three of the judgment criteria (safety, effectiveness, and ease of use), and, thus, it was chosen as the predicted inferior system. While the other three types of systems ranked equally well in terms of statistical significance, **Direct Manual Individual** (DI) control was chosen as the predicted superior system for the task, because this control system performed slightly better or equal to the others when the actual mean values were consulted.

9.1 Experimental Setup

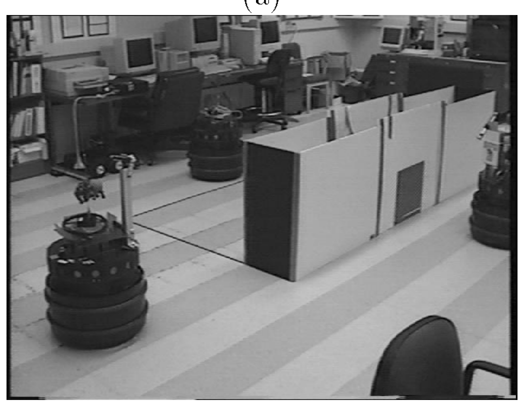
The initial task setup is shown in Figure 9.1. There was a box in the center of the task area, and the robots started at various positions around this box. The operator had to move the robots such that all four of them were within a rectangle marked on the floor on one of the long sides of the box. This rectangle was large enough that all four robots could fit within it at the same time. When all four robots were inside the rectangle, the task was completed.

9.2 Experimental Procedure

These experiments used the same experimental procedure as in the earlier tests. The only difference was that only two systems were tested, once again in experimental blocks (*i.e.*, the replication were conducted as follows: DI, DG, DI, DG, ..., DI, DG). Six replications were conducted for each system, so a total of 12 participants were used. The participants received the same training as before and were allowed to ask questions. They were given the same guidelines, *i.e.*, they should complete the task



(a)



(b)

Figure 9.1: The initial setup for the box-pushing task. In (a), the gray rectangle is the box. The other rectangle is the area that the operator had to maneuver the robots into. The black objects are robots. The photograph in (b) also shows the initial setup for the task. Three robots can be seen. The fourth is just to the right of the edge of the picture.

as fast as possible and with as few collisions as possible. The timeout period was 10 minutes.

9.3 Results

The data analysis technique was similar to that used for the earlier experiments, as presented in Chapter 7. Since only two systems were being compared here, however, the Tukey multiple-comparison procedure was not necessary and therefore omitted. The F-test described in Section 7.1.2 is sufficient to determine whether the means are equal or not, and if they are not equal, it is obvious from the actual means which is greater. Additionally, the Box-Cox procedure indicated that no transformation was necessary to insure normality and equal variance for any of the data sets.

The **Direct Manual Individual** control system proved better than the **Direct Manual Group** control system for all three of the criteria: safety, effectiveness, and ease-of-use. The difference was even greater than in the earlier experiments. As this was the anticipated result, based on the previous analysis, it lends validity to the earlier experiments.

The means for each of the systems are presented in Table 9.1. The actual data values are shown in tabular form in Appendix C. The F^* value used to determine if the means are significantly different is listed in Appendix D.

The most significant difference between the systems was found in terms of Effectiveness. *None* of the participants who used the **Direct Manual Group** control system was able to complete the task. The task is possible with this system, as the designer tried it himself and was able to complete it. This, however, was performed post facto: the experimenter had watched the other participants try and had gained ideas on how to accomplish it from them. In order to accomplish the task with the **Direct Manual Group** control system, it is necessary to intentionally drive the

Table 9.1: Means of the Predictive Study Data Sets

Safety Means

System	Mean
Direct Manual Individual	0.3
Direct Manual Group	4.3

Effectiveness Means

System	Mean
Direct Manual Individual	232.8
Direct Manual Group	600.0

Ease-of-use Means

System	Mean
Direct Manual Individual	66.5
Direct Manual Group	88.0

robots into non-movable obstacles in order to move the robots closer to each other. Therefore, one must sacrifice safety in order to accomplish the task, or they must sacrifice effectiveness to avoid collisions.

9.4 Summary

The predictive study helped to validate the earlier results, and demonstrate their utility for designing MTSs (of the types considered in this dissertation) for the examined tasks. This study examined an experimental task (part of box-pushing) that had not been used in the earlier experiments, but fit the *movement-to-convergence* class. **Direct Manual Individual** and **Direct Manual Group** control, which the earlier analysis indicated were superior and inferior, respectively, for this task, were compared. The results were as anticipated, with the **Direct Manual Individual** MTS performing better than the **Direct Manual Group** system.

Chapter 10

Contributions

There are three main contributions from this work:

- the rankings and general findings that are the results of the experiments conducted
- an adaptation of a general experimental methodology to make it appropriate for large-scale telerobot evaluations
- a taxonomy of mobile multiagent tasks

Each of these contributions is discussed in the following sections.

10.1 Experimental Results

Large-scale evaluations of different types of systems are necessary to advance the fields of robotics and telerobotics to a more scientific stage. Currently, most researchers use intuition to develop robotic systems. Those evaluations which are conducted usually either compare the researcher's system to one other system for a single task, or compare the researcher's system to itself for multiple tasks. These are useful comparisons, but unbiased evaluations of multiple systems for multiple tasks are important for developing scientific principles which can be used in other research.

This dissertation's research is the first such evaluation at such a large scale. Ninety-six subjects were used to evaluate the Multiagent Telerobotic Systems (MTSs) (Chapter 6), and an additional twelve were used to test the predictive qualities of the results (Chapter 9). This number of human participants is believed to be unprecedented for system evaluations in the field of mobile telerobotics. Many more such large-scale evaluations are needed, as this one covers only a small subset of the possible MTSs and the task types. This study begins the task, however, and hopefully, it will inspire other researchers to conduct similar evaluations.

Two types of results were generated by these experiments: system rankings for each task and, in some cases, general findings relating one of the examined dimensions to the task type. These results are presented in Chapter 8. The rankings show which system types are best for a task in terms of each of the three criteria (safety, effectiveness, and ease-of-use). The general findings indicate that one system dimension has a particular effect for a task type regardless of the setting of the other dimension. An example general finding that was found is "**Group** control is more effective than **Individual** control for *movement-while-maintaining-convergence* tasks regardless of whether **Supervisory** or **Direct Manual** control is used."

The rankings and general findings that resulted from these experiments are useful to both MTS developers and users. Developers can use them to help create safe, effective, and easy-to-use MTSs. If they know the nature of the task or tasks that their system will be used for, they can see what sort of systems were found to be best for each criteria in this evaluation, and then develop a system that will be best for them, based on their own needs. If an MTS user can choose between multiple control methods at execution time, then he can use the results of these evaluations to decide on the control technique that is best for his current task.

