Collaborative Control of Robot Motion: Robustness to Error [*]

Ken Goldberg and Billy Chen

IEOR and EECS Departments, UC Berkeley

We consider "collaborative control" systems, where multiple sources share control of a single robot. These sources could come from multiple sensors (sensor fusion), multiple control processes (subsumption), or multiple human operators. Reports suggest that such systems are highly fault tolerant, even with large numbers of sources.

In this paper we develop a formal model, modeling sources with finite automata. A collaborative ensemble of sources generates a single stream of incremental steps to control the motion of a point robot moving in the plane. We first analyze system performance with a uniform ensemble of well-behaved deterministic sources. We then model malfunctioning sources that go silent or generate inverted control signals. We discover that performance initially improves *in the presence of malfunctioning sources and remains robust even when a sizeable fraction of sources malfunction. Initial tests suggest similar results with non-deterministic (random) sources. The formal model may also provide insight into how humans can share control of an online robot.*

1 Introduction

In this paper we consider "collaborative control" systems, where multiple sources share control of a single robot. Inputs from each source are combined to generate a single control stream for the robot. This definition is intended to include systems where the sources are multiple sensors (sensor fusion), multiple control processes (subsumption), or multiple human operators. There is a large body of research in this general area and many experiments demonstrate robustness. When statistical averaging is used to combine inputs, the resulting central tendencies can be robust to noise. In this paper we propose a formal model of collaborative control and study its behavior with simulation.

Figure 1 illustrates a collaborative control architecture. Can a group of n sources achieve anything resembling coherent control?

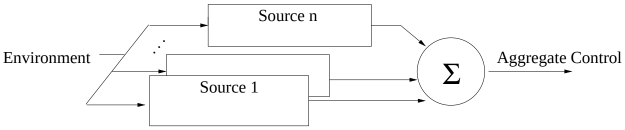

Figure 1: *In a collaborative control system, an ensemble of sources are aggregated into a single control signal.*

For example, consider a large group of scientists working together to control a telerobot as it excavates an archaeological site. Each scientist monitors a different camera or sensor and generates a motion control input appropriate to that sensor. These inputs are aggregated to produce a single control stream for the telerobot. If the inputs can be put into vector form, one aggregation algorithm is to compute the vector mean. Since each scientist has access to a different noisy sensor, the Central Limit Theorem suggests that the mean may yield a more effective control signal than that from any individual input (the CLT assumes independence and zero-mean noise, which are rarely satisfied in practice) [12].

Anecdotal evidence with Cinematrix [9, 21], an interaction system for human audiences, suggests that collaborative motion control is not only possible but surprisingly robust to deviations in individual behavior. In the early 1990's, Loren and Rachel Carpenter performed a series of experiments with this system. Each audience member is given a plastic paddle, colored red on one side and green on the other. By rotating his or her paddle each player simultaneously provides input. Overhead cameras detect which color is being presented by each participant in real time. The camera output is used to drive a live display projected onto the front screen of the theater. The average level of red or green conveyed by the group provides an aggregate audience signal that is re-computed several times a second.

The theater is divided down the central aisle and a cursor is projected on the screen. Participants on the right control the horizontal motion of the cursor, participants on the left control the vertical motion. A large circle is displayed on the screen and the audience is requested to move the shared cursor to trace a path around the circle. Since each player only controls one small component of the average signal, and the participants are a heterogeneous group with different personalities, one might conjecture that the shared cursor motion would resemble random Brownian motion.

But in repeated experiments, groups of participants were quickly able to adapt their individual paddle signals to achieve coherent motion control of the shared cur-

[*] IEEE/RSJ International Conference on Robots and Systems, October 2001, Maui, HI. This work was supported in part by NSF Award CDA-9726389. Contact: goldberg@ieor.berkeley.edu.

sor. Groups were not only able to track given trajectories, but to play competitive games such as Pong, and even to collaboratively control an airplane flight simulator! Audiences ranged from 5000 graphics professionals at Siggraph 1991 to groups of unruly high school students in Pittsburgh.

