Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation

Terry Fong

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania

23 January 1998

Abstract

Telerobotic systems have traditionally been designed and operated from a human point of view. Though this approach suffices for some domains, it is clearly sub-optimal for tasks such as operating multiple vehicles or controlling planetary rovers. Thus, I believe it is worthwhile to examine a new teleoperation approach: *collaborative control*. In this robot-centric model, instead of the human always being "in charge", the robot works as a peer and makes requests of the human. In other words, the human is treated as an imprecise, limited source of planning and information, just like sensors and maps and other noisy modules. To examine the numerous human-machine interaction and system design issues raised by this new approach, I propose to build a vehicle teleoperation system based on collaborative control. In my research, I will show how this approach enables efficient teleoperation and optimizes use of human resources.

1 Introduction

1.1 Traditional Teleoperation

Throughout the history of telerobotics, systems have been human-centric. Since telerobotics evolved directly from other human controlled systems, this approach seems only natural. Whatever the nature of the system (be it a light switch, washing machine or telerobot) and regardless the operating method (direct or supervisory control) the dominant paradigm has always been *human as controller*: the human receives information, processes it, and selects an action. The action then serves as control input to the system. For telerobotics, however, this human-machine relationship often proves to be inefficient and ineffective.

The first problem with *human as controller* is that it can result in an overuse of valuable human resources and may awkwardly bind the system's capability to the operator's skill. These difficulties are particularly acute with direct teleoperation. Some of the common problems are: operator handicap (limited skill, knowledge, attention), sensorimotor system limitations (reactions, decision making bandwidth), cognitive errors (incorrect mental model, sub-optimal decisions), perceptual problems (misclassification, judgement errors), and physical difficulties (simulator sickness, nausea, fatigue) [Ferrel67][Murphy96][Sanders93][Sheridan92].

The second *human as controller* problem is that the quality of the human-machine connection significantly impacts performance. An effective operator interface (or *control station*) is critical for conveying information and feedback to the operator. Poor displays, inappropriate modeling, and inefficient control inputs can all contribute to operator error [Murphy96]. Additionally, if the operator and robot are widely separated, communications may be affected by noise, power limits or signal transmission delays. Delay is particularly insidious because it can make direct teleoperation impractical or impossible [Sheridan93].

The third manner in which *human as controller* causes problems is the imbalance in roles (human as supervisor, robot as subordinate). Since the operator is forced to remain "in-the-loop", she has reduced capacity for performing other tasks, leading to a reduction of work efficiency. Additionally, since the robot is controlled by the human, the system halts whenever the robot has to wait for directives. The imbalance in roles is also reflected in the operational dialogue: the human gives commands and the robot responds. This one-sided conversation means that the relationship between human and robot is forever static.

1.2 Case in Point: Vehicle Teleoperation

Consider the task of remotely driving a robotic vehicle. There are three basic problems: figuring out where the vehicle is, determining where it should go, and getting it there. These problems can be difficult to solve, particularly if the vehicle must be moved in a hazardous environment using a poor communication link. This is fairly common occurrence with exploration robots (subsea, planetary) [Mishkin97][Hine95] and unmanned ground vehicles (reconnaissance, surveillance) [Shoemaker90]. The problems are further exacerbated when we add additional constraints: limited operator resources (e.g., time available for teleoperation), multiple vehicles, extremely low bandwidth, and environments with active hazards.

Vehicle teleoperation has often proven to be fraught with difficulty. Sandia National Laboratory conducted a study of rate-controlled ground vehicles and reported operator problems including: slow driving, imprecise control, loss of situtational awareness, poor attitude and depth judgement, and failure to detect obstacles [McGovern88]. Additionally, poor communications has been shown to reduce system efficiency and performance. The Sojourner rover on Mars, for example, was connected to it's earth-based operator via a 9600 baud link with a delay of up to 40 minutes. Since Sojourner could only provide limited feedback, it was operated via supervisory commands once per day, severely limiting the science return [Mishkin97].

