RESULTS IN SLIDING AUTONOMY FOR MULTI-ROBOT SPATIAL ASSEMBLY

Frederik W. Heger[1], **Laura M. Hiatt**[2], **Brennan Sellner**[1], **Reid Simmons**[1][2], **and Sanjiv Singh**[1]

[1] *The Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, U.S.A.*
[2] *Computer Science Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, U.S.A.*
fwh@cs.cmu.edu, lahiatt@cs.cmu.edu, bsellner@andrew.cmu.edu, reids@cs.cmu.edu, ssingh@ri.cmu.edu

ABSTRACT

We have developed a software architecture for teams of robots and humans to jointly perform tightly coordinated tasks, such as assembly of structures in orbit or on planetary surfaces. While we envision that robots will autonomously perform such work in the future, the state of the art falls short of the capabilities necessary to handle all possible contingencies. Our architecture provides a principled methodology for human involvement to optimize both task efficiency and robustness by combining robot capability with human intuition. We call such mixed control strategies "sliding autonomy". Robots accomplish as many of the tasks as they can autonomously, and human operators take over control to perform those that cannot be easily automated or to provide help when the robots fail. In this paper, we discuss results from recent experiments that quantify the effect of different levels of autonomy on the system's overall performance. By introducing two modes of sliding autonomy, we are able to achieve the high reliability of a teleoperated system combined with the high efficiency of autonomous operation. The incurred mental demand of the operator is directly proportional to the increase in system efficiency.

Key words: Human Robot Cooperation, Architecture, User Modeling, Multi-Agent Systems.

1. INTRODUCTION

As expectations for robotic systems grow, it becomes increasingly difficult to meet them with the capabilities of a single robot. Instead, using multiple simpler robots to perform tasks that would require a very complex single mechanism is advantageous in many respects: these teams not only bring a much broader spectrum of potential capabilities to a task, but they also may be more robust in the face of errors and uncertainty.

Our motivating scenario is the semi-automated assembly of large structures in orbit or other hostile environments, where human labor is unavailable or prohibitively expensive. While in the future robots will likely be able to perform such work autonomously, currently autonomous operation is beyond the state of the art. In particular, the number of contingencies that must be handled in order to make robots fully autonomous is prohibitively large. On the other hand, teleoperated control of such robots is unlikely to be efficient enough because of the communication delays involved, as well as the large number of human operators required.

Fig. 1. Three robots with very different topology are used to autonomously assemble a square structure of four beams supported by four nodes.

We are interested in how small teams of robots in space and a few humans on Earth could work together to assemble large orbital structures such as kilometers-wide solar power arrays. In order to be efficient overall, the teams of robots should operate autonomously in standard situations, asking humans for help only when problems arise that they cannot resolve by themselves or when human control provides significant benefits. We posit that the resulting blend of robot capability and human intuition, which we call "sliding autonomy", is more robust than the fully autonomous system and more efficient than

Proc. 'ISAIRAS 2005 Conference', Munich, Germany, 5-8 September 2005 (ESA SP-603, September 2005)

either the system or an operator working alone.

To test this hypothesis, we have assembled a heterogeneous team of three robots, consisting of an imprecise heavy-lift agent, a weaker but more precise mobile manipulator, and a dedicated sensing robot that work together to construct the square structure shown in Fig. 1. Coordination of the team is provided by Syndicate, our distributed, layered architecture, which allows each agent to operate at varying levels of temporal and task granularity while seamlessly communicating with its peers at all levels of abstraction. Syndicate is also the foundation for our version of sliding autonomy (also called adjustable autonomy [8] [13]), which provides a meaningful synergy between fully autonomous robot operation and complete teleoperation by remote humans. In the sliding autonomy mode of operation, the autonomous system is given the ability to ask for human assistance. In order to allow the system to make principled decisions about when to make such requests, we have developed models that allow the system to make predictions of future performance of the autonomous system and the human operators based on past data.

In this paper, we report the results of a pilot study to compare the effect of varying human involvement on the completion time and mental operator loading. This work builds on our previous efforts [2], which considered only time taken for a less complex task. In between autonomous operation (no human involvement) and teleoperation (100% human effort) we added system- and mixed-initiative sliding autonomy modes. In the system-initiative case, the human waits for the system to ask for help before performing any task, while in mixed-initiative he can take over control of active subtasks at any time.