Although the number of sources is reduced, an ensemble of independent processes in a Subsumption-based or sensor-fusion robot control system resembles an ensemble of human participants in the Cinematrix system. To gain insight into the performance of such systems, we simulate the performance of each source with a finite automaton: each automaton takes as input the current position of the robot or cursor, and each generates a desired motion increment. The increments of the ensemble are averaged to obtain a single motion increment that is executed by the robot.

We start by simulating robot performance with an ensemble of well-behaved deterministic sources. We then model malfunctioning sources that go silent or generate inverted control signals, and simulate robot performance as malfunctioning sources are introduced. In the last section we describe preliminary results with non-deterministic (random) sources.

2 Related Work

In conventional robotics and telerobotics, one source (human or computer) controls a single robot. Most autonomous and human-teleoperated robot systems are characterized by a single feedback loop including sensors, actuators, and the environment as illustrated in 2.

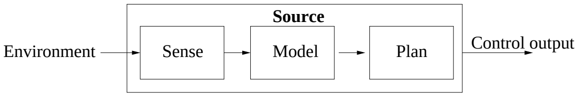

Figure 2: *Single source control architecture (based on Brooks 1986 [4]).*

Pirjanian studies how reliable robot behavior can be produced from an ensemble of sources [19]. Drawing on research in fault-tolerant software [16], Pirjanian considers systems with a number of homogenous sources (sharing a common objective), and considers a variety of voting schemes. He shows that fault-tolerant *behavior fusion* can be optimized using plurality voting [3] but does not study the motion paths generated by specific malfunctioning modes.

McDonald, Small, Graves, and Cannon [18] describe an Internet-based collaborative control system that allows several users to assist in waste cleanup using Point-and-Direct (PAD) commands [6]. In their system, collaboration is pipelined, with overlapping plan and execution phases. They demonstrate that collaboration improves overall execution time but they do not address conflict resolution between users. Chong et al [7] study a multi-operator-multi-robot (MOMR) networked system and propose several control methods to cope with collision arising from network time delays. Adapting their terminology, ours would be a multi-operator-single-robot (MOSR) system.

Fong et al use the term "collaborative control" to describe systems where collaboration occurs between a single mobile robot and a single human operator who is treated as a peer to the robot and modeled as a noisy information source [10]. Related models of single robot/single human collaboration are analyzed in [1, 17].

In [12] we described an Internet-based collaborative telerobotic system that averaged multiple human inputs to simultaneously control a single industrial robot arm. We reported preliminary experiments with maze-following that suggested that groups of humans perform better than single humans in the presence of noise, but complicating factors such as human variation and learning curves made that system difficult to analyze.

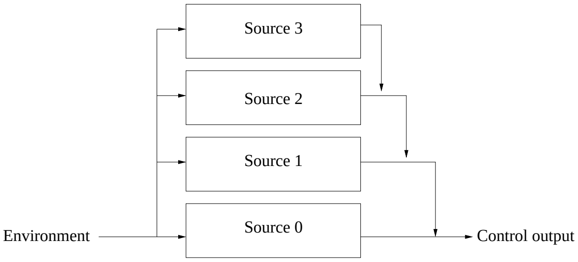

Figure 3: *Brooks' Subsumption Control Architecture. Groups of onboard computational processes interact to control a mobile robot. (Based on [4]).*

Collaborative Control is closely related to research in Subsumption control architectures, where groups of onboard computational processes interact to control an autonomous robot. As illustrated in Figure 3, Brooks [4] proposed Subsumption as a biologically-inspired "bottom-up" control architecture where feedback loops are layered in terms of priority, so that signals relevant for basic survival pre-empt signals relevant for higher functions such as exploration and mapmaking. This architecture was tested in a number of applications such as "Genghis," a six legged robot which was able to walk based on local independent sensor information from each leg. Although our theoretical model is not layered, it may provide insight into experimental successes with subsumption architectures.

Collaborative control is related to the very active research topic of Cooperative (behavior-based) robots, where groups of *autonomous* robots interact to solve an objective [2]. Recent results are reported in [8, 23, 20, 5]. Collaborative Control is also related to work in online collaborative games such as *Quake (Capture the Flag)*, where users remotely control individual avatars. In our model of collaborative control, the focus is on group control of a *single* shared resource in contrast to groups of resources.