A predictive study (Chapter 9) demonstrated the utility of the experimental results for developing MTSs. The system rankings were used to determine which system types *should* and *should not* perform well for part of a box-pushing task. These two systems (**Direct Manual Individual** and **Direct Manual Group** control) were then compared for this task, and they performed as had been predicted. This shows that the experimental results presented in Chapter 8 can be used effectively to develop good MTSs.

Additionally, the results can be used as a starting point to indicate what sort of additional evaluations are needed. The evaluations cover only a small section of the space comprised of all possible MTSs. Similar evaluations should be conducted to broaden our knowledge of the relationships between MTS systems and tasks.

10.2 Experimental Methodology for Evaluating Telerobot Systems

Most robotics research does not consider system evaluation in any structured manner. However, this is an essential part of any scientific or engineering endeavor. Without formal analysis, and the discovery of the underlying design principles, robotics will remain more of an art than a science.

Since no comparisons of this sort have been conducted in the past, no experimental methodology existed that was tailored to this type of study. A general experimental methodology has been adapted to fit these kinds of evaluations, and has been applied to a comparison of mobile multiagent telerobotic systems. The methodology combines elements from the areas of user-interface evaluation, human-factors studies, and general experimental design.

This methodology is presented in Chapter 3 in a generalized manner, to provide an easy-to-use procedure for other researchers conducting their own evaluations. The basic procedure is as follows:

1. Determine evaluation criteria.
2. Decide what types of systems to compare.
3. Formulate tasks to use the systems for.
4. Conduct experiments with real robots to compare the systems for the tasks.
5. Analyze the data collected from the experiments to determine how each system performed.

The methodology provides suggestions for formally determining which system and task types to examine. It also provides guidelines for conducting experiments involving both human operators and real robots.

Furthermore, the methodology is not specific to either multiagent or mobile robot systems. It can also be used for single-agent systems and manipulator arms. Many of the techniques used, such as the methods for selecting systems and tasks for the evaluation, are also applicable to fully autonomous robot evaluations.

The methodology, as presented, is intended to serve as a standard procedure for other telerobotics researchers to evaluate the utility of various system designs, as well as to compare existing applications. Hopefully, the presentation of this work and the explanation (Chapter 3) of how to apply the methodology to other robot system evaluations will provide the impetus for other researchers to conduct these sorts of evaluations and to realize their importance in robotics research.

10.3 Mobile Multiagent Task Taxonomy

When two or more researchers compare their robotic systems for a particular task or tasks, it is important to know whether the tasks are similar in nature or of different types. A formal taxonomy of tasks allows researchers to know if they are evaluating their systems for the same kind of tasks as other researchers. For instance, two tasks may seem different in their application details, yet the underlying movement types (or other characteristics) may be similar. An example is pushing a box and flying in flocks, in which the high-level descriptions seem different, yet both applications involve moving as a coordinated group. Taxonomies allow us to classify application tasks into general categories, thereby helping to identify similarities and differences between tasks.

Additionally, when comparing mobile multiagent systems for tasks, it is necessary to know what types of tasks exist. Most evaluations will only be able to examine a subset of all the possible task classifications. A formal taxonomy of tasks will allow a structured choice of which task types to examine. For instance, one set of experiments could examine one branch in the taxonomy tree, including all of that branch's sub-categories. Then another set of experiments could examine a different branch. This ensures both that the tasks chosen for each evaluation are similar in nature, and that no general task type is forgotten.

No existing taxonomy for classifying mobile multiagent tasks could be found. A taxonomy (Figure 10.1) has been developed that classifies tasks or subtasks in terms of the relative motion of the robots (Chapter 5.1). It provides a means for comparing the mobility requirements of various multiagent tasks. This taxonomy differentiates tasks based on whether they are *coverage* or *convergence* tasks, *movement-to* or *movement-while-maintaining* tasks, and *known-positions* or *unknown-positions* tasks. These classifications are described in more detail in Chapter 5.1.

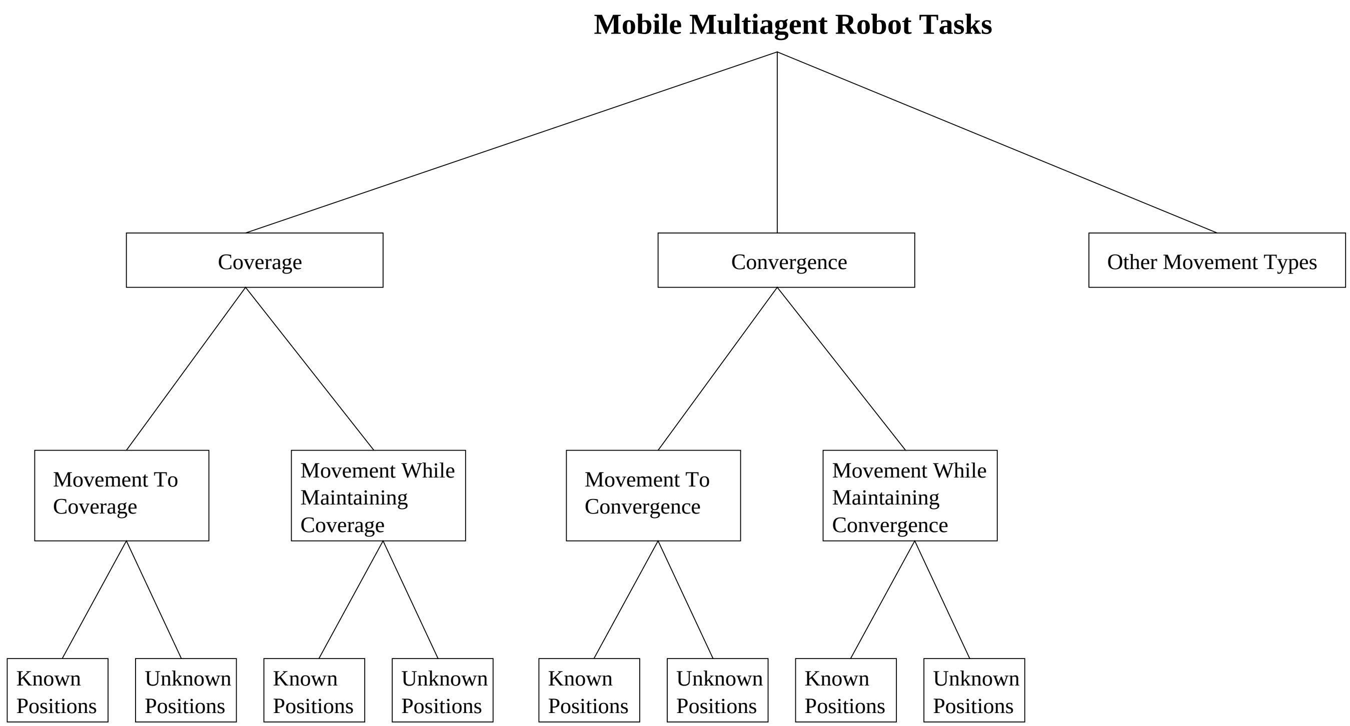


Figure 10.1: Mobile Multiagent Task Taxonomy.

