USER MODELLING FOR PRINCIPLED SLIDING AUTONOMY IN HUMAN-ROBOT TEAMS

Brennan Sellner, Reid Simmons, Sanjiv Singh
Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
*United States of America**
bsellner@andrew.cmu.edu, reids@cs.cmu.edu, ssingh@ri.cmu.edu

Abstract

The complexity of heterogeneous robotic teams and the domains in which they are deployed is fast outstripping the ability of autonomous control software to handle the myriad failure modes inherent in such systems. As a result, remote human operators are being brought into the teams as equal members via sliding autonomy to increase the robustness and effectiveness of such teams. A principled approach to deciding when to request help from the human will benefit such systems by allowing them to efficiently make use of the human partner. We have developed a cost-benefit analysis framework and models of both autonomous system and user in order to enable such principled decisions. In addition, we have conducted user experiments to determine the proper form for the learning curve component of the human's model. The resulting automated analysis is able to predict the performance of both the autonomous system and the human in order to assign responsibility for tasks to one or the other.

Keywords: Mixed Initiative, User Modelling, Sliding Autonomy, Multiagent, Cooperation

1. Introduction

As complex robotic systems are deployed into ever more intricate and real-world domains, the demand for system abilities is growing quickly.

*This work is partially supported by NASA grant NNA04CK90A.

Since many tasks cannot be easily accomplished by a single machine, much research has turned towards utilizing heterogeneous robotic teams. While this approach multiplies the theoretical capabilities of the deployed hardware, the actual abilities of a team often are constrained by its control software. In complex tasks, such as those required by automated assembly domains, it is nearly impossible for the system designer to anticipate every possible system failure and provide a method for recovery. While automated systems excel at rapid repetition of precise tasks, they are weak when dealing with such unexpected failures. As a result, research is now moving towards including a human in such teams, leveraging the accuracy and strength of robotic teams and the flexibility of the human mind to create a whole greater than the sum of its parts.

A great difficulty in creating these sliding autonomy systems is enabling smooth and efficient transitions between modes of autonomy - ideally both the human and the autonomous system should be able to initiate such transitions as they see fit. If the system is to do so, it needs some method for making decisions about when and how to involve the human in its task. The approach we have taken is to form models of the capabilities of both the autonomous system and the human, in order to provide a principled basis for the system to perform cost-benefit analysis. The autonomous system does not learn to improve its task performance, resulting in a model based on a static distribution derived from observed data. The human model is similar, but incorporates an explicit model of the human's learning curve, allowing the system to predict future performance of a human still learning a particular task. We have experimentally determined that a logarithmic function provides a good fit to our subjects' actual learning curves, with the model providing useful predictions during the learning period. Coupled with a cost-benefit analysis framework, these models allow the system to estimate the overall expected cost of transferring control to the human at various points during the task, enabling it to proactively involve the human when the human will provide the team with significant assistance.

2. Related Work

Our Syndicate architecture (Sellner et al., 2005) (Simmons et al., 2002) (Goldberg et al., 2003) provides a flexible, tiered, multi-agent architecture which we have extended to support sliding autonomy. Syndicate differs from most other multi-robot architectures by allowing close coordination without the need for a central planner. Our user modelling implementation continues this decentralization by allowing each agent to individually form models for itself and the human performing tasks using that agent's hardware.

A number of other sliding autonomy systems exist, of greater or lesser similarity to our work. (Fong et al., 2003) enable the robot to ask the operator for help with localization and to clarify sensor readings, while the operator can query the robot for information. This framework uses the human as an information source, rather than a true partner, and assumes the robot's control software is capable of performing all tasks when provided with complete state information. Our approach allows the operator to be a partner in the completion of the scenario, rather than simply a source of information. An architecture for sliding autonomy as applied to a daily scheduler has been proposed by (Scerri and Pynadath, 2002). The autonomous system is responsible for resolving timing conflicts among team members, who are able to adjust the system's autonomy by indicating intent or willingness to perform tasks. Using similar hardware to ours, (Kortenkamp et al., 1999) have developed and tested a software architecture that allows for sliding autonomous control of a robotic manipulator. While these projects all involve the human in the task, they do not explicitly reason about when to request help.

