

Sliding Autonomy for Complex Coordinated Multi-Robot Tasks: Analysis & Experiments

Frederik W. Heger
The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, U.S.A.
fwh@cs.cmu.edu

Sanjiv Singh
The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, U.S.A.
ssingh@ri.cmu.edu

Abstract—Autonomous systems are efficient but often unreliable. In domains where reliability is paramount, efficiency is sacrificed by putting an operator in control via teleoperation. We are investigating a mode of shared control called “Sliding Autonomy” that combines the efficiency of autonomy and the reliability of human control in the performance of complex tasks, such as the assembly of large structures by a team of robots. Here we introduce an approach based on Markov models that captures interdependencies between the team members and predicts system performance. We report results from a study in which three robots work cooperatively with an operator to assemble a structure. The scenario requires high precision and has a large number of failure modes. Our results support both our expectations and modeling and show that our combined robot-human team is able to perform the assembly at a level of efficiency approaching that of fully autonomous operation while increasing overall reliability to near-teleoperation levels. This increase in performance is achieved while simultaneously reducing mental operator workload.

I. INTRODUCTION

The demands on robotic systems are rapidly increasing toward more and more complex application scenarios. In many cases, the requirements far exceed the abilities of any single robot of manageable complexity. One approach to this problem is to use teams of robots whose collective capabilities are required to complete the scenario, but where each individual robot’s abilities cover only a small subset of the solution. We would like to deploy such teams in difficult environments (with high uncertainty, tight coordination requirements, etc.) that may be inaccessible to humans. Such scenarios demand great autonomy on the part of the robots. However, reliable handling of all possible contingencies in such complex environments is far beyond the current state of the art.

Traditionally, robots are either fully autonomous or must be carefully controlled by a remote operator. The latter is particularly the case in applications where reliability is paramount. The goal of complete reliable autonomy remains elusive as systems become increasingly complex and it becomes intractable to anticipate and provide for all possible failure conditions. As a result, autonomous systems are not likely to be able to function in complex environments entirely without human input in the near future. However, human control of

robots through teleoperation is often painstakingly slow and requires great concentration on the part of the operator, in addition to latency and the high cost of data communication incurred by the system to provide the necessary situational awareness to the user. Thus, there is a need for a hybrid of autonomy and teleoperation for robot systems to extend their capabilities and usefulness.

We are investigating an approach, called “Sliding Autonomy” (also referred to as “Adjustable Autonomy” [1][2][3]) where control over subtasks is dynamically assigned to either an autonomous robot or a human operator, and use it to control a team of robots to perform coordinated assembly operations in a constrained workspace. In our scenario, the performance characteristics of full autonomy are high efficiency (measured by low completion time) and a non-trivial failure rate, whereas complete teleoperation is characterized by lower efficiency with very few failures. Our primary objective with Sliding Autonomy is to enable the team of robots to combine the two modes of operation in a way that both increases overall team efficiency and reduces the number and probability of irrecoverable failures.

Here we introduce our approach to modeling the effects of Sliding Autonomy by using Markov models to describe the flow of control between autonomy and teleoperation for all functional blocks of the scenario. The method is used to analyze overall system performance based on performance of main component tasks of our assembly operation. We show that our models agree with the qualitative expectations that Sliding Autonomy allows the team to operate at speeds comparable to autonomous operation while significantly improving reliability. In addition, the model can be used to perform system-level analysis of the flow of control among agents on the team to help us identify bottlenecks and potential for improvement. We also report the results of a user study during which expert operators worked with our robots to complete assemblies with and without Sliding Autonomy. The results of the study support both our intuitive expectations and modeling results and show that Sliding Autonomy provides an increase in efficiency of 30-50% over teleoperation together with an increase in reliability of up to 13% over autonomous operation depending on the operator’s ability.

II. RELATED WORK

Our work combines and extends concepts from primarily three fields of research: coordinated manipulation, robots assisting humans and joint human-robot teams.