1.3 A Novel Approach: Collaborative Control

As we have seen, there are numerous problems and limitations arising from the conventional *human as controller* model. Since we would like to construct teleoperation systems which are able to robustly operate in difficult environments, in spite of poor communications, and with high performance regardless of operator differences we need a new approach. Instead of *human as controller*, therefore, I propose the following thesis:

Teleoperated systems can be significantly improved by modeling
the human as *collaborator* rather than *controller*

In this new *collaborative control* model a human operator and robot are peers who work together, *collaborating* to perform tasks and to achieve goals. Instead of a supervisor addressing (or dictating to) a subordinate, the human and the robot engage in *dialogue* to exchange their ideas and resolve their differences. Thus, instead of the human always being “in charge”, we allow the robot to be more equal and to take control. Moreover, we allow the robot to decide how to use human advice: to follow it when available and relevant; to modify (or ignore) it when inappropriate or unsafe. For example, if the robot is operating autonomously and has problems, it can ask the operator “what should I do?” If the human is capable of responding and can do so in a timely fashion, then the advice will be used. However, if the advice is not timely (e.g., communication delay) or if it is unsafe (e.g., “drive off that cliff”), then the robot may view the advice with skepticism.

In short, when we construct a teleoperation system, rather than designing only from a human-centric viewpoint (*human as controller*), we also consider issues from a robot-centric perspective (*human as collaborator*). This is not to say that robot becomes “master”: it is still a subordinate following higher-level strategy (goals and tasks) set by the operator. However, with collaborative control, the robot has more freedom in execution and is able to better function if the operator is distracted, inattentive, making errors, and so on.

The term *collaborative control* is quite apt. I use it because it is directly analogous to the interaction between human collaborators. Specifically, when we engage in collaboration, we encourage each collaborator to work jointly with others towards a common goal. We also allow each collaborator to take self-initiative and to contribute in the manner in which she is best suited. At the same time, however, we leave room for discussion and negotiation to occur, so that potential solutions are not missed.

Collaborative control raises many human-machine interaction and system design issues. To examine some of these issues, and to demonstrate that we can build better teleoperation systems, I propose to build a vehicle teleoperation system based on collaborative control. In my research, I will show how collaborative control enables efficient teleoperation and optimizes use of human resources. In order to limit the scope of my work, I intend to focus on vehicle mobility issues (remote driving, navigation, etc.) and not examine broader vehicle teleoperation (e.g., remote task execution).

2 Related Research

2.1 Supervisory Control

The supervisory control concept appeared as part of research on how earth-based operators might teleoperate lunar vehicles[Ferrel67]. The term *supervisory control* is derived from the analogy between a supervisor's interaction with subordinate staff in a human organization and an operator's interaction with a robot [Sheridan92]. To effect supervisory control, the operator must be able to divide a problem into a sequence of sub-tasks which the robot can successfully achieve on its own. The majority of research in supervisory control has focused on telemanipulation (e.g., [Blackmon96]) rather than vehicle teleoperation. Among the researchers who do discuss vehicle teleoperation are [Wettergreen95], [Lin95], and [Stone96].

In a sense, supervisory control models traditional military structure: it is strictly hierarchical, has rigid control flow, the supervisor is “in charge” and the subordinates are restricted in what they can and cannot do. Collaborative control more closely resembles a research group. Although there is hierarchy, control is more flexible and dynamic. Furthermore, each collaborator has more freedom to take the initiative and to lead.

2.2 Multi-operator and cooperative teleoperation

In multi-operator teleoperation, multiple operators share or trade control. [Cannon97] describes the use of “virtual tools” for telemanipulation. In his system, operators use these tools to define key actions at a supervisory level. A networked operator interface allows multiple operators to share control. Cannon refers to this interaction as “collaborative control” since multiple human operators collaborate to effect control.

Cooperative teleoperation, also known as *teleassistance*, tries to improve the teleoperation process by supplying aid (data filtering, decision-making tools, etc.) to the operator in the same manner an expert would render assistance. For example, [Murphy96] describes a teleassistance system which combines a limited autonomy robot architecture with a knowledge-based operator assistant. During teleoperation, this system provides “strategic assistance” so that the operator and robot can cooperate in cognitively demanding tasks.

2.3 Human-Robot Control Architectures

Although most robot control architectures are designed for autonomy, some have addressed the problem of mixing humans with robots. One approach is to directly incorporate humans into the design, i.e., treating human perception or decision making as a system element. DAMN, for example, is a behavior-based architecture in which individual modules vote for and against a range of possible actions[Rosenblatt95]. Command arbitration allows modules as disparate as autonomous safety behaviors and teleoperation to coexist.

Another approach is the use of prioritized control, in which operator commands may be overridden by autonomous modules. The best-known example of this type is NASREM, which explicitly incorporated an operator interface into a layered, hierarchical control system [Albus87]. More recently, the concept of *safeguarded teleoperation* has been used to enable novices to teleoperate a planetary rover[Krotkov96].