Outside of robotics, the notion of collaborative control is relevant to a very broad range of collaborative human activities including economic markets, pricing behavior, voting, traffic flows, etc. Excellent overviews of the broader context can be found in [15, 22]. There is also a substantial body of research on Distributed Artificial Intelligence (DAI) and Multi-Agent Systems (MAS), emphasizing multiple actors, multiple contexts, multiple representations, resource limitations, compatibility-based problems and robustness [11].

3 Formal Model

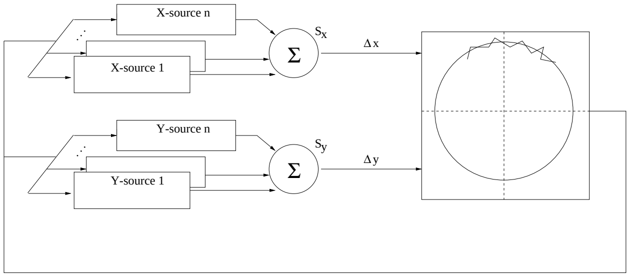

Figure 4: *Collaborative control of robot motion: two ensembles of sources share control of the path.*

We study the collaborative control system illustrated in Figure 4. On the right is the robot motion display showing the desired reference path and the path generated by two ensembles of sources. Each source is modeled with a finite automaton that takes as input the position of the robot and generates one of three outputs: +1, -1, and 0. The ensemble is divided into two equal sized sets, S$_x$ and S$_y$, half controlling horizontal position and half controlling the vertical.

For example, an output of +1 from a member of S$_x$ corresponds to a command to move the robot to the right, and an output of -1 from a member of S$_y$ corresponds to a command to move the robot downward.

We can define a "well-behaved" source based on a circular interpolation algorithm that computes the optimal motion increment from the current cursor position. Consider a source voting on horizontal (Δx) motion, given the current position of the cursor at time step t: (x(t), y(t)). To follow a circular path of radius r, the output of the well-behaved horizontal partcipant is:

$$\mathrm{A}(t+1) = (-1)^b \mathrm{sgn}(\mathrm{x}^2(t) + \mathrm{y}^2(t) - r^2)$$

$$b = \mathrm{inv}(\mathrm{Q}((\mathrm{x}(t), \mathrm{y}(t)) - 1) \& 1)$$

Where sgn is the signum function mapping to 0, -1, +1 and Q is a quadrant function mapping to 1, 2, 3, 4. The inv and & are bitwise operators corresponding to inverse and AND. A similar function describes a well-behaved source in S$_y$. For details see [13].

There are $2k$ members of the ensemble, k in each half. During each time step t, each source computes an output.

$$\Delta_x(t) = \frac{1}{k} \sum \mathrm{A}(t)$$

The robot's horizontal motion increment is based on the ensemble's mean vote, and similarly for the vertical increment.

We can define the following invariant:

$$\cos^{-1}\left(\frac{\mathrm{A}(t) \cdot \mathrm{A}(t+1)}{\|\mathrm{A}(t)\mathrm{A}(t+1)\|}\right) > 0.$$

Each cursor increment always makes counter-clockwise movement around the circle. Initially, the cursor begins at the rightmost tangent point of the circle, the state where Q(x(t),y(t)) = 1 and sgn($x^2(t) + y^2(t) - r^2$) = 0.

No group will be comprised purely of well-behaved sources. Real systems are heterogeneous and include sources that may break down, have sluggish responses, or compute erroneous commands. We can specify formal models to reflect such error modes treating them as malfunctioning sources:

1. Silent: $\mathrm{B}(t) = 0$
2. Inverted: $\mathrm{C}(t) = -\mathrm{A}(t)$

Note that all finite automata considered thusfar are deterministic. We consider non-deterministic finite automata (NFA) in the Discussion section.