Our experiments indicate that the autonomous system is consistently faster, but less reliable, than a purely teleoperated approach. In both sliding autonomy scenarios, the speed of assembly matches that of the autonomous system. We conclude that our approach to sliding autonomy improves overall reliability with minimal loss of efficiency.

2. RELATED WORK

Coordinated assembly performed by teams of mobile robots is of prime interest to the space community. Stroupe et al. use the CAMPOUT architecture to coordinate robots with purely behavior-based strategies to perform very tightly coupled tasks [14]. Two heterogeneous robots carry a beam and position it with respect to an existing structure with sub-centimeter accuracy.

Recently, there has also been strong interest in adding human collaboration with robots to such assembly scenarios. The COBOT project [7] [15] seeks to make manually operated machines more intelligent by providing guidance so that the operator does not have to finesse control. Typically, the human provides the force input, while the system steers the mechanism into the right place. The roles of the human operator and the system are clear and unvarying, and both the human and the system must operate simultaneously. NASA's ASRO project [3] developed a mobile robot to assist a space-suited human by carrying tools, helping to manipulate objects, and providing sensor information. While the robot was physically working alongside the astronaut, it was teleoperated by another person in communication with the astronaut from a remote site. Our system, in contrast, allows the remote user to take control of parts of the assembly task while leaving the remainder active under robotic control. The human and robots cannot directly interact physically, since they are not colocated, unlike [7] and [15].

A system related to our approach to human-robot interaction is described by Fong et al. in which the robot and the user participate in a dialogue [6]. The robot can ask the operator to help with localization or to clarify sensor readings. The operator can also make queries of the robot. This framework assumes that the robot is capable of performing all tasks as long as it has full state information. Another effort has examined the effectiveness of an operator when controlling a robot at different levels of autonomy given increasing inattention to the robot [8].

Scerri has proposed an architecture for sliding autonomy applied to a daily scheduler [11]. The autonomous system attempts to resolve timing conflicts (missed meetings, group discussions, personal conflicts, etc.) among some set of team members. Members are able to adjust the autonomy of the system by indicating their intent to attend gatherings or willingness to perform tasks.

Using a roving eye and a (fixed) manipulator similar to ours, Kortenkamp et al. developed and tested a software infrastructure that allows for sliding autonomous control of a robot manipulator [10]. Their task involved a pick-and-place operation during which sliding autonomy allowed the operator to recover from visual servoing errors, participate in high-level planning, and teleoperate the manipulator to complete tasks beyond its autonomous capabilities. Our work extends these experiments with a more complex assembly task and a finer granularity of sliding autonomy. The term sliding autonomy is interchangeable with adjustable autonomy as presented by Dorais et al. [5], in which the authors provide several examples of how sliding autonomy will be essential for space operations where demands on the operator must be focused and minimized.

3. THE TRESTLE SYSTEM

In order to set the stage for a description of our sliding autonomy approach, we introduce the main components of our system to provide the reader with an understanding of its capabilities and limitations.

3.1. Hardware

The Trestle system consists of three very different robots: a NIST RoboCrane [1], a skid-steer ATRV-2 base with a

Fig. 2. The mobile manipulator (top left), the roving eye (top right), the RoboCrane (bottom left) and the completed structure (bottom right).

5-DOF Metrica arm and a robot based on a RWI B24 with a stereo camera pair on a pan-tilt mount. Together they are capable of assembling a square structure of four beams supported by four nodes (see Fig. 2).

In order to simulate conditions in space, the nodes are supported by casters that roll easily along the floor, providing planar compliance. Each node has two beam docking receptacles. The male ends of the docking connections are attached to either end of the beams (see Fig. 3). Three spring-loaded hooks prevent the beam from sliding out after the docking is complete. Pulling the beam away from a braced (i.e. stationary) node, the robots are strong enough to overcome the springs and free the beam.

Fig. 3. Close-up of a beam being inserted into a node (left) and a node being braced by the crane (right).

The mobile manipulator can hold a beam by using an electromagnet at the end of its arm to connect to a steel plate attached at the center of the beam. The magnet strength is calibrated to allow a reliable grasp of the beam. However, if the beam gets stuck it will slide on the magnet before the resulting torques can cause damage to the arm.