2.4 Vehicle Teleoperation Systems

During the past twenty years, the majority of work in vehicle teleoperation has centered on rate-controlled systems for use in hazardous environments. These remote vehicles are typically operated with single-mode “inside-out” control: the operator controls the vehicle's rotation and translation rates via hand-controllers and receives feedback from on-board video cameras and sensors. [McGovern88] reports on a large body of this work with a fleet of vehicles, ranging from small indoor robots to large outdoor military automobiles.

More recently, vehicle teleoperation systems have emphasized the use of multi-modal operator interfaces and supervisory control. Multi-modal interfaces provide the operator with a variety of control modes (individual actuator, coordinated control, etc.) and displays (numeric, visual, haptic, etc.). Supervisory control is used to compensate for a variety of problems, most often poor communications. Notable systems include: Dante II [Fong95], STRIPE [Kay97], VEVI [Hine95], Nomad [Wettergreen96], and Sojourner [Mishkin97].

3 Current work

3.1 Vehicle Teleoperation

During the past year, I have been developing a safeguarded vehicle teleoperation system. My system consists of a small, semi-autonomous robot, wireless communications, and a multi-modal operator interface. The system architecture is shown in Figure 1 below:

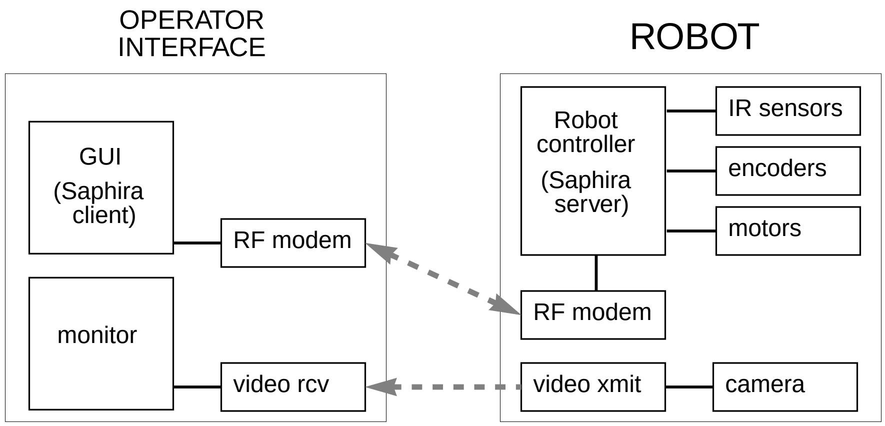

Figure 1: Vehicle teleoperation system architecture

I am currently using a Koala robot (see Figure 2), a small, skid-steered wheeled vehicle capable of limited outdoor work. The Koala is equipped with a ring of infrared proximity sensors, power monitoring, and wheel encoders. A fixed CCD camera provides forward-looking video. Analog transmitters are used for video and data communication. Hardware control is performed by an on-board microprocessor.

Figure 2: Koala with camera, video transmitter, and RF modem

The Koala is controlled with Saphira, an integrated sensing and control system which provides task coordination, environment modeling, and reasoning[Konolige96]. Saphira is implemented via a client / server model. A server process runs on the Koala's microprocessor and provides basic services including motion control, localization, sensor management, and status. A client process runs on a different processor and controls “high-level” operations including fuzzy behaviors (obstacle avoidance, tracking, etc.), perception, and task-level control. The Saphira controller currently supports independent motor control, position and rate control, Cartesian control, safeguarding (proximity obstacle avoidance) and simple navigation tasks.

I am using a 2-D graphical user interface, which incorporates the Saphira client, for direct access to the robot controller and for feedback (current vehicle pose, status, behavior / task activity). Thus, I am able to command the Koala in a variety of modes from direct motor control to safeguarded, supervisory control.

3.2 Experiences

I have used this system for remote driving at EPFL. In practice, the system's safeguarded teleoperation approach suffers from poor proximity sensing (IR's are range limited and cannot detect "thin" obstacles) and inadequate proprioception (no tilt sensing). As a result, safeguarding has proven to be unreliable and the operator is forced to maintain continuous attention whenever the robot is in motion.