The presented taxonomy is appropriate for classifying the relative motion of multiagent human and animal tasks as well as robot or telerobot tasks. Furthermore, it allows a MTS designer to classify his task so that he can use the results of this research. The taxonomy can be used to guide further system evaluations. Only part of this taxonomy was examined as part of this research; the other task categories remain to be studied.

10.4 Future Work

This research raises as many questions as it answers. For instance, would the results be the same if the number of robots used was varied? What effect do other possible system dimensions have on the performance of MTSs? What sort of results would be found if other points along the same two dimensions were examined?

Many more similar experiments need to be conducted. These are only the beginning. There is an infinite space of possible MTSs and tasks, and these experiments cover only a small section of that space. Only two dimensions of possible telerobot systems were examined. Within each of these dimensions, only two points were examined. Only a portion of the task taxonomy was considered. Furthermore, there are potentially infinite different taxonomies of mobile multiagent tasks, which may classify the applications in other ways. Each MTS was tested with exactly four robots, rather than examining various numbers.

The goal of telerobot system evaluators should be to examine as much of this space as possible. The next logical step would be to examine the remainder of this taxonomy with the same systems and tasks. This dissertation has begun these evaluations. It provides a general description of a formal telerobotic evaluation methodology for researchers to use. It is hoped that others will see the importance of formal evaluations and use this methodology to conduct further experiments.

Evaluations of this sort will help transform robotics and telerobotics into a science, rather than only an engineering field.

Appendix A

Terminology

behavior-based: Behavior-based robots determine their actions through some combination of the output of one or more simple behaviors. Each behavior takes care of one aspect of the task, such as obstacle avoidance.

block: A grouping of experimental treatments, with every treatment occurring once. In temporal blocking, which is used in this thesis, one replication is conducted on all treatments in the block before the next replication is begun on any treatment.

direct manual control: This means that the human operator has complete control over the robots' actions. Aside from its physical limitations, the robot does not contribute to determining its motion. An example is a radio controlled toy car.

factor: A factor is a predictor, or independent, variable to be studied in an investigation. For instance, in an investigation of the effect of price on product sales, price is the examined factor.

factor level: A factor level is a particular value for that factor. If price is the factor, then $50 would be one of many possible factor levels.

mobile telerobot: This is a telerobot that is capable of moving itself. For instance, a telerobotic car would be considered a mobile telerobot, while a telerobot arm that is bolted to a table would not.

multiagent robotics: This refers to using more than one robot to complete the same task. The robots may be working on the same or different subtasks, and they are not required to use the same approach to the task.

multiagent telerobotic system (MTS): This is a group of more than one telerobot controlled by a human operator.

replication: Replications are repeat trials for the same experimental treatment.

shared control: With shared control, the instructions given by the human and the robot are combined in some manner to determine the motion of the robot.

strict supervisory control: With strict supervisory control, the human operator instructs the robot and then observes it as it attempts to autonomously carry out its instructions. If there is a problem, the human may help out by giving more instructions. The term *supervisory control*, in the less strict sense, is often used to refer to either shared control, strict supervisory control, or combinations of the two approaches.

teleoperator: A teleoperator is a machine that uses *direct manual control*.

telerobot: The strict definition of a telerobot is a robot that determines its actions based on some combination of human input and autonomous control. A telerobot can use *shared control* or *strict supervisory control*. In this research, the term telerobot will be used to refer to both true telerobots and to *teleoperators*.

treatment: A treatment is a combination of levels from each factor in an experiment. So, if one factor being examined is price (with levels $50 and $100) and another is size (with levels small and large), then each of the four combinations of price and size (\$50/small, \$50/large, \$100/small, and \$100/large) is a treatment. In a one-factor experiment, the factor levels are the treatments.

Appendix B

Details of the Control Systems

The first section of this appendix describes how each of the motor schemas used in this research functions. The second section tells which motor schemas were used with each of the four systems and the parameter settings used.

B.1 Motor Schema Details

Each motor schema produces a vector output consisting of a direction and magnitude. The vector outputs of the active schemas are then summed and normalized to produce a final motion vector for the robot. The motor schemas used in this research are computed as follows:

Avoid-obstacle

Moves the robot away from an obstacle [5].

$V_{magnitude} =$

0 for $d > S$

$\frac{S-d}{S-R} * G$ for $R < d \leq S$

∞ for $d \leq R$

$V_{direction}$ = along a line from the robot to the center of the obstacle, pointing away from the obstacle.

S, R, G, and d are defined as follows:

S = sphere of influence (radial extent of force from the center of the obstacle),

R = radius of the obstacle,

G = gain,

d = distance of the robot to the center of obstacle.

Avoid-robot

Moves the robot away from another robot. The details of this schema are the same as for the **avoid-obstacle**, except that another robot is substituted for the obstacle.

Formation

Moves the robot to its proper place in the group formation. The **Formation** schema used in this research uses the unit-center reference [13]. A unit-center is computed by averaging the x and y positions of all the robots involved in the formation. Each robot determines its own formation position relative to that center.

$V_{magnitude}$ =

0 if $d \leq D$

$\frac{d-D}{B} * G$ if $D < d < B$

G if $d \geq B$

$V_{direction}$ = toward the formation position for this robot based on the type of formation.

d, D, B, and G are defined as follows:

d = distance from the robot to its formation position,

D = dead zone radius,

B = ballistic zone radius,

G = gain.

Move-to-waypoint

Moves the robot towards the location of the next waypoint.

$V_{magnitude}$ = fixed gain value.

$V_{direction}$ = in the direction of the next waypoint.

Teleautonomy

Moves the robot in the direction and speed that the human operator inputs into the joystick.

$V_{magnitude}$ = set by user (between 0 and a fixed gain)

$V_{direction}$ = set by user.

B.2 Parameter Settings for Each System

Parameters common to all systems

Parameter name	Setting
Base velocity	0.15 m/s
Max velocity	0.20 m/s
Cautious velocity	0.01 m/s

Direct Manual Control Assemblage

The following schema and parameter were used for both the **Direct Manual Individual** control and the **Direct Manual Group** control.