The crane, an inverted Stewart platform, is equipped with four receptacles used to brace the nodes. The cables supporting the crane's end-effector limit its range of motion such that a single receptacle cannot reach each node. The crane braces a node by lowering one of its receptacles onto a piece of aluminum stock attached to the top of the node (see Fig. 3). The square cross-section of the part just fits into the square receptacles, thus preventing the node from rotating independently of the crane once it is braced. The tolerance between the node and receptacle is small enough that exact alignment to within a few degrees is necessary for successful bracing.

By design, the roving eye's cameras are the team's only extrinsic sensors. The roving eye tracks fiducials attached to the other robots and assembly components. Similarly, the mobile manipulator is the only robot capable of handling the beams, while only the crane can brace the nodes from sliding away during docking. Thus, the robots' task easily decomposes into subtasks suited to the capabilities of each agent, and it cannot be completed by any one robot alone.

3.2. Assembly Task

To assemble one side of the square structure, the crane first braces the node, using the roving eye to provide relative pose information between the node and the crane's receptacle. Once the node is braced and in a position that guarantees that the entire structure remains within the crane's workspace during assembly, the mobile manipulator servos the beam based on input from the roving eye until the right end of the beam is securely docked. At that time the roving eye repositions itself near the free end of the beam while the crane releases the first node and braces the second node in the same manner as before. Finally, when the second node is in place, the left end of the beam is docked. When the docking is complete, the mobile manipulator releases the beam and moves to the next side of the square. This sequence repeats four times to assemble the entire square.

Fig. 4 shows a high-level representation of the assembly steps required to dock one beam between two nodes. Each agent has a chain of tasks it has to perform as indicated by the three rows of ovals. However, some tasks require certain steps to be completed prior to their execution. These dependencies are represented by arrows between rows. For example, the roving eye cannot start watching the mobile manipulator (task RE3) until the crane has positioned the node (task C2), and the mobile manipulator cannot dock a beam (task MM4) without being watched by the roving eye (task RE3) and knowing that the node is in the correct place (task C2). The task script in Fig. 4 represents a standalone node-beam-node assembly. When a full square is being assembled, the left end node of one side automatically turns into the right end node of the next side so that the crane does not have to release and re-brace the node in this case.

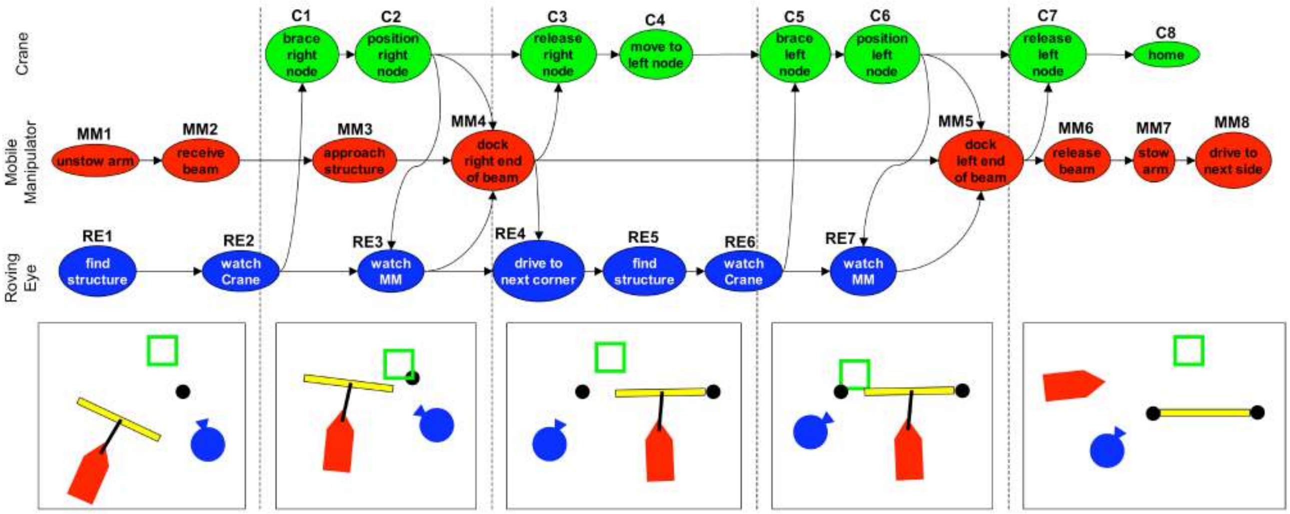

Fig. 4. Task script for and a plan view graphical representation of a single node-beam-node subassembly. The robots are represented by their characteristic shapes (the square represents the crane's end-effector), Arcs between nodes on different agents indicate preconditions for the target node's task.