Table 1: Observed teleoperation problems

Problem	Cause
imprecise control (tracking errors, oversteering)	system latencies (communications, processing)
failure to detect obstacles difficulty understanding remote environment	ambiguous video images unusual (low-height) camera viewpoint
vehicle rollover and pitchover	inability to stop in time, judgement errors
judgement errors (distance, orientation)	unusual (low-height) camera viewpoint
loss of situtational awareness	inadequate pose feedback, unstabilized video

The system also suffers many problems (see Table 1) commonly encountered in vehicle teleoperation. Moreover, remote driving performance (precision, task time, etc.) is clearly linked to operator skill and experience. To remedy these problems, it is clear that a better approach (i.e., collaborative control) is required.

4 Research Issues

4.1 Human-Computer Interaction (HCI)

When we build a collaborative control system, the traditional roles of operator and robot change. To the human, the robot becomes a co-worker seeking dialogue, instead of a subordinate awaiting direction. Though the human may make requests, there is no need for the robot to strictly obey them. This frees the human from performing continuous control or supervision. If the human is available, she can provide direction. But, if she is not, the system can still function. This allows use of human perception and cognition without requiring time-critical response. It also provides a framework for an operator to control multiple robots.

To the robot, the human is no longer omniscient, but is more like a peer who can provide opinions. Thus, the robot is free to use the human such that its needs are best satisfied. This means that the robot is able to query the human in different ways and frequencies. At the same time, however, the robot has to be more self-sufficient. Specifically, the robot needs to recognize to whom it is talking and change its manner of speech accordingly. After all, one does not talk to a novice the same way as to an expert. Lastly, the robot needs to decide if the human is "unhelpful" (i.e., the operator is unavailable or is endangering the system).

Collaborative control also changes the way we view telerobotics. In conventional systems, there is an underlying notion of "robot as tool": the robot is used to extend human sensing and acting. This parallels the "computer as tool" model [Laurel86] and suffers the same limitations (e.g., failure to hold user interest). With collaborative control, the robot is more equal, more "robot as partner". Though the robot may ask for approval, it is not *required* to do so. Thus, interaction with a collaborate system will differ from expectations.

Overall, these HCI issues raise questions such as:

- At what level does the robot need to model the human in order to converse appropriately? Will requests to a particular human depend on what kinds of answers have been received over time?
- Does the robot need to provide context (history, supporting information) when making a request?
- How to decide who is currently and ultimately "in charge"? Can the user always mandate action?

4.2 Dialogue

In order to use collaborative control, we must build the system with the capacity for dialogue. That is, we need to enable the operator and the robot to converse, to exchange opinions, and to negotiate. Each one should be able to say "here is what I think", to ask "what do you think?" and to interpret the response:

robot: "Based on *a-priori* map data, I think I should go to waypoint C directly."

operator: "I think it is better if you go to A, then to C via B."

robot: "That's an interesting opinion, but I'm currently stuck at A. Why don't you look at this image and tell me how I should proceed."

operator: "Sorry, I'm too busy to look at it right now."

robot: "Okay, I'm going to randomly wander until I find an unobstructed path. When you are ready, I will tell you what happened."

Though simple, there is a strong dynamic at work here. Initially, the robot seeks advice and the human provides a response. Since the response is unhelpful, the robot clarifies with additional detail. By this point, the human has become unavailable. So, the robot takes the initiative and (momentarily) suspends the dialogue. In this scenario, we see the human and the robot working as peers, not supervisor and subordinate,

In short, good dialogue is two-way: it requires each party to understand who they are talking to, what the other is capable of, and to speak so the other can understand. To an extent, traditional teleoperation has dialogue (i.e., the feedback loop), but the conversation is limited. Dialogue offers the potential for richer, more flexible teleoperation. However, it also creates many research questions including:

- How does the robot decide when to say something? How does it decide what is the "right question to ask" (most urgent, most in need of clarification) for a given situation?
- How does the robot format its queries and interpret the responses?
- How does the "manner of speech" change with different operators (i.e., as skill level, knowledge or rationality changes)?

4.3 Interface Design

In traditional teleoperation, the user interface serves the operator: displays provide information for decision making, mode changes are human triggered, etc. In a collaborative control system, however, the user interface has to support dialogue and to also serve the robot. Thus, it is not clear how we should apply, or to what extent we must adhere to, conventional user-interface design methods.

In conventional user-centered interface design, the fundamental goal is to support human activity: to enable humans to do things faster, with fewer errors, and with greater quality[Newman95]. The interface mediates conversation (following the pace and direction of the activity) between the user and the underlying application. A variety of human performance or usability metrics (speed of performance, incidence of errors, retention of learned skills, aesthetics, etc.) guide the design process and allow objective evaluation.