Similar to our modelling approach, (Fleming and Cohen, 2001) perform cost-benefit calculations to determine whether an agent should ask the user for information that may allow it to generate better plans. Although the basic cost-benefit concept is the same, our user models differ significantly. They represent the user by a series of ad-hoc probabilities (such as the probability that the user will have the requisite knowledge to answer a question), expected utilities, and costs. Their work does not consider the problem of user model acquisition, which is clearly far from trivial. In addition, their agent queries the user only when it believes that it needs help and that the user can provide the requisite information. There is no concept of ceding control to the user merely because the user is better at some element of the task; instead, the user is again treated as an information source, rather than as a partner.

Our sliding autonomy implementation allows any component of our multi-layered system to be switched between autonomous and manual (tele-operated) modes. The fine granularity of control over the team's autonomy level afforded by this approach allows many combinations of human intuition and robotic calculation, rather than limiting the human to the role of oracle. This switching may be performed in three ways: (1) pre-scripted, such as tasks which the autonomous system had not been programmed to perform and must be completed by the operator, (2) human-initiated changes in autonomy resulting from the operator deciding he wants to take control, and (3) system-initiated autonomy changes, which occur when the system's analysis indicates the benefits of requesting help would outweigh the costs. This allows a synergy of human flexibility and robotic accuracy which yields a team with greater efficiency and reliability than either a purely autonomous or purely tele-operated approach. See (Singh et al., 2004) for a discussion of our implementation of sliding autonomy and related experimental results.

3. The Task

For our work on architectures, sliding autonomy, and user modelling, we developed several assembly scenarios that require close coordination between disparate agents. The scenario discussed here requires the team to assemble a square from four beams and four planarly compliant nodes (Figure 1d). The nodes are free to move about in the plane of the workspace, in a weak parallel to orbital assembly. When a beam is inserted into a node, enough force is required to cause an unconstrained node to roll away, rather than the beam's latches engaging the node. In order to provide a countervailing force, the team must brace each node while inserting every beam. To further complicate matters, neither of our manipulator agents possess any extrinsic sensors.

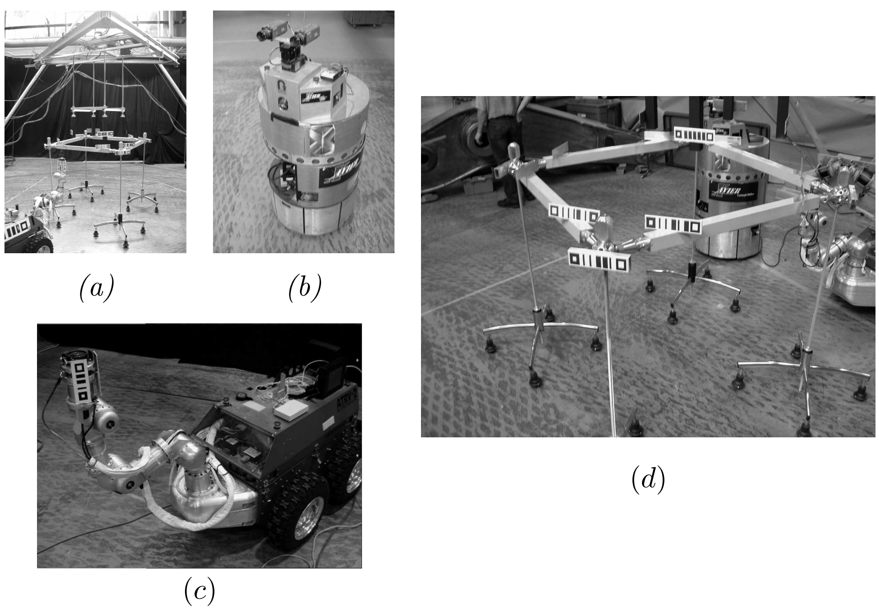

Figure 1. **(a)** The Robocrane. The vertical sockets are used to grasp the nodes from above. **(b)** Xavier, the roving eye of our work crew. **(c)** The mobile manipulator is composed of Bullwinkle (the differential-drive base) and Whiplash (the 5 degree-of-freedom anthropomorphic arm). **(d)** A closeup of the completed structure.

Thus, the scenario can be naturally split into three duties: docking, bracing, and sensing. Our mobile manipulator (Figure 1c) is responsible for docking the beams to the nodes with its 5-DOF anthropomorphic arm. The crane (Figure 1a) handles bracing, while the roving eye (Figure 1b) is responsible for providing information to the other agents about the relative positions of objects in the workspace. Each of these three agents independently performs cost-benefit analysis to determine whether it should ask for human assistance during the scenario.