4 Performance Metric

To measure collaborative performance, we compute the error in area between the reference path and the generated path as a function of the total area of the reference circle. At any given segment of the generated path, the error is the difference in area between the triangle described by the two linear endpoints p_i, p_{i-1} and the origin, and the sector parameterized by the change in angle $\Delta\theta$ corresponding to the two linear endpoints and r the radius of the circle. Hence, the total error is:

$$\mathrm{E}(t) = \frac{1}{2} \sum_{i=1}^{t} \left| p_{i-1}^x p_i^y - p_i^x p_{i-1}^y - r^2 \Delta\theta \right| \qquad (1)$$

In the case where the line intersects the circle arc, we subdivide the line and recompute the area as the sum of its parts. We define Performance, P = 1 - E$_n$, where E$_n$ is E normalized to the area of the circle. Performance is 100% when the two trajectories match perfectly, and set to 0% when the generated path fails to converge.

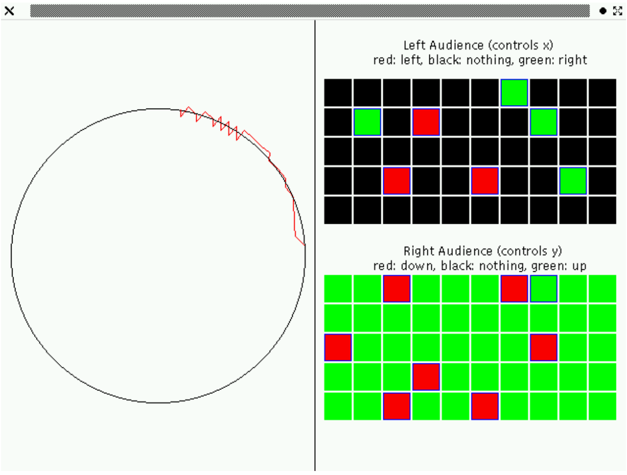
Left Audience (controls x)
red: left, black: nothing, green: right
Right Audience (controls y)
red: down, black: nothing, green: up

Figure 5: *Snapshot of the simulator. The left window displays the progress of the robot path. The right window displays the current output of each source.*

5 Implementation

A snapshot of the simulator is shown in Figure 5. It is written in Java JDK 1.1.7 using the Abstract Window Toolkit. We chose Java because of its ease in prototyping the simulation software and in displaying the results using the Java Abstract Window Toolkit (AWT).

The simulator is modularized into separate components according to functionality. On an abstract level, these main components consist of the environment, polling thread and sources. The evironment displays represents the current position of the point robot and the reference path (in this case a circle). The sources, modeled with a few lines of java code, outputs a value according to the current position of the robot, irrespective of other sources. A polling thread averages commands from all sources. and iterates.

5.1 Results

We consider an ensemble of 100 sources, 50 controlling horizontal position and 50 controlling vertical. The reference path is a circle with radius 20 units. When all sources are well-behaved, system performance, as defined in Section 4, is 63.03%.

1) In the first experiment, we consider malfunctioning sources that go silent; they cease operating, so their vote is a constant 0. Figure 6 plots performance as a function of the percentage of silent sources in the ensemble. We expected a gradual decrease in performance. Instead, we found that performance *increases* steadily as sources go silent! Performance improves until only one source remains (99% have gone silent). The reason: since motion increments are based on the average output of the ensemble, introducing silent sources reduces the size of the motion increment, which causes motion to more accurately track the reference path.

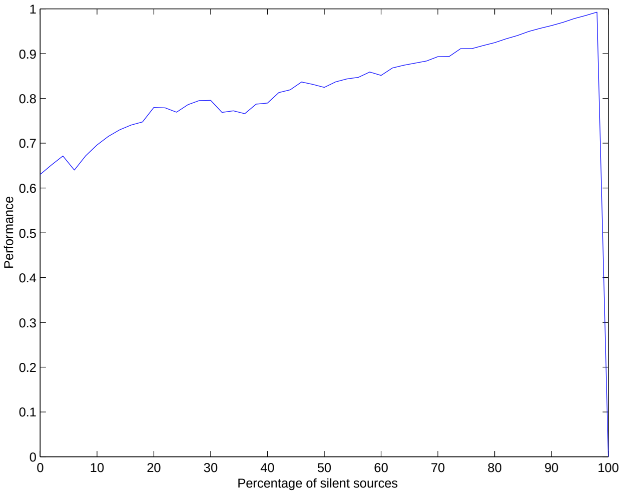

Figure 6: *Performance as sources go silent (send only a 0 signal). Horizontal axis is percentage of sources that have gone silent. Performance* **increases** *as sources go silent until all but one remains.*