3.3. Architecture

Our Syndicate architecture is a hybrid of deliberative and reactive approaches supporting multiple heterogeneous agents by distributing processing and control across agents and providing for flexible inter-agent communication. Syndicate's defining feature is its ability to support transparent communication at multiple levels of abstraction. This support for flexible communication and layers of abstraction makes Syndicate an ideal foundation upon which to build our sliding autonomy system.

In principle, the architecture consists of three layers for each agent: a behavioral layer at the bottom responsible for low-level reactive behaviors and interfacing with the hardware, a high-level planning layer at the top and an executive layer in between that maintains state and coordinates the behaviors based on a given plan. In our current implementation, the planning layer is replaced by a static task script. Within each agent, the executive is connected to both the planning and behavioral layers. In addition, corresponding layers on different agents are also connected directly. This connectivity allows agents to communicate information at whichever level of abstraction is most appropriate and can be used to set up distributed control loops across multiple agents. The visual servo operations performed by the crane and the mobile manipulator are prime examples of such a strategy. Relative pose information from the roving eye's behavioral layer is used directly by the mobile manipulator's behavioral layer to to move the beam closer to its goal.

4. SLIDING AUTONOMY

The sliding autonomy spectrum spans the entire range from fully autonomous operation at one end to complete teleoperation at the other. While teleoperation is reliable but slow, the system in autonomous mode performs more efficiently. However, a failure rate of roughly 10% of attempted assemblies with no dominating failure source is an indication that the number of contingencies that would have to be taken into account for reliable autonomous operation is much too large for feasible implementation. To ameliorate this problem, our approach explicitly incorporates the option of assistance from a remote human.

4.1. Modes of Control

At the two extremes of the autonomy spectrum, there are the obvious control modes of fully autonomous operation and complete teleoperation. In *autonomous* mode, the system performs the entire assembly without human intervention. If the task cannot be completed due to an autonomously unrecoverable failure or if a dangerous situation is imminent, the run is aborted and counted as a failure. In *teleoperation* mode, the operator performs all tasks without assistance from the system by commanding each of the robots in turn using a 6-DOF "SpaceMouse" input device [4].

In the interior of the autonomy spectrum, there are two distinct situations where human involvement becomes relevant: either the system is unable to complete a certain subtask, or the system predicts that a human operator is better at performing the current task at hand, based on past data. In addition, there is the case where the operator simply decides to take over control irrespective of the system's need for assistance at the time. We differentiate between *system-initiative* and *mixed-initiative* sliding autonomy modes. The two approaches differ in who makes the decision to switch control. In system-initiative sliding autonomy, the system decides when to give control to the operator. In contrast, when working under mixed-initiative, users also have the ability to take control from the system whenever they see fit. For example, an operator watching the system can decide that they can complete the task faster and take control from the system, while in system-initiative they would have to wait for the system to ask for help.

Robots, while good at performing repetitive, pre-scripted tasks, often fail to detect or recover from unforeseen error conditions. On the other hand, humans are excellent at handling unexpected situations, but have a more limited workload capacity and are a much scarcer resource. In addition, their effectiveness for intervention is limited because by definition, remote operators are not able to interact physically with the environment. Thus, some method of coordinated cooperation is needed in order for the autonomous system and human operators to work together to successfully achieve their common goal.

4.2. Mechanism for Switching Control

At the core of any sliding autonomy system lies the question of exactly how a system's level of autonomy should be adjusted and who should adjust it. Our goal is to balance the flexibility yielded by fine control over the autonomy level against the resulting complexity. Syndicate allows for different modes of operation that "slide" control of subtasks between the autonomous system and the operator.