The scenario consists of four repetitions of the following:

1 Grasp beam with mobile manipulator's arm.

2 Acquire beam and node with roving eye's sensors.

3 Position and brace first node with crane.

4 Insert one end of beam into first node by visually servoing mobile manipulator's arm.

5 Reposition roving eye to second end of beam and acquire beam and second node.

6 Release crane's grasp on first node, grasp second node, position node, and brace it.

7 Insert second end of the beam into node.

8 Release mobile manipulator's grasp on beam.

9 Move mobile manipulator to the square's next side.

The team is *able* to accomplish the entire task autonomously, except for step 1, in which a human places the beam in the mobile manipulator's grasp. However, the human also may become involved in any of the other steps. If during step 3 or 6 the crane becomes stuck, the operator will need to intervene, since the current system cannot detect this failure. In step 2 or 5, if the roving eye is in a position such than an object of interest is obscured, the autonomous system will be unable to acquire, and will request assistance from the user. Docking one end of a beam to a node (steps 4 and 7) is a difficult task; the system will often fail one or more times before succeeding or dropping the beam. This is another opportunity to involve the operator, since an initial failure of the autonomous system is a good predictor of future failure; this occurence often results in a request for help after one or two failures.

Although the scenario can be accomplished autonomously, there are many opportunities for the system to request help from the human operator to increase its robustness and efficiency.

4. Using the User

The original sliding autonomy system we created was effective, but somewhat lacking in initiative. The system requested help only when told to do so ahead of time or when the team detected a failure from which it could not recover. This is clearly suboptimal: in the ideal case the autonomous system should request help not only when it *needs* assistance, but also when assistance would be beneficial to the reliable and efficient completion of the scenario. For instance, if the system has a failure recovery procedure for a particular error, but the procedure proves ineffective, it could ask the user for help after determining that further attempts are likely to be useless, rather than repeatedly attempting to blindly apply its recovery procedure. The node-beam docking action (steps 4 and 7 above) is an excellent example of this. In addition, there are occasionally tasks which the human is often more efficient at performing via tele-operation than the system, due to her superior ability to make inferences from noisy observations. Such tasks within our scenario include manuvering in cluttered environments and visually searching for partially obscured objects.

If the system is to further involve the human in the scenario, it must have some method of reasoning about when to do so. The approach that we have taken is to perform cost-benefit analysis at various decision points during the scenario, using empirically derived models of the individual users and the autonomous system to inform the analysis. By maintaining such individual models, the system's requests for help may depend on the abilities and state of the individual operator, yielding a team that adapts not only to the current state of the environment but also to the current state of its members. Such a principled approach allows the autonomous system to leverage the skills of the human operator to increase both the team's robustness and its efficiency.

4.1 Cost-Benefit Analysis

Cost-Benefit Analysis simply consists of estimating the costs and benefits associated with various courses of action in order to choose the most beneficial action to perform. In our domain, such decisions are binary: the system must decide whether to request operator assistance for a particular task. Obviously, the option with the greatest $benefit - cost$ value will be chosen. Given methods for estimating the relevant variables, this provides a framework for making principled decisions, rather than implementing arbitrary policies. Within our robotic assembly scenarios, one form of this equation is:

$$\begin{aligned} cost: &\quad price(h)E(t_h) + price(r_t)E(t_h) + price(rep)P(fcat_h) \\ benefit: &\quad price(r_a)E(t_r) + price(rep)P(fcat_r) \end{aligned} \tag{1}$$

where:

$E(t_h)$:	Expected time for human to complete task
$E(t_r)$:	Expected time for autonomous system to complete task
$P(fcat_h)$:	Probability of catastrophic failure while under human control
$P(fcat_r)$:	Probability of catastrophic failure while under autonomous control
$price(rep)$:	Average monetary cost of repairing a catastrophic failure
$price(h)$:	Monetary cost of operator per unit time
$price(r_t)$:	Monetary operating cost of system per unit time while teleoperated
$price(r_a)$:	Monetary operating cost of system per unit time while under autonomous control

The costs are those incurred during the human's teleoperation of the system, while the benefits consist of the cost savings associated with not running the system under autonomous control. In a real-world application, the price functions would be informed by factors such as the amortized cost of the hardware, upkeep, the salary of the human operator, and what other duties he is responsible for (since assisting the system will monopolize his time). These functions act as gains, with the relative values of $price(h)$, $price(r_t)$, and $price(r_a)$ encouraging or dissuading the system from asking for help, and $price(rep)$ adjusting how averse the system is to risk . The probability of catastrophic failure is estimated from experimental data. Note that catastrophic failure is distinct from the failure to perform a task in that the former results in damage to the robots which must be repaired while the latter merely results in the non-accomplishment of a particular task.