2) In the second experiment, we consider malfunctioning sources that invert their outputs: if the well-behaved output is 1 the inverted source outputs -1 and vice versa. Figure 7 plots performance as a function of the percentage of inverted sources in the ensemble. Here, we expected performance to deteriorate rapidly as inverted sources are introduced. But performance improves until about half the ensemble is inverted. A logical explanation exists: each inverted source cancels out the vote of one well-behaved source; the net effect is equivalent to having two sources go silent. Hence the slope of the performance improvement for the inverted sources is about twice as steep as for silent sources.

This initial model confirms experimental reports showing that collaborative control is surprisingly robust in practice. In this system, diversity *improves* performance.

6 Proving Convergence

Simulation shows that performance increases as sources go silent. We can formalize this result as follows. Let k be the number of linear segments in the path generated by the ensemble. When all segments are of equal length, k is approximately equal to $2\pi r/L$, where L is the length of each segment. Because of our linear aggregation model over S$_x$ and S$_y$, L can be defined as:

$$L = \sqrt{c_0 \Delta_x(t)^2 + c_1 \Delta_y(t)^2} \qquad (2)$$

where c_0 and c_1 are constants and $\Delta_x(t)$ and $\Delta_y(t)$ are the aggregate votes from S$_x$ and S$_y$ respectively.

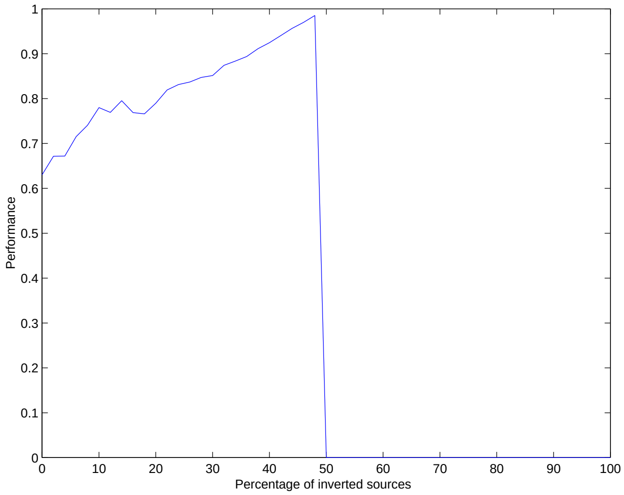

Figure 7: *Performance as inverted sources are introduced into the ensemble. Horizontal axis is percentage of sources that are inverted. Again, performance* **increases** *as inverted sources are introduced until half the sources are operating in opposition to the well-behaved output.*

Consider the aggregate vote $\Delta_x(t)$, for an ensemble composed of m well-behaved and n silent sources:

$$\Delta_x(t) = \frac{\sum_{i=1}^{m} \mathrm{A}_i(t) + \sum_{i=1}^{n} \mathrm{B}_i(t)}{m+n}$$

Recall from Section 3 that $\mathrm{A}_i(t)$ and $\mathrm{B}_i(t)$ correspond to the outputs from well-behaved and silent sources respectively. Since $\mathrm{B}(t) = 0$, $\Delta_x(t)$ reduces to:

$$\Delta_x(t) = \frac{\sum_{i=1}^{m} \mathrm{A}(t)}{m+n}$$

As more well-behaved sources go silent, $\Delta_x(t)$ approaches 0, and similarly for $\Delta_y(t)$. This causes the segment length, L, to approach 0. Consequently, the area of the triangle described by the two endpoints and the origin also goes to 0. Hence, the error $\mathrm{E}(t)$ as defined in Equation 1 approaches 0 and system Performance approaches 1. A similar proof can be constructed for inverted sources.