A task in our system is split into two components: an action part and a monitoring part. Each part can be thought of as a separate task. This partition gives us fine control over what aspects of a task we want to switch. For example, the system may be capable of executing a particular subtask, but its sensing capabilities are insufficient to determine when the task is complete. In that case, the monitoring of the task can be switched to human control while the actual task execution remains autonomous. Similarly, if the system autonomously maneuvers itself into a situation it cannot recover from, the human can be asked to take over the execution of the task while the system continues to watch for task completion.

Tasks may be either statically preassigned to the human or the autonomous system, or the assignment may be made dynamically. We are moving away from static task assignment, preferring to use our user modeling package to dynamically assign tasks. Dynamic task allocation allows the autonomous system to evaluate its expected efficiency against any available human operators, making reasoned decisions about when to adjust its own level of autonomy based on current conditions and the specific human(s) available. These decisions are made both when tasks are initially launched and whenever a failure occurs, allowing the system to attempt the task itself but hand control to the human if it is unable to complete the task. Alternatively, if the operator has shown he is usually better than the autonomous system, the task will be assigned to him immediately.

In addition to the system's decision making process, the human may switch the assignment of tasks (i.e. change the level of autonomy) when working in mixed-initiative mode. This allows the human to take control of any task, as well as relinquish control of a task they currently "own" at any time. As long as the human follows the autonomous system's plan, this approach works well. However, if the human diverts from the plan by undoing previous accomplishments, performing tasks out of order, or performing more tasks than expected, the autonomous system likely will be unable to successfully continue. Open research questions include how best to track the operator's actions, what cooperative actions should be taken by agents still under autonomous control in response to the operator's actions, and how best to continue the mission when control is passed back to the autonomous system at an arbitrary point in the plan.

4.3. User Modeling

Our data shows that the best strategy to optimize robustness is complete teleoperation. However, to increase efficiency, we must combine autonomous and teleoperated modes in an intelligent way. For this to happen, the system must determine when it is advantageous to pass control to the operator. This decision making process is employed in both sliding autonomy scenarios. It is particularly important under system-initiative operation, where it is the only way control can be transferred and thus the only way the autonomous system's performance can be improved.

The system makes decisions about the transfer of control both at the beginning of a task and after any failures. At every such decision point (indicated by a diamond in Fig. 5), the system should (re)assign the task so as to minimize the expected time to complete the task. This is equivalent to evaluating a decision tree, such as the one shown in Fig 5, whose branch points correspond to failed attempts (or the beginning of the task).

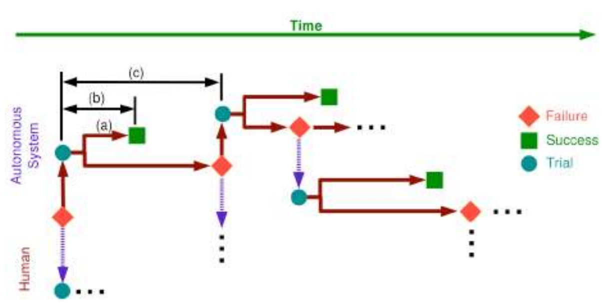

Fig. 5. An example of the decision trees which are evaluated by the user modeling system.

There are three components to this recursive prediction: the probability of success for a given party's attempt, the expected time given success, and the expected time given failure. For the option to give the autonomous system control of the next attempt, these correspond to the probability of branch (a), timespan (b), and timespan (c) in Fig. 5, respectively. To estimate these values, we use prior observations, conditioned on the number of preceding failures by the controlling party that have occurred so far during this particular task. The number of preceding failures is roughly equivalent to the current depth in the decision tree, and is used because a failure is empirically a good predictor of future failures. In addition, we condition our estimated time calculations on the outcome of the attempt, as failures often take significantly longer than successful attempts. Since our models are updated during task execution and are maintained on a per-user basis, the system's decisions will dynamically change in response to the operator's current performance and will depend on the specific operator available.