We can use this approach when designing a collaborative control interface. To support dialogue, for example, we know that we must convey rich information to and from the operator. To avoid overloading, this means the interface must support compact forms of expression. Thus, if the robot queries for areas to avoid, it should be able to provide an image (or a map) on which the user can directly respond (e.g., circling regions with a pen). In other words, we maximize usability by choosing dialogue support appropriately.

It is clear, however, that a user-centered approach has limitations. If we focus only on human usability, we will certainly find that the interface fails to support the robot. For example, what do we do if the robot needs to operate in a mode other than the one selected by the operator? If we abruptly switch the interface configuration for the robot, we may disorient the operator.

Thus, collaborative control forces us to consider the following interface design questions:

- How to apply conventional, user-centered design methods?
- How to incorporate the robot’s needs in the design process? What metrics are appropriate?
- How should the interface operate? Under strict user control? Or shared with the robot?

4.4 System Design

Collaborative control adds new constraints to system design. In addition to conventional issues (e.g. architecture), we must also consider the impact of dialogue and peer interaction. In traditional teleoperation, the flow of control is clear: the operator controls the robot's actions. Though she may share or trade control, she retains ultimate authority. Collaborative control, however, allows control to be negotiated. It also allows the robot to consider human commands as *approximate* and to function more freely. Thus, a collaborative control system must provide a mechanism for deciding who is “right” and who should be “in control”.

Similarly, the handling of information requires more flexibility with collaborative control. Since the human and robot perceive and make decisions differently, the system must be able to handle sensor data in a variety of ways. For example, the human may decide that the terrain is flat from a camera image; the robot may decide the same terrain is rough using proprioception.To have meaningful dialogue (e.g., “why do you say it's rough when it's obviously flat?”), both need to be able to exchange their data in a coherent manner.

Perhaps the most difficult system issue to resolve, however, is what to do with invalid advice. Consider the situation in which the human answers a query, but by the time she does the robot has already found an answer itself. Should the robot ignore the human answer or should it reconsider it's actions? The problem is worse when we consider that out-dated advice is hard to distinguish from unsafe advice. Thus, if we allow a range of users (child to expert), how do we cope with the varying speed and quality of information?

At the very least, we must confront the following system design issues:

- How do we decide what action to take? How do we know which control directive is valid and appropriate for a given situation? What control modes are appropriate?
- How do we filter sensor data for use by both the operator and the robot?
- How to incorporate (interpret, classify, use, reject) seemingly irrelevant advice?

5 Proposed work

To answer some (but not all) of these research questions, I propose building the Collaborative Architecture for TeleOperation (CATO). I will use my current system as a base, but will make substantial changes and improvements. Then, I will conduct a number of remote driving experiments with a single mobile robot and with multiple mobile robots.

5.1 Collaborative Controller

I will address questions in HCI, dialogue and system design by building a collaborative controller to mediate between operator and robot. My current design (Figure 3) contains: *Query Arbiter* (selects user queries), *User Modeler* (adapts dialogue), *Sensor Manager* (manipulates data for display), *Event Archiver*, *Controller Manager* (interface to robot controller), *Visual Servo* (object tracker), STRIPE (path tracker), *Input Manager* (interpret user input) and *Collaboration Manager* (controller kernel). I am considering using a message-based architectures (IPT, NDDS, etc.) to connect the modules. The design will support multiple users and robots.

I anticipate that the most difficult work will be to build the *Controller Manager*, *Query Arbiter*, and the *User Modeler*. The *Controller Manager* will decide who (operator or robot) is “in charge” of the system and what action to take. Thus, the manager will have to arbitrate commands, select control modes and interact with the Saphira controller. The *Query Arbiter* will decide when and what to ask the human. Since several pro-