The most difficult element of these equations to estimate is the expected time to complete a given task for both the autonomous system and the human ($E(t_r)$ and $E(t_h)$, respectively), especially if the operator is a novice. We have built a user model to estimate these expected times based on previous experience, as well as a number of other factors.

4.2 User Model

A user model can consist of any collection of rules or set of assumptions that predicts the value of interest or otherwise allows the system to decide when to involve the user. In fact, our initial sliding autonomy system incorporated an extremely basic user model by requesting help only when an unrecoverable failure occurred. This simple approach allowed the user to slightly increase the system's robustness, but not its efficiency. A more refined model could include fixed thresholds for when help should be requested. Such a model could assert that if the system has failed to recover from an error twice it should ask for help. Again, this allows the user to contribute to the system's robustness, but the human is likely not being utilized in an efficient manner. In order to create an efficient overall system and take into account external constraints on the human, a much more detailed and data-driven model is required.

4.2.1 The Ideal Model. We have developed predictive models for both the autonomous system and the human operator; we address the system's model first. Since our current autonomous system does not learn, we may treat each attempt at a task as a sample from a static distribution. The set of all observed performances is in all likelihood multimodal. However, by segmenting the observed attempts based on the outcome of each attempt and the number of times the system had previously failed to perform the task during the current trial, we may easily form a set of unimodal (or nearly unimodal) distributions. We may then estimate $E(t_r)$ directly from these distributions:

$$E(t_r|F_r = i) = \begin{array}{c} P(S_r|F_r = i)E(t_r|S_r, F_r = i) \\ +P(\neg S_r|F_r = i) \begin{pmatrix} E(t_r|\neg S_r, F_r = i) \\ +E(t_r|F_r = i+1) \end{pmatrix} \end{array} \tag{2}$$

$$E(t_r|F_r = h) = E(t_h|F_h = 0, R_h = j, F_r = d_r + 3) \tag{3}$$

$$E(t_r|F_r = d_r + 1) = E(t_r|F_r = d_r) \tag{4}$$

$$E(t_r|F_r = d_r + 3) = 0 \tag{5}$$

$$E(t_r) = \min_{h=max(f,1)...d_r+2} E(t_r|F_r = f) \tag{6}$$

where:

$E(t_r\|F_r = i)$:	Expected time to complete the task if the system performs the next attempt, given i preceding failures.
$P(S\|F = i)$:	Probability of completing the task, given i preceding failures.
$E(t\|S, F = i)$:	Expected value of the distribution formed by all data points in which the task was completed with i preceding failures.
F:	Number of preceding failures.
h:	Number of failures after which control will pass to the operator.
R_h:	Number of previously observed human runs.
d:	Max number of preceding failures for which data exists.
j:	Current number of previously observed human runs.
f:	Current number of preceding failures.

As can be seen from Equation 2, the expected time to complete the task if the autonomous system performs the next attempt is a recursive sum, representing the successive attempts made after a failure to complete the task. Equation 4 permits the autonomous system to make an attempt with one more preceding failure than has been previously observed. As we can see from Equation 6, the final value for $E(t_r)$ is chosen by determining the proper point in the future to hand control to the human ($h >= 1$ because $E(t_r)$ represents the assignment of the next attempt to the autonomous system). Equation 5 prevents infinite mutual recursion, since the human's user model includes passing control to the autonomous system (see Equation 9).