7 Discussion and Future Work

In this paper we have developed a formal model, treating sources as deterministic finite automata. A collaborative ensemble of sources generates a single stream of incremental steps to control the motion of a point robot moving in the plane. We consider a set of 100 sources, half controlling horizontal motion and half controlling vertical motion. We measure system performance with an ensemble of well-behaved sources and then introduce malfunctioning sources. We find that performance initially *improves* in the presence of malfunctioning sources and is robust even when a sizeable fraction of sources go silent or generate inverted signals. Diversity of sources *improves* performance in such cases.

Collaboration is a crucial ingredient for human education and teamwork. It may be possible for groups of Internet users not only to view but to *participate* in remote experiences [14]. Rather than competing for resources, control of a remote mobile robot might be shared among large groups of simultaneous sources. Our formal model may suggest how humans might share control over the Internet, where time delays can be significant and vary in length. We might model such sources as follows:

1. Random: $\mathrm{D}(t) = \mathrm{sgn}(\mathrm{rand}(-1, 1))$
2. Time Delayed: $\mathrm{F_C}(t) = \mathrm{A}(t - \mathrm{c})$

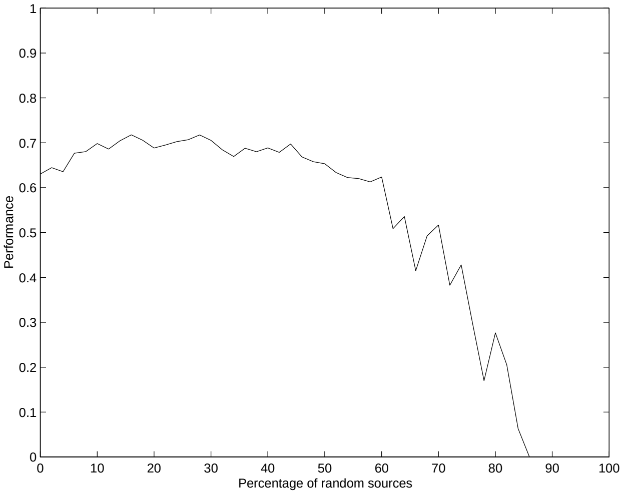

Figure 8: *Performance as random sources are introduced, based on 5000 trials (50 trials per mixture). When more than half the sources are random, the path can fail to converge (Performance = 0) .*

In a preliminary experiment, we introduced nondeterministic sources that generate random output. Figure 8 shows average performance over 5 random trials. Here, performance also increases but deteriorates badly when half the sources are random. Understanding why this occurs will be a priority for future research.

Mark Moll suggested a nondeterministic source that is superior to the well-behaved sources considered earlier. Each source performs a weighted coin flip where the probabilities are proportional to the current path error. If the normalized error $e \in [0, 1]$ then for any source output x_i, $\mathrm{P}(x_i = \mathrm{B}(t)) = 1 - e$ and $\mathrm{P}(x_i = \mathrm{A}(t)) = e$. It follows that the expected value of the ensemble will be the expected value of a single source, e, which will yield overall performance that converges to 100%. A derandomized deterministic version can be constructed where each source $i, (i = 1, ..., n)$, gets assigned an active voting threshold of i/n. Each source will then vote $A(t)$ when the error exceeds its threshold and $B(t)$ otherwise.

We will define general performance metrics for these models by generalizing the path error metric. We will also consider time-to-completion metrics. How should these metrics scale with the number of sources?

We will also study alternative consensus algorithms that are computationally fast and robust to time delays, noise, and variations in source response. Alternative aggregation methods such as winner-take-all and Kalman filters [24] may be superior in terms of noise-rejection or robustness to time delays. For example it may be possible to monitor correlations between sources to cluster and classify sources. Filtering based on such classification may reduce disturbances.

Acknowledgments

We thank John Canny, Mark Moll, Norm Johnson, Loren Carpenter, Eric Paulos, David Pescovitz, Judith Donath, Roger Dannenberg, Rod Brooks, Ron Arkin, George Bekey, and Paul Jacobs for helpful comments on this work.