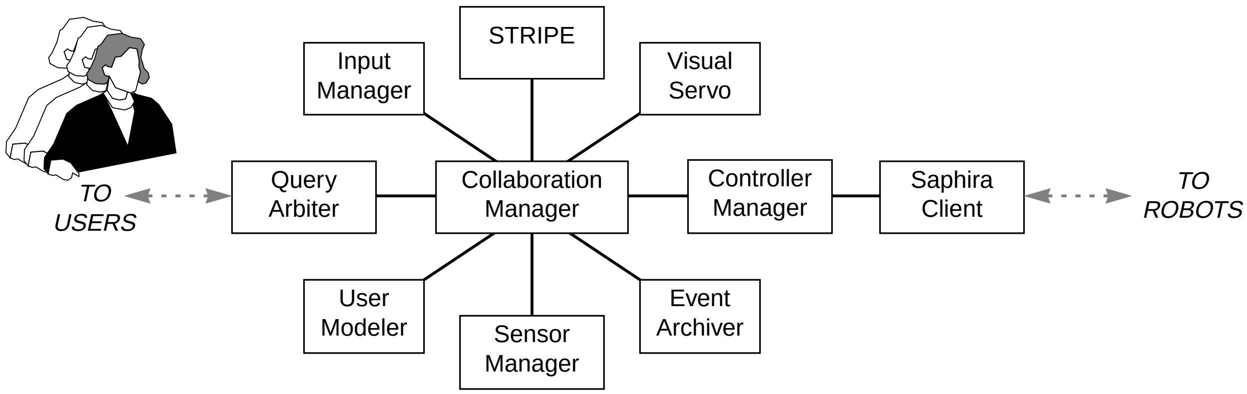

Figure 3: Collaborative controller (preliminary design)

cesses may all have requests to make of the user (in different formats and frequencies) the arbiter will need to be smart enough to decide which process gets priority. Lastly, the *User Modeler* will be used to vary the type of user requests and the "manner of speech". To do this, the modeler must be able to evaluate the user and the answers received over time, then decide how to converse appropriately.

As previously mentioned, dialogue raises many design and research questions. To limit the scope of my work, therefore, I plan to use dialogue only to address vehicle mobility issues (e.g. navigation) and not broader topics enabled by this approach (e.g., use of natural language). The dialogue supported in CATO is given below (Table 2). The resulting control and information flow is shown in Figure 4.

Table 2: Operator / robot dialogue for vehicle mobility

Operator to Robot	**Robot to Operator**
direct motion control (position, rate, Cartesian)	status (pose, rates), query (how to move)
path following: track curve or waypoints	status (path progress)
visual servo: follow object	status (follow progress)
image / map region: explore or avoid	constraint query (region to avoid or to go to)
query (current status, event summary)	status (current, event summary)
query (sensor data, pose)	sensor data display (image, map)
confirm (yes / no, percent agreement, choice)	action query (approval, mode selection)
assume direct teleoperation control	alert

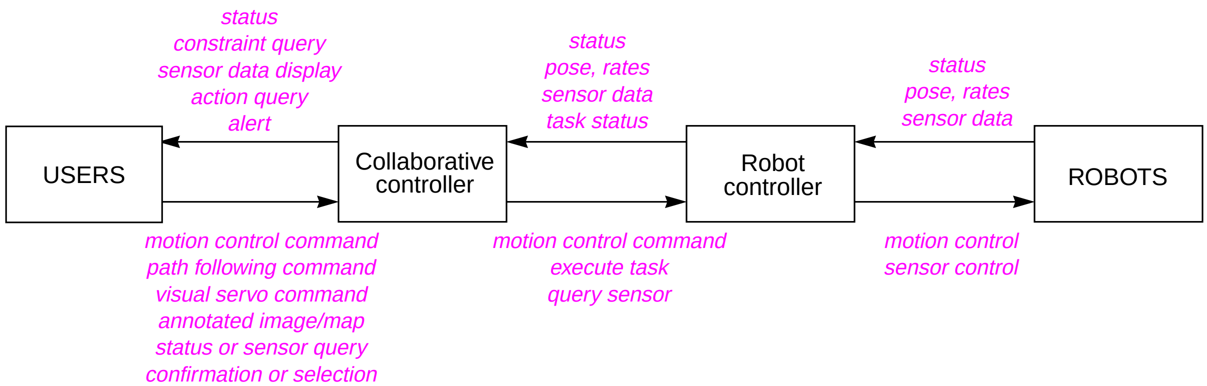

Figure 4: Control and information flow

5.2 User Interface

I will address questions in interface design by designing and building a non-intrusive user interface. By "non-intrusive", I mean that the interface will not excessively consume human resources (attention, cognitive capacity, etc.). The guiding principles for its design will be:

- high usability (should be usable by mom, unbreakable by a baby)
- low cognitive workload
- user should be able to "tell at a glance" what is occurring and what has happened

At this time, I am planning to implement the user interface using a touch screen. I believe this type of device is well-suited for a "non-intrusive" interface (e.g., it is efficient for rapid input), yet provides sufficiently rich interaction to allow study of the research issues described in Section 4.3. I anticipate the primary difficulties will be determining how to incorporate the robot's needs into the design process, whether the interface should only be controllable by the user, and how to tailor the display to each user's preferences.