Work by Rus et al. examines a team of robots collaborating to reorient and move furniture in a room by appropriately pushing at it [4]. The emphasis of this research is on composing an assembly plan from a grammar of component tasks. Investigating a similar problem from a control perspective, Chaimovizc et al. enabled a team of robots to manipulate boxes by driving in formation to maintain the desired forces acting on the objects [5]. Stroupe et al. use the CAMPOUT architecture to coordinate fully autonomous robots with purely behavior-based strategies to perform very tightly coupled component tasks [6]. Two heterogeneous robots carry a beam and position it with respect to an existing structure with sub-centimeter accuracy. While our previous work required manipulation by multiple robots [7], the current focus is on higher-level coordination between robots and humans. Accordingly, we switched from a very detailed Petri-net-based single-task model to a more abstract complete system representation. The main advantage of this new representation is its ability to evaluate system performance without the need to run extensive simulations.

NASA's ASRO project has investigated mobile robot assistants to help an astronaut by carrying tools, helping to manipulate objects, and providing sensor information [8]. While the robot was physically working alongside the astronaut, it was completely teleoperated by an operator at a remote site. The COBOT project seeks to make manually operated machines more intelligent by performing the fine manipulation required for a task autonomously based on force input from a human operator [9][10]. For example, by simply pushing against a mechanism, the operator can command a robot to move very accurately along a desired trajectory. The roles of the human operator and the robot are clear and unvarying, and both must operate simultaneously. By design, the operator in our scenario is not collocated with the robots and thus is only able to have limited interaction with what happens in the workspace. As a result, our robots need a certain degree of autonomy to function while the operator is not actively attending to them.

Offloading part of a tedious operation to an autonomous agent has been implemented in applications such as scheduling meetings with many participants. Scerri has proposed Sliding Autonomy for software agents applied to a daily scheduler [11]. It autonomously attempts to resolve timing conflicts (missed meetings, group discussions, personal conflicts, etc.) among some set of team members. While this system is able to handle last-minute changes in plans, once a valid meeting schedule is found no execution-time failures will occur. Due to sensor noise and inaccuracy in real-world applications such as ours, we specifically need to be able to deal with failures in nominally valid plans.

Most closely related to our work are joint teams of humans and robots where control is shared between both in response

to the overall system's performance. 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 [3]. Their scenario involved a pick-and-place operation during which adjustable 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. Using an approach to human-robot interaction related to ours, Fong et al. enable the robot and the user to participate in a dialogue [12]. The robot can ask the operator to help with localization or to clarify sensor readings. The operator can also make queries of the robot. Our work extends these experiments with a more complex multi-robot scenario and a finer granularity of Sliding Autonomy.

Due to high demands on reliability, space applications are often found among scenarios employing Sliding Autonomy. Dorais et al. provide several examples of how Sliding Autonomy will be essential for space operations where demands on the operator must be focused and minimized [2]. Our work presents a working implementation of Sliding Autonomy in a complex domain and tries to provide some general insights into the effects of this control paradigm on the performance of the overall system.

III. MOTIVATION

The motivating scenario for our work is a robotic construction crew of three heterogeneous robots. Each agent has a distinct set of capabilities needed to perform a complex assembly operation that decomposes into many component tasks. The team (see Fig. 1) consists of a strong but imprecise NIST RoboCrane [13], an ATRV-2-base Mobile Manipulator with a 5-DOF arm built by Metrica/TRACLabs [14] and an RWI B24 synchro-drive robot with a pan-tilt stereo pair, the Roving Eye [15].



Fig. 1. The Mobile Manipulator (top left), the Roving Eye (top right), the Crane (bottom left) and the completed structure (bottom right).

The robots assemble four sides of a square structure of four beams supported by four corner nodes (Fig. 1, bottom right) following the script shown in Fig. 2. The nodes are passively mobile and cannot resist the forces required to