Teleautonomy Schema

Parameter name	Setting
Gain	1.0

Supervisory Control Assemblage

The following schemas and parameters were used for both the **Supervisory Individual** control and the **Supervisory Group** control.

Avoid-obstacle Schema

Parameter name	Setting
Gain	1.0
Sphere of Influence	0.75 m
Safety Margin	0.1 m

Avoid-robot Schema

Parameter name	Setting
Gain	1.2
Sphere of Influence	0.6 m
Safety Margin	0.4 m

Formation Schema

Parameter name	Setting
Gain	1.0
Type	*Set by user*
Reference	Unit-center-referenced
Spacing	0.5 m
Dead Zone Radius	0.25 m
Saturation Length	1.0 m

Teleautonomy Schema

Parameter name	Setting
Gain	1.0

Move-to-waypoint Schema

Parameter name	Setting
Gain	1.0
Success Radius	0.75 m

Appendix C

Data Values

The actual data that was collected during the experiments is presented in tabular form here. The data is shown on five tables, one for each of the four task classes in the main experiments (described in Chapter 6) and one for the predictive experiment (described in Chapter 9).

Each table is divided into four sections, one for each of the types of data collected. In each section, the four systems are listed. The six data values (from the six replications for that system/task combination) for that system are shown across the row.

Table C.1: *Movement-to-Coverage* Data Values

	Subject Number					
	1	2	3	4	5	6
Number of collisions						
Direct Manual, Individual	0	0	0	0	0	0
Direct Manual, Group	12	7	6	5	12	6
Supervisory, Individual	0	0	0	0	0	0
Supervisory, Group	0	0	1	0	0	0
Completion of the task (Y = Completed, N = Incomplete)						
Direct Manual, Individual	Y	Y	Y	Y	Y	Y
Direct Manual, Group	N	N	N	N	N	N
Supervisory, Individual	Y	Y	Y	Y	Y	Y
Supervisory, Group	Y	N	N	Y	Y	N
Task completion time (seconds)						
Direct Manual, Individual	212	181	237	253	196	179
Direct Manual, Group	600	600	600	600	600	600
Supervisory, Individual	198	308	500	534	221	239
Supervisory, Group	552	600	600	599	539	600
Number of user actions						
Direct Manual, Individual	47	26	32	51	42	36
Direct Manual, Group	45	40	108	49	55	77
Supervisory, Individual	32	58	80	82	56	30
Supervisory, Group	45	65	67	53	80	73

Table C.2: *Movement-to-Convergence* Data Values

	Subject Number					
	1	2	3	4	5	6
Number of collisions						
Direct Manual, Individual	0	0	0	0	0	0
Direct Manual, Group	4	1	1	0	5	0
Supervisory, Individual	0	0	0	0	0	0
Supervisory, Group	0	0	0	0	0	0
Completion of the task (Y = Completed, N = Incomplete)						
Direct Manual, Individual	Y	Y	Y	Y	Y	Y
Direct Manual, Group	N	N	N	N	N	N
Supervisory, Individual	Y	Y	Y	Y	Y	Y
Supervisory, Group	Y	Y	Y	Y	Y	Y
Task completion time (seconds)						
Direct Manual, Individual	174	245	173	195	169	123
Direct Manual, Group	600	600	600	600	600	600
Supervisory, Individual	348	201	342	234	151	241
Supervisory, Group	314	256	131	313	211	213
Number of user actions						
Direct Manual, Individual	28	56	27	51	49	38
Direct Manual, Group	116	52	61	57	103	66
Supervisory, Individual	70	34	56	43	37	31
Supervisory, Group	46	14	15	16	36	47

Table C.3: *Movement-While-Maintaining-Coverage* Data Values

	Subject Number					
	1	2	3	4	5	6
Number of collisions						
Direct Manual, Individual	0	1	0	0	0	0
Direct Manual, Group	0	0	3	2	2	0
Supervisory, Individual	0	0	0	0	0	0
Supervisory, Group	0	0	0	0	0	0
Completion of the task (Y = Completed, N = Incomplete)						
Direct Manual, Individual	Y	Y	Y	Y	Y	Y
Direct Manual, Group	N	Y	Y	Y	Y	Y
Supervisory, Individual	Y	Y	Y	N	Y	Y
Supervisory, Group	N	Y	Y	N	N	Y
Task completion time (seconds)						
Direct Manual, Individual	186	166	186	217	177	238
Direct Manual, Group	264	156	333	244	173	158
Supervisory, Individual	210	163	251	260	269	134
Supervisory, Group	197	148	146	216	318	206
Number of user actions						
Direct Manual, Individual	50	41	36	60	57	47
Direct Manual, Group	36	24	38	24	27	20
Supervisory, Individual	38	35	83	56	54	30
Supervisory, Group	8	25	19	17	35	21

Table C.4: *Movement-While-Maintaining-Convergence* Data Values

	Subject Number					
	1	2	3	4	5	6
Number of collisions						
Direct Manual, Individual	0	0	1	0	1	2
Direct Manual, Group	0	1	1	0	2	1
Supervisory, Individual	0	1	0	0	0	0
Supervisory, Group	0	0	0	0	1	0
Completion of the task (Y = Completed, N = Incomplete)						
Direct Manual, Individual	Y	Y	Y	Y	Y	Y
Direct Manual, Group	Y	Y	Y	Y	Y	Y
Supervisory, Individual	Y	Y	Y	Y	Y	Y
Supervisory, Group	Y	Y	Y	Y	Y	Y
Task completion time (seconds)						
Direct Manual, Individual	458	694	962	476	502	384
Direct Manual, Group	355	369	386	496	360	312
Supervisory, Individual	767	642	917	978	859	592
Supervisory, Group	1539	549	299	550	583	292
Number of user actions						
Direct Manual, Individual	178	186	263	105	128	99
Direct Manual, Group	42	23	36	54	34	29
Supervisory, Individual	175	132	172	191	221	162
Supervisory, Group	194	58	29	26	54	27

Table C.5: Predictive Task Data Values

	Subject Number					
	1	2	3	4	5	6
Number of collisions						
Direct Manual, Individual	1	0	1	0	0	0
Direct Manual, Group	1	3	0	7	4	11
Completion of the task (Y = Completed, N = Incomplete)						
Direct Manual, Individual	Y	Y	Y	Y	Y	Y
Direct Manual, Group	N	N	N	N	N	N
Task completion time (seconds)						
Direct Manual, Individual	207	227	248	272	212	231
Direct Manual, Group	600	600	600	600	600	600
Number of user actions						
Direct Manual, Individual	79	49	63	84	42	82
Direct Manual, Group	116	127	95	103	35	52

Appendix D

Confidence intervals, λ, F^*, and q^* values

The following is a list of the λ values used to transform the experimental data (using Y^λ transform), the F^*-values obtained from the single-factor and two-factor ANOVA analysis of the experimental data values, and the family confidence intervals used to determine the rankings. If the two-factor F test indicated that a main effect was present, then the q^* value used by the Tukey multiple comparison procedure to test for main effects is also listed. For more information about the techniques used to determine the λ value, F-values, and confidence intervals, as well as what each of these reveals, see Chapter 7. All of the following tests were performed using an α value of 0.05.