5.3 Experiments

To validate and assess the effectiveness of collaborative control, I am planning to conduct a number of experiments. In these tests, I will try to answer the questions raised in Section 4 as well as the following:

- under what conditions (scenario, task, environment, resources) will this approach work?
- under what conditions will the system fail?
- how general and scalable (multiple operators, multiple robots) is the approach?

In order to acquire my data, I will conduct field tests and a human performance study of a remote driving task. To bound the study, I will only investigate single operator / single robot and single operator / multiple robot scenarios. One potential test scenario is to instruct an operator to "drive from A to B" while varying her cognitive load. In my experiments, I plan to examine some of the following:

- *independent variables*: communication link (bandwidth, latency, loss), user resource (cognitive capacity, attention, time available), user (skill, training, knowledge, rationality), percent of system activity involving human interaction, dialogue (type, amount)
- *dependent variables* (metrics): system performance (time to completion, precision, task achievement), usability (ease of use, ease of learning), awareness (spatial, situational), operator workload (fatigue), flexibility (adaptability to task or environment changes)

Finally, I plan to perform a detailed error analysis of my collaborative control system design and of the experimental data. In this analysis, I will ascertain and classify the sources and the impact of errors. I am specifically interested in understanding the influence of ambiguous dialogue caused by sensor noise, user input, etc. on system performance.

5.4 Robot Improvements

In order to perform these experiments, I need to make a number of changes to my current system. Thus, I plan to make three primary hardware improvements on the Koala. First, I will add a three-axis orientation sensor. This will allow proprioceptive safeguarding (e.g., rollover protection) and improved localization. Second, I will replace the monochrome camera with a color CCD camera. This should help operators perceive the remote environment more precisely. Finally, I will add a ring of short-range distance sensors to improve proximity obstacle detection and avoidance. Most likely, I will use narrow-beam ultrasonic sonar.

In parallel with the Koala improvements, I will make similar modifications to a Pioneer AT mobile robot. The Pioneer AT has ultrasonic sonar and can traverse rougher terrain than the Koala. I also plan to make any necessary changes to use CMU's Pandora robot. Lastly, I plan to improve CATO's Saphira controller by adding proprioceptive safeguarding, improving obstacle detection, and adding control modes.

6 Schedule

Spring 1998	robot hardware and control improvements collaborative controller development
Summer 1998	user interface development validation experiments
Fall 1998	complete software development implement system on Pandora
Spring 1999	remote driving experiments data collection and analysis
Summer 1999	thesis writing and defense

7 Conclusion

I believe that collaborative control will allow us to solve many of the problems associated with conventional *human as controller* teleoperation:

Table 3: solving *human as controller* problems

Problem	Solution
performance differences due to operator variation	vary dialogue and autonomy to fit the operator
simulator sickness, fatigue, and high workload	allow operator to step out of the control loop
operator perceptual and cognitive errors	verification and safeguarding through dialogue
low performance due to poor communications	vary control and information flow to fit the link
limited operator resources	guide operator to critical (urgent) items via dialogue
poor human-machine synergy	better communication and joint problem solving
inflexibility and lack of robustness	negotiated control, increased robot freedom of action

In short, I believe that collaborative control can compensate for inadequacies in autonomy, in human capabilities, and in communications. Furthermore, by reducing the need for continuous control, I believe that it will enable a single operator to productively operate multiple robots. Most importantly, since collaborative control will allow the operator and robot can work as partners, I believe we will be able to build more competent, flexible and robust telerobotic systems.

Thus, I expect to demonstrate the following in my thesis:

- a new model for vehicle teleoperation which is significantly better than existing methods
- the importance of dialogue for improving teleoperation performance and productivity
- a non-intrusive user interface which places minimal requirements on the operator (training, skill, knowledge, availability), on the communications link (bandwidth, latency), and on the interface hardware
- a teleoperation system which is robust, easy to use, and performs well in dynamic, uncertain, hazardous environments
- how collaborative control enables a single operator to robustly and efficiently teleoperate a single vehicle (in minimal time) and to control (simultaneously) a heterogeneous fleet of vehicles

8 References

[Albus87] Albus, J., et al., "NASA / NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)", NIST Tech. Note 1235, Gaithersburg, MD, July 1987.