References

[1] H. Arai, T. Takubo, Y. Hayashibara, and K Tanie. Human-robot cooperative manipulation using a virtual nonholonomic constraint. In *IEEE International Conference on Robotics and Automation*, April 2000.

[2] Ronald C. Arkin. Cooperation without communication: Multiagent schema-based robot navigation. *Journal of Robotic Systems*, 9(3), 1992.

[3] D. M. Blough and G. F.Sullivan. A comparison of voting strategies for fault-tolerant distributed systems. In *9th Symposium on Reliable Distributed Systems*, 1990.

[4] Rod Brooks. A layered intelligent control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2, 1986.

[5] Z. Butler, A. Rizzi, and R. Hollis. Cooperative coverage of rectilinear environments. In *IEEE International Conference on Robotics and Automation*, April 2000.

[6] D. J. Cannon. *Point-And-Direct Telerobotics: Object Level Strategic Supervisory Control in Unstructured Interactive Human-Machine System Environments*. PhD thesis, Stanford Mechanical Engineering, June 1992.

[7] N. Chong, T. Kotoku, K. Ohba, K. Komoriya, N. Matsuhira, and K. Tanie. Remote coordinated controls in multiple telerobot cooperation. In *IEEE International Conference on Robotics and Automation*, April 2000.

[8] B. Donald, L. Gariepy, and D. Rus. Distributed manipulation of multiple objects using ropes. In *IEEE International Conference on Robotics and Automation*, April 2000.

[9] R. Fisher, P. Vanouse, R. Dannenberg, and J. Christensen. Audience interactivity: A case study in three perspectives. 1996.

[10] T. Fong, C. Thorpe, and C. Baur. Collaborative control: a robot-centric model for vehicle teleoperation. In *Agents with Adjustable Autonomy. Papers from the 1999 AAAI Symposium*, March 1999.

[11] Les Gasser. Multi-agent systems infrastructure definitions, needs, and prospects. *Proceedings of the Workshop on Scalable MAS Infrastructure, Barcelona Spain*, 2000.

[12] Ken Goldberg, Steve Bui, Billy Chen, Bobak Farzin, Jacob Heitler, Derek Poon, Rory Solomon, and Gordon Smith. Collaborative teleoperation on the internet. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2000.

[13] Ken Goldberg and M. Goldberg. Xy interpolation algorithms. *Robotics Age*, August, 1984.

[14] Ken Goldberg and Roland Siegwart, editors. *Beyond Webcams: An Introduction to Internet Telerobotics*. MIT Press, 2001.

[15] N. L. Johnson, S. Rasmussen, and M. Kantor. New frontiers in collective problem solving. Technical Report NM LA-UR-98-1150, May 1998.

[16] Y W Leung. Maximum likelhood voting for fault-tolerant software with finite output space. *IEEE Transactions on Reliability*, 44(3), 1995.

[17] K. Lynch and C. Liu. Designing motion guides for ergonomic collaborative manipulation. In *IEEE International Conference on Robotics and Automation*, April 2000.

[18] M. McDonald, D. Small, C. Graves, and D. Cannon. Virtual collaborative control to improve intelligent robotic system efficiency and quality. In *IEEE International Conference on Robotics and Automation*, April 1997.

[19] Paolo Pirjanian. *Multiple Objective Action Selection and Behavior Fusion Using Voting*. PhD thesis, Aalbord University, 1998.

[20] Paolo Pirjanian and Maja J Mataric. Multi-robot target acquisition using multiple objective behavior coordination. In *IEEE International Conference on Robotics and Automation*, April 2000.

[21] Rachel and Lauren Carpenter. **http://www.cinematrix.com** .

[22] S. Rasmussen and N.L. Johnson. Self-organization in and around the internet. In *Proceedings of 6th Santa Fe Chaos in Manufacturing Conference*, April 1998.

[23] I. Rekleitis, G. Dudek, and E. Milios. Multi-robot collaboration for robust exploration. In *IEEE International Conference on Robotics and Automation*, April 2000.

[24] S. Roumeliotis and G. Bekey. Collective localization: a distributed kalman filter approach to localization of groups of mobile robots. In *IEEE International Conference on Robotics and Automation*, April 2000.