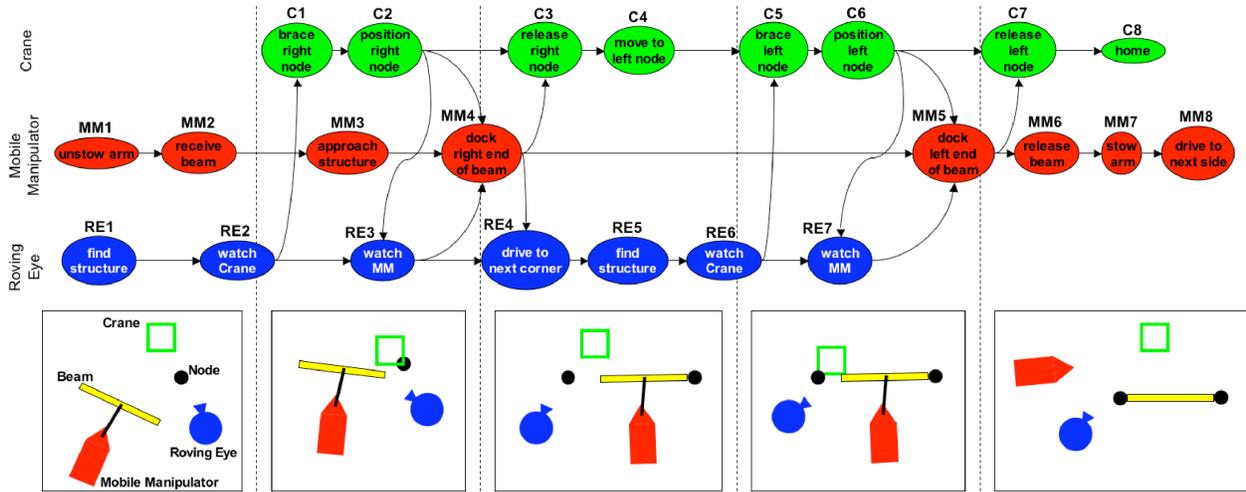


Fig. 2. Task script for and a plan view graphical representation of a single node-beam-node subassembly. The robots are represented by their characteristic shapes (the square represents the Crane’s end-effector). Arcs between nodes on different agents indicate preconditions for the target node’s task.

securely dock a beam into them – they slide away unless they are “braced” by another robot. To perform the assembly task, the Crane braces a node using relative pose information the Roving Eye can extract from fiducials attached to all relevant bodies. The Mobile Manipulator then moves a beam into place and connects it to the node (again using tracking information provided by the Roving Eye). After the Roving Eye relocates to the next corner of the square, the same sequence of bracing and docking operations is repeated to construct a node-beam-node subassembly. To complete the task, the Mobile Manipulator repositions to the next side of the square. During testing and experimentation, we repeat this scenario four times to assemble an entire square structure.

Our team of robots regularly constructs this structure without supervision or intervention from an operator in seven to eight minutes per side or approximately thirty minutes for the entire square. However, in the course of the assembly much can go wrong. Failures range from the Roving Eye losing track of a part of the structure it needs to see, to the Mobile Manipulator maneuvering the beam into a spot from which docking becomes impossible, to terminal failures such as a beam being knocked off the Mobile Manipulator’s arm.

It is possible to recover from some of the failures with simple contingencies. Others, however, are impossible to recover from autonomously because the available sensors do not provide adequate information. Still, in the majority of problem cases, the autonomous system can at least detect that something is wrong, even if it cannot determine the particular problem. In such a situation, our robots are able to ask for help from an operator using Sliding Autonomy.

As a whole, the entire assembly team is a complex system consisting of many subsystems. Gathering data to understand the inner workings of the team as a whole is very time-consuming. However, during development and testing we can easily collect large amounts of data for specific subsystems working autonomously or under teleoperation. In order to gain a broader understanding of the team’s performance, we introduce a modeling approach that combines abstract

representations of component performance (for which we have lots of data) into a model of the entire robot team. With this composite model we want to predict overall efficiency and reliability without having to run system simulations. We also want to use the model to perform system analysis to predict bottlenecks and potential deadlocks.

IV. SLIDING AUTONOMY

A. Infrastructure

We have developed a tiered distributed multi-agent architecture to support Sliding Autonomy for our team of robots. Each component task of the scenario is split into a monitor and an action component, each of which can either be under autonomous or manual control. It is often the case that the robots are good at carrying out a specific task, but they lack the necessary sensors to reliably determine when they are done. Alternatively, they may have the sensors but lack autonomous control. For example, the Mobile Manipulator’s arm could be under human control during a docking while the Roving Eye autonomously repositions itself to maintain a clear view of all fiducials required to determine accurate pose information and track task progress.