As discussed in Section 7.1.2, the appropriate decision rule for the single-factor ANOVA tests is:

If $F^* \leq 3.10$,
then conclude H_0,
else conclude H_a.

If H_0 was concluded, then no confidence intervals are presented, since they were not needed to determine a ranking. Also, as presented in Section 7.2.1, the decision rule for determining interactions and the possible presence of main effects is:

If $F^* \leq 4.35$,
then conclude H_0,
else conclude H_a

This test uses the three F^* values for interactions, columns, and rows to determine if there are interactions, main effects due to the number of robots controlled, and main effects due to the level of autonomy respectively.

The decision rule for the multiple comparison test for main effects is as follows:

If $\lvert q^* \rvert \leq 2.95$,
then conclude H_0,
else conclude H_a.

This is described in more detail in Section 7.2.2.

In the following tables, DI stands for **Direct Manual Individual** control, DG is **Direct Manual Group**, SI is **Supervisory Individual**, and SG is **Supervisory Group**. The confidence intervals and means presented are for the transformed data values. The data was transformed as described in Section 7.1.1.

D.1 *Movement-to-coverage* task

	Safety	Effectiveness	Ease-of-use
Transformation λ	1	1	-0.1
Single-factor F^*	37.26	37.5	2.724
Two-factor Interaction F^*	36.21	5.165	0.8427
Two-factor Column F^*	39.36	104.5	5.25
Two-factor Column q^*	NA	-14.4568	1.2982
Two-factor Row F^*	36.21	2.842	2.08
Two-factor Row q^*	NA	NA	NA

Safety Confidence Intervals

System	Mean	Lower Boundary	Upper Boundary
DI	0.0	-1.2887	1.2887
DG	8.0	6.7113	9.2887
SI	0.0	-1.2887	1.2887
SG	0.2	-1.1220	1.4554

Effectiveness Confidence Intervals

System	Mean	Lower Boundary	Upper Boundary
DI	209.7	147.8157	271.5176
DG	600.0	538.1491	661.8509
SI	333.3	271.4824	395.1843
SG	581.7	519.8157	643.5176

No Ease-of-use confidence intervals needed (all systems equivalent).

D.2 *Movement-to-convergence* task

	Safety	Effectiveness	Ease-of-use
Transformation λ	1	1	0.3
Single-factor F^*	4.416	69.23	7.152
Two-factor Interaction F^*	4.416	89.45	11.36
Two-factor Column F^*	4.416	78.91	1.47
Two-factor Column q^*	NA	NA	NA
Two-factor Row F^*	4.416	39.33	8.3
Two-factor Row q^*	NA	NA	NA

Safety Confidence Intervals

System	Mean	Lower Boundary	Upper Boundary
DI	0.0	-0.8638	0.8638
DG	1.8	0.9695	2.6971
SI	0.0	-0.8638	0.8638
SG	0.0	-0.8638	0.8638

Effectiveness Confidence Intervals

System	Mean	Lower Boundary	Upper Boundary
DI	179.8	134.4737	225.1929
DG	600.0	554.6404	645.3596
SI	252.8	207.4737	298.1929
SG	239.7	194.3071	285.0263

Ease-of-use Confidence Intervals

System	Mean	Lower Boundary	Upper Boundary
DI	3.0	2.7404	3.3249
DG	3.6	3.3346	3.9191
SI	3.1	2.8160	3.4005
SG	2.7	2.3771	2.9616

D.3 *Movement-while-maintaining-coverage* task

	Safety	Effectiveness	Ease-of-use
Transformation λ	0.5	-0.7	0.1
Single-factor F^*	3.579	0.4533	9.918
Two-factor Interaction F^*	2.419	0.001137	1.066
Two-factor Column F^*	2.419	0.756	27.43
Two-factor Column q^*	NA	NA	7.4062
Two-factor Row F^*	5.9	0.6027	1.26
Two-factor Row q^*	3.4349	NA	NA

Safety Confidence Intervals

System	Mean	Lower Boundary	Upper Boundary
DI	0.2	-0.2111	0.5444
DG	0.8	0.3823	1.1378
SI	0.0	-0.3778	0.3788
SG	0.0	-0.3778	0.3788

No Effectiveness confidence intervals needed (all systems equivalent).

Ease-of-use Confidence Intervals

System	Mean	Lower Boundary	Upper Boundary
DI	4.7	4.2686	5.1444
DG	3.8	3.3387	4.2145
SI	4.7	4.2487	5.1246
SG	3.3	2.8621	3.7379

D.4 *Movement while maintaining convergence* task

	Safety	Effectiveness	Ease-of-use
Transformation λ	0.5	-1	0.4
Single-factor F^*	1.825	4.592	18.04
Two-factor Interaction F^*	0.1522	0.00173	0.2027
Two-factor Column F^*	0.1522	8.689	52.4
Two-factor Column q^*	3.2155	-4.1687	10.2369
Two-factor Row F^*	5.169	5.086	1.521
Two-factor Row q^*	NA	3.1893	NA

No Safety confidence intervals needed (all systems equivalent).

Effectiveness Confidence Intervals

System	Mean	Lower Boundary	Upper Boundary
DI	0.012	0.0099	0.0147
DG	0.016	0.0134	0.0182
SI	0.010	0.0071	0.0119
SG	0.013	0.0108	0.0155

Ease-of-use Confidence Intervals

System	Mean	Lower Boundary	Upper Boundary
DI	7.5	7.0630	7.9388
DG	4.2	3.7342	4.6100
SI	7.9	7.4408	8.3166
SG	4.9	4.4778	5.3536

D.5 Predictive study task

	Safety	Effectiveness	Ease-of-use
Transformation λ	1	1	1
Single-factor F^*	5.669	2.393	23.5

Appendix E

Subject Distribution

The two tables in this section show the distribution of participants across the different systems and tasks. The first table shows the distribution across the types of systems. Each column of this table represents a different system type. The second table shows the distribution across the different tasks. Each column of the second table represents one of the four task classes. Each of the tables is divided into four sections of rows. The first set of rows shows the distribution of males and females. The second set shows the age groups. The third set shows the highest level of education that the participant had already achieved. The last set indicates whether the participant had previous experience working with mobile robots.