The expected time of an attempt given its outcome is treated as a sample from a static distribution when considering the autonomous system or an expert operator. This distribution is formed from the subset of all prior observations that match this combination of success and preceding failures. Since it is nearly always unimodal, the expected value of such a sample is merely the mean of the component data points. However, this simple model does not apply in the case of a novice. Since the operator is still learning, it is more appropriate to model the expected time taken by the operator by predicting the next point on his learning curve. We have previously conducted experiments [12] to determine a reasonable model for this curve and how best to use it as a predictor of the operator's performance. According to our data, a logarithmic curve fitted to the available data was the most accurate predictor of future performance out of the potential fitting functions we examined. The fit of this curve is continuously updated as more data is acquired about an operator's performance on a task until it levels off (generally after 15-20 attempts). The system then assumes the operator is an expert and switches to using the static distribution assumption with the asymptoted data.

While for the moment we are simply comparing times, this could be easily incorporated into a cost framework in order to model varying labor costs, the amortized cost of hardware, continuing expenses associated with teleoperation or autonomous control of the robots, the cost of repairs, etc.

5. RESULTS

To test our conjectures about sliding autonomy and user modeling, we have conducted a pilot study with three expert users in order to compare how well the human-robot team performs at differing levels of autonomy. In this paper we report the results of that study and the resulting small data set. Large scale experiments with many novice users and carefully prepared and executed protocol are currently underway.

In order to create a realistic teleoperation experience, operators sit at a workstation facing away from the robots and the workspace. They are able to see only the raw video output from one of the roving eye's cameras and the output of a visualization tool, which performs computational stereo on images taken by the roving eye to extract and display depth information relevant to the current task (Fig. 6). The robots are controlled via a 6 degree-of-freedom "Space Mouse".

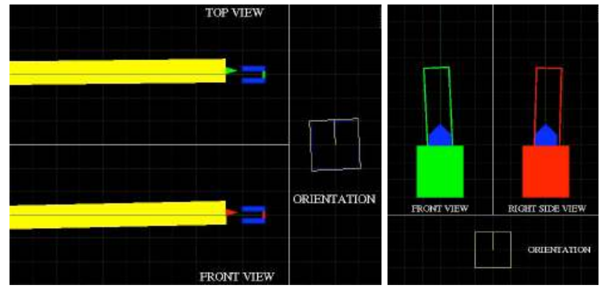

Fig. 6. Screen shots of the visualization tool for beam docking (left) and node bracing (right).

We have established a baseline for comparison across different modes of user involvement by running the system completely autonomously through ten attempts to assemble the entire square structure. A double-sided t-test with confidence threshold 0.95 revealed that the time taken for assembling one beam between two nodes is independent of which side of the square the beam is on, allowing us to choose a node-beam-node subassembly as our unit of analysis. Of 35 attempts, the autonomous system failed 4 times (11%), taking an average time of 516 seconds (see Tab. 1 and 2).

As a second reference, we had our users complete a total of 16 such subassemblies in full teleoperation mode. Each run took on average 732 seconds, with only one failure orruring. Between these two autonomy extremes, our data also includes 16 runs each of system- and mixed-initiative sliding autonomy. System-initiative trials were the most efficient with an average time of 500 seconds and a 100% success rate (Tab. 1 and 2).

Tab. 1. Efficiency results.

	mean time to completion (std dev)
Autonomous System	516 (125) seconds
Teleoperation	732 (227) seconds
System Initiative	500 (182) seconds
Mixed Initiative	529 (148) seconds

Tab. 2. Robustness and workload results.

	success rate [# of trials]	TLX workload (std dev)
Autonomous System	89% [35]	0
Teleoperation	94% [16]	52 (16)
System Initiative	100% [16]	27 (21)
Mixed Initiative	94% [16]	29 (13)

The histograms in Fig. 7 show that the fastest time of 331 seconds was recorded during a system-initiative run. All run times for system-initiative, mixed-initiative and fully autonomous operation had similar spreads between 331 and 905 seconds with clear peaks near the lower end of the range. In contrast, the fastest recorded teleoperation time was only 499 seconds, slightly faster than the mean time to completion with some amount of autonomy.

The histogram of the autonomous system's performance shows a bimodal distribution of run times with peaks around 400 and 580 seconds. While the first peak represents smooth runs without failures, the second peak indicates that the system had to spend additional time to recover from a near failure condition before completing the task. With only 16 data points, the other histograms are too sparse to see similar trends.