In order to take advantage of Sliding Autonomy, the robot team needs a principled way of deciding who should be in control of which part of a component task in order to maximize the team’s overall performance. In order to make a more informed decision, we developed a user performance modeling system [16] to collect data about the robots’ performance as well as that of each operator working with the team. Each time a new component task starts or an active one fails, this system evaluates the expected benefit over the remainder of the entire scenario of assigning control of a particular component task either to an autonomous robot or an operator.

An operator working with the team of robots sits in front of a computer screen away from the robots’ workspace. His screen shows a graphical interface that allows him to interact with the robots and change control of active component tasks. In addition, he is provided with a video feed from the Roving

Eye’s cameras, a visualization tool that provides 3-D scene information, and audio cues if one of the robots asks for help. When asked for assistance, the operator can control one robot at a time using a 6-DOF SpaceMouse [17].

B. Modes of Control

In addition to the obvious control modes of Teleoperation and full Autonomy, we introduced two intermediate modes in the autonomy spectrum. System-Initiative Sliding Autonomy (SISA) and Mixed-Initiative Sliding Autonomy (MISA) differ in the degree to which the human is involved in the operation of the team.

In SISA mode, the operator does not follow the robots’ progress until called to help with a problem or a component task is assigned to him based on his prior performance. At that time he must become aware of the state of assembly from a cold start to perform the task at hand [18]. At any time the human may pass control to the autonomous system if he feels that the particular problem has been solved and the system will be able to continue autonomously. While in SISA mode, the operator can perform other unrelated tasks in parallel to working with the robot team. Since there are situations where the appropriate reaction to failures is time critical, the inherent delay due to the acquisition of situational awareness in SISA makes this mode ill-suited for such applications.

To address this issue, we implemented a second Sliding Autonomy mode, MISA, where in addition to waiting to be assigned a task, the operator continually monitors progress and is able to actively take over control whenever he sees fit. The ability to proactively take control enables the operator to avoid certain failures before they are detected or predicted by the autonomous system. However, this mode demands a much higher mental workload of the operator and does not allow him to multi-task while working with a team of robots.

TABLE I
EXPECTED QUALITATIVE SYSTEM BEHAVIOR.

	Time to Completion	Success Rate	Operator Workload
Teleoperation	worst	best	worst
Autonomous	best	worst	best
SISA	good	good	better
MISA	better	better	good

Table I summarizes our qualitative expectations for overall system performance. For each metric, we expect Teleoperation and Autonomy to behave in opposite fashions. While Teleoperation is slow, demanding and reliable, we expect autonomous operation to be fast and less reliable, without any workload on an operator. In the Sliding Autonomy modes, we expect a higher workload for MISA to yield better performance while SISA’s performance suffers from its lower operator demand.

V. SYSTEM MODELING AND ANALYSIS

A. System Modeling

Our current assembly script (Fig. 2) can be represented by a linear sequence of functional blocks as shown in Fig. 3.

Each block involves multiple robots and can be decomposed into finer levels of detail describing the actual tasks of each individual agent. While such fine granularity is of interest when studying the internal workings of a system [7], it unnecessarily complicates the understanding of the effects of Sliding Autonomy on a robot team’s performance for a given scenario. For the remainder of this paper, we will consider this simplified task script as the model underlying our analysis.

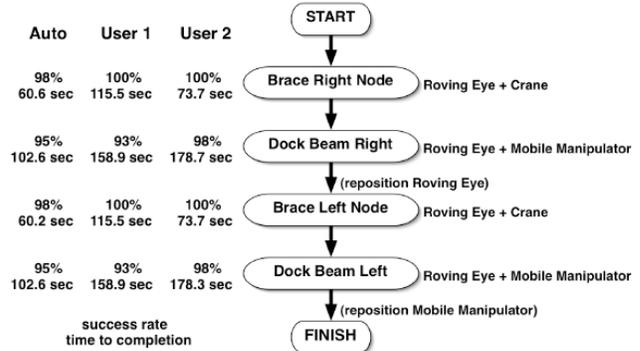


Fig. 3. Simplified script for our assembly scenario. Each of the tasks shown decomposes into a sequence of smaller tasks. The data shown on the left was used to predict system performance. The numbers were collected on a per task basis based on several months of experience of the users with our system.