Each cell in the tables has two numbers. The first number is the actual number of participants that fit this profile. The second number, in parentheses, is the percentage of the participants *for that system or task* that fits this profile.

The subjects were mostly males between the ages of 20 and 29. The ratios of males to females and between the different age groups is reasonably consistent across the different systems and tasks. High school was the highest level of education completed for most subjects. Although the level of education was constant across most of the tasks and systems, the *movement-while-maintaining-convergence* task had greater numbers of subjects with higher education levels than the other tasks. Almost all of the participants had no prior experience using mobile robots. Those subjects who had used mobile robots before were evenly distributed across the system types and somewhat evenly across the task classes.

Table E.1: Participant distribution by system. The first set of rows show the participant's sex, followed by the age, highest level of education completed, and then whether or not they have experience working with mobile robots.

	Direct Manual Individual	**Direct Manual Group**	**Supervisory Individual**	**Supervisory Group**
male	18 (75%)	16 (67%)	20 (83%)	17 (71%)
female	6 (25%)	8 (33%)	4 (17%)	7 (29%)
10 - 19	5 (21%)	6 (25%)	5 (21%)	7 (29%)
20 - 29	16 (67%)	15 (63%)	14 (58%)	14 (58%)
30 - 39	0 (0%)	1 (4%)	5 (21%)	2 (8%)
40 - 49	1 (4%)	0 (0%)	0 (0%)	0 (0%)
50 - 59	2 (8%)	2 (8%)	0 (0%)	1 (4%)
pre-high school	1 (4%)	0 (0%)	0 (0%)	0 (0%)
high school	14 (58%)	15 (63%)	10 (42%)	17 (71%)
Bachelor's degree	5 (21%)	4 (17%)	8 (33%)	2 (8%)
Master's degree	2 (8%)	2 (8%)	4 (17%)	4 (17%)
Ph.D./M.D.	2 (8%)	2 (8%)	1 (4%)	1 (4%)
other	0 (0%)	1 (4%)	1 (4%)	0 (0%)
experience	2 (8%)	3 (12%)	2 (8%)	2 (8%)
no experience	22 (92%)	21 (88%)	22 (92%)	22 (92%)

Table E.2: Participant distribution by task. The first set of rows show the participant's sex, followed by the age, highest level of education completed, and then whether or not they have experience working with mobile robots.

	Movement to coverage	*Movement to convergence*	*Movement while maintaining coverage*	*Movement while maintaining convergence*
male	16 (67%)	19 (79%)	15 (63%)	21 (87%)
female	8 (33%)	5 (21%)	9 (37%)	3 (13%)
10 - 19	3 (13%)	6 (25%)	9 (38%)	5 (21%)
20 - 29	16 (67%)	17 (71%)	11 (46%)	15 (63%)
30 - 39	2 (8%)	0 (0%)	3 (13%)	3 (13%)
40 - 49	0 (0%)	1 (4%)	0 (0%)	0 (0%)
50 - 59	3 (13%)	0 (0%)	1 (4%)	1 (4%)
pre-high school	0 (0%)	0 (0%)	0 (0%)	1 (4%)
high school	15 (63%)	17 (71%)	16 (67%)	8 (33%)
Bachelor's degree	3 (13%)	5 (21%)	5 (21%)	6 (25%)
Master's degree	3 (13%)	1 (4%)	1 (4%)	7 (29%)
Ph.D./M.D.	3 (13%)	1 (4%)	2 (8%)	0 (0%)
other	0 (0%)	0 (0%)	0 (0%)	2 (8%)
experience	3 (12%)	1 (4%)	1 (4%)	4 (17%)
no experience	21 (88%)	23 (96%)	23 (96%)	20 (83%)

Appendix F

Forms Used During the Experiments

Figures F.1 and F.2 are the consent form that the human subjects signed before participating in the experiments. The Georgia Institute of Technology Institutional Review Board approved this form and the experiments. Figure F.3 is the survey that the subjects completed. After that, the script that the experimenter used when talking to the subjects is presented.

Figure F.1: Experimental Consent form, Page 1.

Figure F.2: Experimental Consent form, Page 2.

Figure F.3: The survey that the subjects completed.

The script the experimenter used when talking to the subjects:

First of all, let me tell you what I am doing. A multiagent telerobotic system is a system that allows a human to control a group of robots. There are many different kinds of systems. What I want to do is to determine what kind of these systems are best for what types of tasks. To do this, I have identified four types of systems for controlling groups of robots, and I have identified the different types of tasks that groups of robots might be asked to perform. Now, I am bringing in a whole bunch of people, and each person is using one of the systems for one of the types of tasks. I want you to understand that I am testing the system and not you. Furthermore, since this is all based on the assumption that certain types of systems are better than others for certain tasks, the system you get to use may be easy to complete the task with, or it may be difficult or even impossible to complete the task.

I'd like you to fill out a consent form now. There are two of them, but they are the same. The reason there are two is that you get to keep one and I keep one.

I am going to be videotaping the robots. You can see the videocamera over there. It can see the robots. It can't see you. Furthermore, your identity will not be associated with the data I collect in any way. It will only be referred to as the data from participant number _____.

Now I'd like you to fill out a survey.

Now I am going to show you how to use the controls for the system that you are going to use. You will learn how to use the controls using simulated robots, here on the computer screen. When you do the actual task that I will be taking data measurement on, you will use the real robots. However, the controls will be the same.

———————————————————————————-

(One of the following)

Sentry Positioning

In this task, you will pretend that the robots are sentries, whose job is to guard this lab at night. If you look at the floor, you can see that it is marked with electrical tape. There is black tape and red tape. Ignore the red tape, as that is another student's work. The black tape divides the lab into four quadrants. A robot can guard an entire quadrant if it is completely inside that quadrant. Since there are four robots and four quadrants, the robots can effectively guard the lab if there is one robot in each quadrant. It is the beginning of the night, and the robots are gathered in one section of the lab, doing whatever robots do when they are not working. Your job is to move the robots so that there is one robot in each quadrant.

Gathering to Perform Work

If you look at the floor, you can see that it is marked with electrical tape. There is black tape and red tape. Ignore the red tape, as that is another student's work. The black tape divides the lab into four quadrants. There is a robot in each quadrant, performing some work in that quadrant. When the task begins, this robot contacts you and informs you that there is a broken machine in its quadrant, and that it needs the help of the other three robots to fix it. Your job is to move the robots such that all four of them are within the quadrant with the broken machine at the same time.