[Blackmon96] Blackmon, T. and Stark, L., "Model-Based Supervisory Control in Telerobotics", Presence, Vol 5., No. 2, Spring 1996.

[Cannon97] Cannon, D. and Thomas, G., "Virtual Tools for Supervisory and Collaborative Control of Robots", Presence, Vol. 6, No. 1, February 1997.

[Ferrel67] Ferrel, W. and Sheridan, T., "Supervisory Control of Remote Manipulation", IEEE Spectrum, Vol. 4, No. 10, 1967.

[Fong95] Fong, T., et al., "Operator Interfaces and Network-Based Participation for Dante II", SAE 25th International Conference on Environmental Systems, San Diego, CA, July 1995.

[Hine95] Hine, B., et al., "VEVI: A Virtual Environment Teleoperations Interface for Planetary Exploration", SAE 25th International Conference on Environmental Systems, San Diego, CA, July 1995.

[Kay97] Kay, J., "STRIPE: Remote Driving Using Limited Image Data", Ph.D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, June 1997.

[Konolige96] Konolige, K. and Myers, K., "The Saphira Architecture for Autonomous Mobile Robots", in "Artificial Intelligence and Mobile Robots" (Bonasso, R. and Murphy, R., eds.), MIT Press, Cambridge, MA, 1997

[Krotkov96] Krotkov, E., Simmons, R., Cozman, F., and Koenig, S., "Safeguarded Teleoperation for Lunar Rovers", SAE 26th International Conference on Environmental Systems, Monterey, CA, July 1996.

[Laurel86] Laurel, B., "Interface as Mimesis", in "User Centered System Design" (Norman, D. and Draper, S., eds.), Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[Lin95] Lin, I., Wallner, F., and Dillmann, R. "An Advanced Telerobotic Control System for a Mobile Robot with Multisensor Feedback", in "Intelligent Autonomous System IAS-4" (Rembold, U. and Dillmann, R. eds.), IOS Press, 1995.

[McGovern88] McGovern, D., "Human Interfaces in Remote Driving", SAND88-0562, Sandia National Laboratory, Albuquerque, NM, March 1988.

[Mishkin97] Mishkin, A., "Operations and Autonomy for the Mars Pathfinder Microrover", briefing, NASA Jet Propulsion Laboratory, Pasadena, CA, April 1997.

[Murphy96] Murphy, R. and Rogers, E., "Cooperative Assistance for Remote Robot Supervision", Presence, Vol. 5, No. 2, Spring 1996.

[Newman95] Newman, W. and Lamming, M., "Interactive System Design", Addison-Wesley, Reading, MA, 1995.

[Rosenblatt95] Rosenblatt, J., "DAMN: A Distributed Architecture for Mobile Navigation", in "Proceedings of the 1995 AAAI Spring Symposium on Lessons Learned from Implemented Software Architectures for Physical Agents", Stanford, CA, March 1995.

[Sanders93] Sanders, M. and McCormick, E., “Human Factors in Engineering and Design”, McGraw-Hill, New York, NY, 1993.

[Sheridan92] Sheridan, T., “Telerobotics, Automation, and Human Supervisory Control”, MIT Press, Cambridge, MA, 1992.

[Sheridan93] Sheridan, T., “Space Teleoperation Through Time Delay: Review and Prognosis”, IEEE Transactions on Robotics and Automation, Vol. 9, No. 5, October 1993.

[Shoemaker90] Shoemaker, C., “Low Data Rate Control of Unmanned Ground Systems”, in “Proceedings: Association for Unmanned Vehicle Systems (AUVS-90)”, Dayton, OH, 1990.

[Stone96] Stone, H. W., “Mars Pathfinder Microrover A Low-Cost, Low-Power Spacecraft”, in “Proceedings of the 1996 AIAA Forum on Advanced Developments in Space Robotics”, Madison, WI, August 1996.

[Wettergreen95] Wettergreen, D., Pangels, H., and Bares, J., “Gait Execution for the Dante II Walking Robot”, In “Proceedings of the 1995 IEE Conference on Intelligent Robots and Systems”, 1995.

[Wettergreen96] Wettergreen, D., et. al., “NOMAD Operator Interface Development Plan”, Intelligent Mechanisms Group, NASA Ames Research Center, Moffett Field, CA, December 1996.