We introduce two new elements in the human's model: the learning curve and the possibility of the autonomous system requesting control. If the operator is inexperienced, it is inaccurate to model her performance as a sample from a static distribution. Rather, she is still learning, and it is more appropriate to model $E(t_h)$ by predicting the next point on the learning curve, rather than simply taking the expected value of a distribution of past data. This learning curve (Equation 7) is a logarithmic curve fitted to the available data. We have conducted a series of experiments, discussed below, to determine a reasonable model of $L(x)$ and how best to use it as a predictor of the human's performance in the next trial. Equation 8 represents the system's belief that the human has failed if they are taking too long to complete a task. This is necessary to detect operator failure, since the human operator rarely, if ever, voluntarily gives up. Additional factors may play a role in $E(t_h)$, such as current workload and fatigue. However, they would likely play a relatively straightforward additive or multiplicative role in Equation 7, and are thus neglected for now.

$$N(t) = L(x_{t+1}|x_{1\ldots t}) \tag{7}$$

$$M(s,i) = P(s|F_h = i) - P(t_h > cN(R_h)|s, F_h = i) \tag{8}$$

$$\begin{aligned} &E(t_h|F_h = i, R_h = j, F_r = k) = \\ &\quad M(S_h, F_h)N(R_h) \\ &\quad +M(\neg S_h, F_h)\left(N(R_h) + E(t_h|F_h = i+1, R_h = j+1, F_r = k)\right) \\ &\quad +P(t_h > cN(R_h))E(t_r|F_r = k) \end{aligned} \tag{9}$$

where:

$N(t)$:	Predicted time to complete the task based on a learning curve L fitted to all prior observations.
$L(x_{t+1}\|x_{1\ldots t})$:	The value of a fitted learning curve for trial $t+1$, given t prior observations.
$M(s,i)$:	The probability of $s \in \{S, \neg S\}$ given i preceding failures, less the probability that the autonomous system will request control, given s and i.
c:	A constant which determines the time when the system believes the human has failed and the time when it will request control.
$P(t_h > cN(R_h)$ $\|s, F_h = i)$	The probability that the human will take more than c times the expected time to complete the task.
$E\left(t_h\| \begin{array}{r} F_h = i, \\ R_h = j, \\ F_r = k \end{array}\right)$	Expected time to complete the task if the human performs the i'th attempt, with j historical attempts, and k preceding failures by the autonomous system.

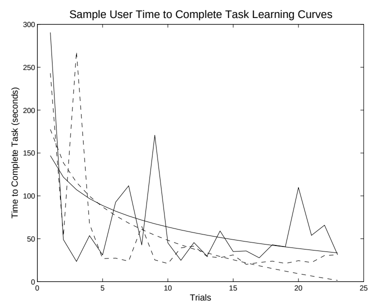

Figure 2. The raw data and fitted logarithmic learning curves for two sample subjects - Subject A's raw data and fitted curve are plotted as solid lines, while Subject B's are dashed lines.

4.2.2 The Implemented Model. When implementing any method, tradeoffs must be made between the fidelity of the model and the difficulty involved in estimating the variables involved. For our initial implementation, we set $P(fcat_h) = P(fcat_r) = 0$, $price(h) = price(r_a)$, and $price(r_t) = 0$, collapsing Equation 1 to a straightforward comparison of $E(t_r)$ and $E(t_h)$. The model of the autonomous system was implemented as described in Equations 2 - 6. However, we chose to simplify the human model by disregarding all factors affecting the prediction of the human's performance except previous observations and an estimate of the human's learning curve. We also set $c = \infty$ to prevent the system from requesting control. Since no subject ever failed to complete the task, $P(S_h|F_h = 0) = 1.0)$, resulting in Equation 9 collapsing to $E(t_h) = L(x_{t+1}|x_{1...t})$, or simply the prediction of her time to complete the task based on her learning curve and prior experience. The resulting simplified calculation directly compares the system's and human's estimated time to complete a task in order to assign responsibility.

4.3 Results

In order to build our initial model of the human learning curve $L(x)$ within our domain, we conducted a series of experiments with eight subjects to assess and model their learning curves for direct control of the mobile manipulator's arm. The goal was to develop an understanding of the repeatability a human's time to complete a task; how many trials it would take to eliminate learning effects; and whether a single learning function could be accurately parameterized on a per-user basis, to allow the system to attempt to predict the performance of a user who had not been fully trained.