Patrolling

In this task, you will pretend that the robots are soldiers, and they have been sent on a patrol of this lab. If you look at the floor, you can see that it is marked with electrical tape. There is black tape and red tape. Ignore the red tape, as that is another student's work. The black tape divides the lab into four quadrants. The robots' patrol route takes them through the four quadrants in this order (show order) and then back to the starting quadrant. Your job is to move the robots through the quadrants in this order. Since the robots represent soldiers, they must stay together as you move them to protect each other. To ensure this, as you are moving the robots from this quadrant to this one, none of the robots can move into the next quadrant until they are all within this one. And as you are moving them into each of the other quadrants, none of them can move into the next quadrant until they are all within the current one. When you get them back into the start quadrant, you will have completed the task.

Dragging a River Bottom

In this task, you will pretend that the lab is a river, and that the robots are boats dragging the river bottom for something. If you look at the floor, you can see that it is marked with electrical tape. There is black tape and red tape. Ignore the red tape, as that is another student's work. The black tape divides the lab into four lanes, and there is a robot in each lane. The lanes go across the lab, or downriver. Your job is to move the robots downriver and across this finish line. To make sure that they don't miss any area, you must keep each robot in its respective lane as you move them. In other words, this robot can not cross over into that robot's lane, and it can't go out of the outside boundary either.

———————————————————————————-

When you have accomplished this, you will have completed the task. I will be watching the robots, and I will tell you when you have completed the task. When I say that you have completed the task, I want you to take your hand off the mouse, even if the robots are still moving, and I will come shut down the system. You will have _____ minutes to complete the task. If, by any chance, the timeout period is exceeded, I will come shut down the system.

These trash cans and this box are obstacles. You want to try to avoid hitting them. However, if you do hit them, don't worry. The robots are moving slow enough that it won't hurt them. The tables on the side of the room are also obstacles. If a robot runs underneath one of them, it could knock something off the top of the robot. So, if I see that the robot is going to go under a table, then I will run over and shut the robot off, and the task will be counted as failed.

You have two guidelines in completing the task. I want you to complete the task as fast as possible, and with as few collisions as possible. Do you have any questions?

Bibliography

[1] J. Adams, R. Bajcsy, J. Kosecka, V. Kumar, R. Mandelbaum, M. Mintz, R. Paul, C. Wang, Y. Yamamoto, and X. Yun. Cooperative material handling by human and robotic agents: Module development and system synthesis. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 200–205, 1995.

[2] A. Agah and G. Bekey. Efficiency assessment of performance of decentralized autonomous multi-robot systems. *Journal of Robotics and Mechatronics*, 8(3):286–291, 1996.

[3] R. Arkin. Motor schema-based mobile robot navigation. *The International Journal of Robotics Research*, 8(4):92–112, August 1989.

[4] R. Arkin. Reactive control as a substrate for telerobotic systems. *IEEE AES Systems Magazine*, 6(6):24–31, June 1991.

[5] R. Arkin. Cooperation without communication: Multiagent scema-based robot navigation. *Journal of Robotic Systems*, 9(3):351–364, April 1992.

[6] R. Arkin. *Behavior-Based Robotics*. MIT Press, 1998.

[7] R. Arkin and K. Ali. Integration of reactive and telerobotic control in multiagent robotic systems. In *From animals to animats 3: Proc. Third International Conference on Simulation of Adaptive Behavior*, pages 473–478, 1994.

[8] R. Arkin, T. Balch, and E. Nitz. Communication of behavioral state in multiagent retrieval tasks. In *Proc. 1993 IEEE International Conference on Robotics and Automation*, pages 588–594, 1993.

[9] R. Arkin and J. Hobbs. Dimensions of communication and social organization in multi-agent robotic systems. In *From animals to animats 2: Proc. 2nd International Conference on Simulation of Adaptive Behavior*, 1992.

[10] H. Asama, A. Matsumoto, and Y. Ishida. Design of an autonomous and distributed robot system: Actress. In *Proc. of the 1989 IEEE/RSJ International Workshop on Intelligent Robots and Systems*, pages 283–290, 1989.

[11] P. Backes. In S. Skaar and C. Ruoff, editors, *Teleoperation and Robotics in Space*, chapter 6, pages 139–158. Institute of Aeronautics and Astronautics, Washington, D.C., 1994.

[12] T. Balch and R. Arkin. Communication in reactive multiagent robotic systems. *Autonomous Robots*, 1(1):27–52, 1994.

[13] T. Balch and R. Arkin. Motor schema-based formation control for multiagent robot teams. In *1995 International Conference on Multiagent Systems*, pages 10–16, 1995.

[14] A. Bejczy. In S. Skaar and C. Ruoff, editors, *Teleoperation and Robotics in Space*, chapter 5, pages 107–138. Institute of Aeronautics and Astronautics, Washington, D.C., 1994.

[15] R. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1):14–23, 1986.

[16] Y.U. Cao, A. Fukunaga, A. Kahng, and F. Meng. Cooperative mobile robotics: Antecedents and directions. In *Proc. of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 226–234, 1995.

[17] J. Deneubourg and S. Goss. Collective patterns and decision-making. In *Proc. of the International Conference on Evolution and Ecology of Social Behaviour*, 1988.

[18] W. Dickson and R. Cannon Jr. Experimental results of two free-flying robots capturing and manipulating a free-flying object. In *Proc. of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 51–58, 1995.

[19] J. Draper. *Manual, Voice, and Automated Control of a Teleoperator Viewing System.* PhD thesis, University of Tennessee at Knoxville, 1987.

[20] J. Draper, S. Handel, E. Sundstrom, J. Herndon, Y. Fujita, and M. Maeda. Final report: Manipulator comparative testing program. Technical Report ORNL/TM-10109, Oak Ridge National Laboratory, 1987.

[21] N. Franks. Teams in social insects: group retrieval of prey by army ants (eciton burchelli). *Behavioral Ecology and Sociobiology*, 18:425–429, 1986.

[22] T. Fukuda and S. Nakagawa. A dynamically reconfigurable robotic system (concept of a system and optimal configurations). In *Proc. of the International Conference on Industrial Electronics, Control, and Instrumentation*, pages 588–595, 1987.

[23] D. Gage. Command control for many-robot systems. *Unmanned Systems Magazine*, 10(4):28–34, Fall 1992.

[24] S. Graves. Action selection in teleautonomous systems. In *Proc. of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 14–19, 1995.