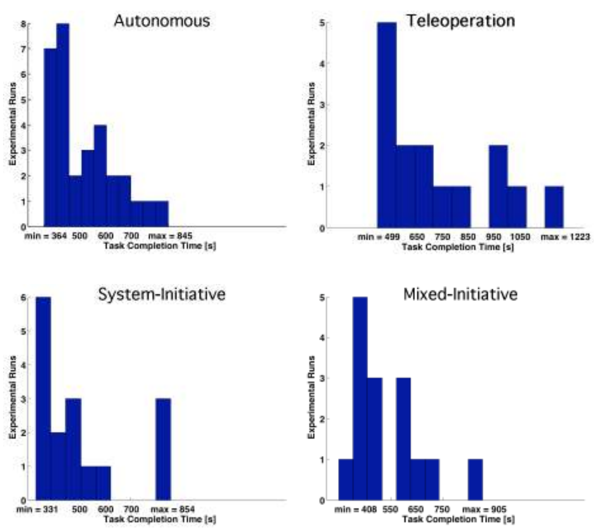

Fig. 7. Histograms of task run times for the four experimental cases.

At the end of each run, the subjects completed a NASA-TLX survey [9] to assess their perceived workload while controlling the robots. The survey takes into account factors such as mental, temporal, and physical demand, as well as effort and frustration (last column of Tab. 2). Measured on a scale proportional to workload, ranging from 0 to 100, the subjects reported a mean workload of 52 during complete teleoperation, compared to 29 and 27 for mixed- and system-initiative sliding autonomy, respectively.

6. DISCUSSION

While in general our hypotheses were accurate, here we discuss deviations likely due to the small sample size of the experiments. We also address open research questions and directions for future work.

6.1. Interpretation of Results

When comparing complete autonomy and pure teleoperation it is obvious that there is an inherent speed versus robustness trade-off. If we were willing to accept a 50% increase in time to completion, we could have a remote operator teleoperate all the robots throughout the entire task with only a few failures. However, our operator workload data indicates that in addition to the increased time to complete the assembly (mostly due to the inability to multi-task during assembly), operators swiftly become frustrated.

Our data suggests that it is possible to gain both efficiency and robustness by introducing sliding autonomy into the system. Either the system-initiative or mixed-initiative strategy might be preferable, depending on which factors are most important for a given application. Both approaches bring the time to completion down to the same level as the fully autonomous system while significantly reducing operator workload. Given the large standard deviations of the run times in Tab. 1 there is no meaningful way to claim that one autonomy strategy is faster than another. In fact, system-initiative being slower than mixed-initiative seems counterintuitive since in the latter case there is no time lost to reacquire situational awareness as the operator is actively following the system's moves. Under system-initiative, we expected a time penalty corresponding to the human shifting attention back to the task they are asked to help with. We expect that the larger data set of our current experiments will clarify these issues and support our expectations.

Once the decision has been made to incorporate sliding autonomy into a system, the trade-off of speed versus robustness changes to a balance of speed and operator workload. For non-time-critical applications, system initiative sliding autonomy is probably the most beneficial solution. Failures were essentially eliminated, and the operator easily would have been able to work on other tasks simultaneously. If speed of assembly is critical, then a mixed initiative sliding autonomy approach is the implementation of choice.

6.2. Future Work

Determining the proper action to take when the operator relinquishes control remains an open research question. If he returns control after completing the assigned task (and only the assigned task), it is straightforward; our current system makes this assumption. However, they may instead perform additional or different tasks, or even undo a previous portion of the assembly, instead of following the exact script the system expects. In order to properly resume control after the human has performed arbitrary actions, the system must first track the human's actions, as well as infer their current goals. Once this tracking is accomplished, the system may be able to infer useful cooperative actions to take with the robots still under autonomous control, as well as properly continue the mission when control is returned to the autonomous system. We are currently investigating this problem.

Our current scenario encompasses a relatively complex task, but contains minimal coordinated manipulation of a single object by multiple manipulator robots. Our earlier work involved a much simpler task, but required such coordinated manipulation [2]. We are now moving to a new scenario that combines a more complex and flexible assembly task with coordinated manipulation. For this new scenario, we will also add the missing planning layer to Syndicate in a way that supports our sliding autonomy paradigm.