In order to focus purely on the skill of controlling the arm and minimize confounding variables, the task consisted of docking one end of a beam to a node while directly observing the workspace and controlling the arm via a SpaceMouse (a six degree of freedom input device). Data from a representative two of our eight subjects can be found in Figure 2 — this data contains roughly an average amount of noise. As can be seen, the raw data is quite noisy, with large deviations between successive runs. However, it does consistently trend downwards, and while examining all eight data sets, we discovered that a logarithmic learning curve of the form $L(x) = a * ln(x) + b$, with the parameters a and b fitted to each user's data, yielded a more predictive fit than linear, exponential, or quadratic models. On average, 10 trials worth of data were necessary for the parameters to settle to their final values, but the logarithmic model proved some predictive worth as early as trial 3. Most subjects' performance had plateaued by trial 14.

Taking this into account, we have extended our sliding autonomy system to include the simplified user model described in Section 4.2.2 for making principled decisions about changes to the team's autonomy levels. The model tracks each operator (and the autonomous system) for every task in the scenario. Given the instability of the initial few trials for most subjects, the model merely predicts the average of the past observed trials for $E(t_h)$ until three trials worth of data are accumulated. Between trials three and fourteen, a logarithmic curve is fit to the data and is used to predict $E(t_h)$ on the next instance of the task. After trial fourteen, the average of the past three trials (discarding outliers) is used, since most subjects' performance plateaus by this point, with the occasional outlier. This allows the autonomous system to make appropriate use of even an inexperienced operator.

5. Future Work

A variety of opportunities to expand upon this work exist. Our simplified model needs to be verified in our assembly system and potentially refined to provide satisfactory predictions. The model could also be extended to extrapolate performance on unobserved tasks from performance on different, but related, tasks. Knowledge of upcoming tasks could also be incorporated into the model, allowing the system to make locally inefficient decisions in order to train a novice user to provide future benefits. Similarly, if the human fatigues over the course of a scenario, the system could avoid asking for help when the human only provides marginal benefit, in order to keep her rested for tasks where she is orders of magnitude better.

6. Conclusion

We have formulated a method for making principled decisions about when to involve a remote human operator in a multi-agent assembly task. We conducted initial user experiments, determining that a parameterized logarithmic function provides an adequate fit to users' observed learning curves. Such a function, tuned to each user as data is observed, provides a usable predictive model of their future performance. Combined with our predictive model of autonomous system performance, this simplified model has been implemented within our sliding autonomy system, allowing the system to make principled decisions about when to request assistance from the operator.

Acknowledgments

The authors would like to thank the many individuals who contributed to the DIRA and Trestle projects over the years: Rob Ambrose, David Apfelbaum, Jon Brookshire, Rob Burridge, Brad Hamner, Dave Hershberger, Myung Hwangbo, Simon Mehalek, Metrica/TRACLabs, Josue Ramos, Trey Smith, Pete Staritz, and Paul Tompkins.

References

Fleming, M. and Cohen, R. (2001). A user modeling approach to determining system initiative in mixed-initiative ai systems. In *Proceedings of User Modeling (UM)*.

Fong, T., Thorpe, C., and Baur, C. (2003). Robot, asker of questions. *Robotics and Autonomous systems*, 42.

Goldberg, D., Cicirello, V., Dias, M., Simmons, R., Smith, S., and Stentz, A. (2003). Market-based multi-robot planning in a distributed layered architecture. In *Multi-Robot Systems: From Swarms to Intelligent Automata: Proceedings from the 2003 International Workshop on Multi-Robot Systems*, volume 2, pages 27–38. Kluwer Academic Publishers.

Kortenkamp, D., Burridge, R., Bonasso, P., Schrenkenghoist, D., and Hudson, M. (1999). An intelligent osftware architecture for semi-autonomous robot control. In *Autonomy Control Software Workshop, Autonomous Agents 99*.

Scerri, P. and Pynadath, D. (2002). Towards adjustable autonomy for the real world.

Sellner, B., Simmons, R., and Singh, S. (2005). Syndicate: A decentralized, layered architecture for tightly coordinating heterogeneous teams. In *Submitted to the International Conference on Robotics and Automation (ICRA-05)*.

Simmons, R., Smith, T., Dias, M. B., Goldberg, D., Hershberger, D., Stentz, A., and Zlot, R. (2002). A layered architecture for coordination of mobile robots. In Schultz, A. and Parker, L., editors, *Multi-Robot Systems: From Swarms to Intelligent Automata*. Kluwer.

Singh, S., Brookshire, J., and Simmons, R. (2004). Preliminary results in sliding autonomy for assembly by coordinated teams. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS) 2004*.