[25] C. Guo, T. Tarn, N. Xi, and A. Bejczy. Fusion of human and machine intelligence for telerobotic systems. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 3110–3115, 1995.

[26] B. Hannaford. In S. Skaar and C. Ruoff, editors, *Teleoperation and Robotics in Space*, chapter 4, pages 87–106. Institute of Aeronautics and Astronautics, Washington, D.C., 1994.

[27] D. Hix and H. Hartson. *Developing User Interfaces: Ensuring Usability Through Product & Process*. John Wiley & Sons, Inc., 1993.

[28] T. Ishikawa, K. Kawabata, Y. Ueda, H. Asama, and I. Endo. Graphical user interface for collaborative system of human and mobile robots with sensors. In T. Lueth, R. Dillmann, P. Dario, and H. Worn, editors, *Distributed Autonomous Robotic Systems 3*, pages 309–318. Springer-Verlag, Berlin, 1998.

[29] A. Kirlik, B. Plamondon, L. Lytton, R. Jagacinski, and R. Miller. Supervisory control in a dynamic and uncertain environment: Laboratory task and crew performance. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(4):1130–1138, July/August 1993.

[30] G. Kondraske. Human performance: Science or art? In *Proc. of the Thirteenth Northeast Bioengineering Conference*, pages 44–47, 1987.

[31] C. Kring, S. Schrock, J. Draper, S. Hayashi, H. Kawakami, S. Horii, E. Omori, and Y. Fujita. Results of the manipulator test test stand study: Comparison of manipulators and evaluation of equipment items. Technical Report ORNL/TM-11023, Oak Ridge National Laboratory, Oak Ridge, TN, 1989.

[32] R. Larsen and M. Marx. *An Introduction to Mathematical Statistics and Its Applications*. Prentice Hall, 2nd edition, 1986.

[33] D. MacKenzie, J. Cameron, and R. Arkin. Specification and execution of multiagent missions. In *Proc. 1995 Int. Conf. on Intelligent Robotics and Systems IROS '95*, pages 51–58, 1995.

[34] M. Maimone, L. Matthies, J. Osborn, E. Rollins, J. Teza, and S. Thayer. A photo-realistic 3-d mapping system for extreme nuclear environments: Chornobyl. In *1998 IEEE/RSJ International Conference on Intelligent Robotic Systems (IROS)*, pages 1521–1527, 1998.

[35] M. Mataric. Minimizing complexity in controlling a mobile robot population. In *Proc. of the 1992 IEEE International Conference on Robotics and Automation*, pages 830–835, 1992.

[36] M. Mataric. Synthesizing group behaviors. In *Proc. of the Workshop on Dynamically Interacting Robots, International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 1–10, 1993.

[37] J. Matijevic. Sojourner : The mars pathfinder microrover flight experiment. *Space Technology*, 17(3/4):143–149, 1997.

[38] Matlab application toolbox: Statistics. The Math Works, Inc.

[39] D. McFarland. Towards robot cooperation. In *From animals to animats 3: Proc. 3rd International Conference on Simulation of Adaptive Behavior*, 1994.

[40] Y. Nakauchi, T. Okada, N. Yamasaki, and Y. Anzai. Multi-agent interface architecture for human-robot cooperation. In *Proc. of the 1992 IEEE International Conference on Robotics and Automation*, pages 2786–2788, 1992.

[41] J. Neter, M. Kutner, C. Nachtsheim, and W. Wasserman. *Applied Linear Statistical Models*. Irwin, 4th edition, 1996.

[42] F. Noreils. From controlled reactivity to man/machine interface through planning: Towards a robust architecture for mobile robots. In *1991 ISART*, pages 61–68, 1991.

[43] F. Noreils and A. Recherche. Adding man/machine interface to an architecture for mobile robots. In *Proc. of the 1991 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1991.

[44] K. Ohkawa, T. Shibata, and K. Tanie. Method for controlling a group of robots by an operator. In T. Lueth, R. Dillmann, P. Dario, and H. Worn, editors, *Distributed Autonomous Robotic Systems 3*. Springer-Verlag, Berlin, 1998.

[45] R. Paul, C. Sayers, and J. Adams. Operabotics. In *Proc. of International Symposium on Microsystems Intelligent Materials and Robotics*, 1995.

[46] K. Schreiner. Operation: microrobot. *IEEE Intelligent Systems & their applications*, 14(1):5–7, January/February 1999.

[47] T. Sheridan. *Telerobotics, Automation, and Human Supervisory Control.* The MIT Press, Camberidge, Massachusetts, 1992.

[48] S. Skaar and C. Ruoff, editors. *Teleoperation and Robotics in Space.* Institute of Aeronautics and Astronautics, Inc., Washington, D.C., 1994.

[49] M. Skubic, G. Kondraske, J. Wise, G. Khoury, R. Volz, and S. Askew. A telerobotics construction set with integrated performance analysis. In *Proc. of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 20–26, 1995.

[50] M. Stein and R. Paul. Operator interaction, for time-delayed teleoperation, with a behavior-based controller. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 231–236, 1994.

[51] G. Sukhatme. *On the Evaluation of Autonomous Mobile Robots.* PhD thesis, University of Southern California, May 1997.

[52] T. Suzuki, T. Fujii, H. Asama, K. Yokota, H. Kaetsu, N. Mitomo, and I. Endo. Cooperation between a human operator and multiple robots for maintenance tasks at a distance. In H. Asama, T. Fukuda, T. Arai, and I. Endo, editors, *Distributed Autonomous Robotic Systems 2*, pages 50–59. Springer-Verlag, Tokyo, 1996.

[53] T. Suzuki, T. Fujii, K. Yokota, H. Asama, H. Kaetsu, and I. Endo. Teleoperation of multiple robots through the internet. In *Proc. of the 5th IEEE International Workshop on Robots and Human Communication*, pages 84–89, 1996.

[54] T. Suzuki, K. Yokota, H. Asama, H. Kaetsu, and I. Endo. Cooperation between the human operator and the multi-agent robotic system: Evaluation of agent monitoring methods for the human interface system. In *Proc. of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 206–211, 1995.

[55] S. Tachi, H. Arai, and T. Maeda. Robotic tele-existence. In *Proceedings of the NASA Conference on Space Telerobotics*, 1989.

[56] Ugv demo 2 concluding notes. Technical report, DARPA.

[57] K. Yokota, T. Suzuki, H. Asama, A. Matsumoto, and I. Endo. A human interface system for the multi-agent robotic system. In *Proc. of the 1994 IEEE International Conference on Robotics and Automation*, pages 1039–1044, 1994.