To model the assembly scenario performed by our robots, we represent each functional block along the sequence in Fig. 3 with a Markov system (Fig. 4) corresponding to the mode of Sliding Autonomy we want to model. With the team working in a particular mode, all functional blocks follow the same transition model, but their transition probabilities may vary. While we can use this model to study the effects of different transition probability assignments (i.e. user and robot abilities), we can also use information collected by our user performance models over many trials to obtain realistic estimates appropriate for our implementation.

The functional models for autonomous and manual operation are simple (Fig. 4, top). The autonomous robots have a certain probability of succeeding at the task at hand and continuing on to the next functional block. Alternatively, they may fail, which will either be a catastrophic failure or one for which an autonomous recovery strategy exists. During teleoperation, we assume that the operator will continue to try and achieve the task goal until either he succeeds and continues on to the next functional block or he gets stuck in an unrecoverable condition.

The models for System-Initiative and Mixed-Initiative Sliding Autonomy are built up from the basic models for teleoperation and autonomy as indicated by corresponding arrows in Fig. 4. They are almost identical, with an additional transition (labeled P(takeover)) from autonomous operation to human operation when the operator decides to take over control in MISA mode. At the beginning of a block, the user model decides who attempts to perform the task [16]. If the autonomous system is chosen and the attempt fails in a way that is recoverable, the user model reevaluates the situation and makes another decision about who will assume control. If the human is in charge, he may choose to yield control

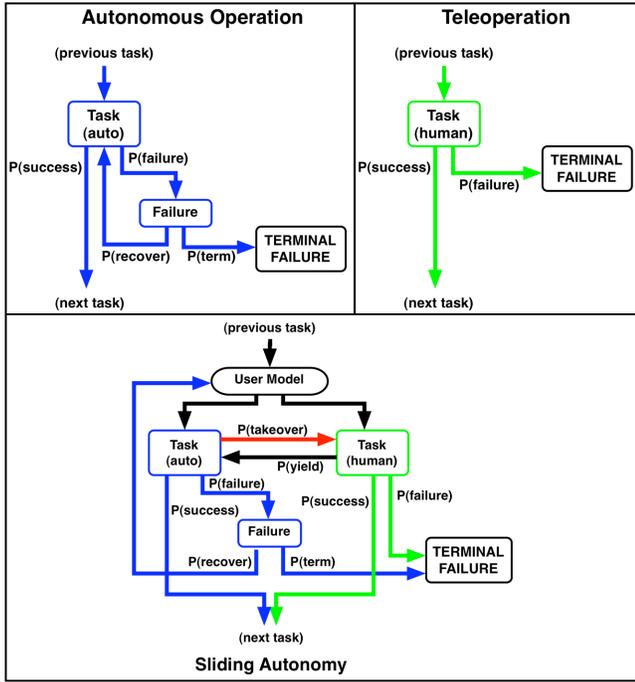


Fig. 4. Functional models for the different modes of control. Each block in the script can be represented by the transition model for the chosen mode of operation. Corresponding arrows are used to indicate how the sliding autonomy model is built up from the basic teleoperation and autonomy models. Note: the transition labeled $P(\text{takeover})$ is only present in MISA mode.

back to the robots at any time. The remaining transitions are analogous to pure autonomy or teleoperation.

Considering the entire task sequence, we obtain a large Markov system with two absorbing states: one for the goal state (successful completion) and one representing terminal failure at any point along the script. We can estimate the robot team's failure rate by computing the probability of being absorbed by the TERMINAL FAILURE state. We can also compute the number of expected visits to any state prior to absorption. If we assign completion times to the appropriate states, we can add the times associated with each state multiplied by the expected number of visits to that state to obtain an estimate of the time taken to complete the scenario. The number of visits to each state also provides us with valuable information about potential bottlenecks in the flow of control. If a particular state is visited disproportionately often, we can either focus our efforts toward improving the autonomous system's performance at the particular step, or we can specifically train operators for the reoccurring problem.