7. CONCLUSIONS

Our experiments have shown that sliding autonomy allows us to combine the advantages of autonomous robot operation with the reliability of teleoperation at a level of mental workload easily tolerable by an operator. What used to be a trade-off of efficiency versus robustness when considering only the ends of the autonomy spectrum has become a balance of efficiency and operator work load. Under the sliding autonomy paradigm, a high degree of robustness can be assured since the availability of an operator serves as a safety net for the majority of failures the autonomous system does not detect. The amount of attention the operator pays to the task has only a slight effect on the time to task completion, and it does not influence system robustness. The time penalty incurred when the operator needs to reacquire situational awareness during system-initiative sliding autonomy is outweighed by the operator's ability to multitask, which he is unable to do in mixed-initiative. As a result, for most applications, system-initiative will likely be the desired mode of sliding autonomy. Clearly, if the autonomous system were unable to detect most failures, mixed-initiative would be preferred in order to compensate for the autonomous system's lack of reliability.

ACKNOWLEDGMENTS

This work has been supported by NASA under grant number NNA04CK90A. 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

1. Bostelman, R., Albus, J., Dagalakis, N., and Jacoff, A. RoboCrane Project: An Advanced Concept for Large Scale Manufacturing. In *Proceedings of the AUVSI Conference*, Orlando, FL, July 1996.
2. Brookshire, J., Singh, S., and Simmons, R. Preliminary Results in Sliding Autonomy for Assembly by Coordinated Teams. In *Proceedings of IROS*, 2004.
3. Burridge, R. R., Graham, J., Shillcutt, K., Hirsh, R., and Kortenkamp, D. Experiments with an EVA Assistant Robot. In *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space: i-SAIRAS 2003*, Nara, Japan, 2003.
4. Company website. http://www.3dconnexion.com/spacemouseplus.htm.
5. Dorais, G., Banasso, R., Kortenkamp, D., Pell, P., and Schreckenghost, D. Adjustable Autonomy for Human-Centered Autonomous Systems on Mars. Presented at the Mars Society Conference, 1998.
6. Fong, T., Thorpe, C., and Baur, C. Robot, Asker of Questions. *Robotics and Autonomous Systems*, 42, 2003.
7. Gillespie, R. B., Colgate, J. E., and Peshkin, M. A General Framework for Cobot Control. In *International Conference on Robotics and Automation*, Detroit, MI, 1999.
8. Goodrich, M. A., Olsen, D. R., Crandall, J. W., and Palmer, T. J. Experiments in Adjustable Autonomy. In *Proceedings of the IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents*, 2001.
9. Hart, S. G. and Staveland, L. E. *Human Mental Workload*, chapter Development of the NASA-TLX (task load index): Results of Empirical and Theoretical Research, pages 139–183. North-Holland, Amsterdam, 1988.
10. Kortenkamp, D., Burridge, R., Bonasso, P., Schreckenghost, D., and Hudson, M. An Intelligent Software Architecture for Semiautonomous Robot Control. In *Autonomy Control Software Workshop, Autonomous Agents 99*, 1999.
11. Scerri, P., Pynadath, D., and Tambe, M. Towards Adjustable Autonomy for the Real World. *Journal of Artificial Intelligence Research*, 17, 2003.
12. Sellner, B., Simmons, R., and Singh, S. User Modelling for Principled Sliding Autonomy in Human-Robot Teams. In Parker, L., Schneider, F., and Schultz, A., editors, *Multi-Robot Systems: From Swarms to Intelligent Automata*. Springer, 2005.
13. Sierhuis, M., Bradshaw, J., Acquisti, A., van Hoof, R., Jeffers, R., and Uszok, A. Human-Agent Teamwork and Adjustable Autonomy in Practice. In *Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space: i-SAIRAS*, Nara, Japan, 2003.
14. Stroupe, A., Huntsberger, T., Okon, A., and Aghazarian, H. Precision Manipulation with Cooperative Robots. In Parker, L., Schneider, F., and Schultz, A., editors, *Multi-Robot Systems: From Swarms to Intelligent Automata*. Springer, 2005.
15. Wannasuphoprasit, W., Akella, P., Peshkin, M., and Colgate, J. E. Cobots: A Novel Material Handling Technology (best paper award). International Mechanical Engineering Congress and Exposition, Anaheim, ASME 98-WA/MH-2, 1998.