B. Performance Analysis

Given a Markov system made up of transition blocks as described above, we can analyze the system's behavior using standard techniques [19]. If we number the states with the two absorbing states consecutively at the end of the list, we can write the system's transition matrix P for n states as

$$P = \begin{pmatrix} Q & R \\ O & I \end{pmatrix} \quad (1)$$

where

$$Q = (n - 2) \times (n - 2) \text{ matrix}$$

$$R = (n - 2) \times 2 \text{ matrix}$$

$$O = 2 \times (n - 2) \text{ zero matrix}$$

$$I = 2 \times 2 \text{ identity matrix}$$

Q is the transition matrix between non-absorbing states while R represents transitions from non-absorbing states to absorbing states. If we only consider the non-absorbing states, we can write the matrix N to determine the expected number of visits to each state as

$$N = I + \sum_{i=1}^{\infty} Q^i = (I - Q)^{-1} \quad (2)$$

where I is the appropriately sized identity matrix. The entry N_{ij} represents the number of times we expect to visit state j when starting in state i before absorption. If we write a vector t whose entries are the average time spent in each state, then the dot product of t with the first row of N gives us the expected time the system will take before completing the scenario (either successfully or unsuccessfully). Note that this timing estimate will differ from the robots' performance since revisiting a state in real operation corresponds to either recovering from a failure (in which case the second attempt may take significantly longer than the first) or the result of a take-over or yield of control in Sliding Autonomy mode (which can cause the task to complete faster than if it had to run from the start). A more faithful representation of the robot team's performance would include additional states for each of these cases with appropriate timings assigned to them.

To compute the probabilities of success or failure for a task sequence described by a given Markov system we compute matrix M as

$$M = NR. \quad (3)$$

The entry M_{ij} represents the probability of ending up in absorbing state j when starting from (non-absorbing) state i . In our case, with the states numbered as described above, the first row of M contains the probability of success (for arriving at the FINISH state) and the probability of failure (for arriving at the TERMINAL FAILURE state).

Using accumulated data for operators and robots to estimate realistic transition probabilities and timing characteristics, we computed our performance metrics for each of the four modes of Sliding Autonomy. As mentioned above, the grouping into functional blocks used here does not correspond directly to component tasks in our actual scenario for which we have detailed timing information. Instead, the main underlying component tasks were selected to estimate completion time.

The system model we have used here makes a simplification of the decision process used to decide whether the operator should be asked for help. Since these decisions are based on the number of preceding failures in a given functional block, to fully represent this behavior each block would have to consist of copies of itself corresponding to specific failure counts. To avoid this explosion of states, our composite model is

based on the assumption that the performance model does not change over the course of the scenario. Thus, we can represent the robots' approximate behavior with a single layer of each functional block. The user model transitions were set to always transition to the state corresponding to the agent who would be assigned that task given its performance model.

C. Results

The results of this analysis are shown in Table II. The empirical data underlying the transition probabilities encoded in this model were collected over the course of several months of the two users working with the system in teleoperation and autonomy modes. Each probability value shown in Fig. 3 is based on between 15 and 40 data points. The constant time blind repositioning moves of the Roving Eye and the Mobile Manipulator were not included in the model. They are pre-assigned to the autonomous robots during Sliding Autonomy operation and can be considered 100% reliable. To allow a comparison of the predicted results with the experimental results in the following section, corrections were made when showing the completion times in Table II (210 seconds were added for teleoperation, 97 seconds for the other modes). These times represent the combined times for the both blind moves based on approximately 20 and 50 data points, respectively, collected over several months.

TABLE II

EXPECTED USER PERFORMANCE FOR COMPLETING ONE SIDE OF THE ASSEMBLY COMPUTED USING THE COMPOSITE MARKOV MODELS.

	User 1		User 2	
	Time to Completion	Success Rate	Time to Completion	Success Rate
Teleop	740 sec	86%	709 sec	96%
Autonomy	571 sec	76%	571 sec	76%
SISA	612 sec	79%	696 sec	91%
MISA	673 sec	83%	706 sec	92%

The overall trend in these results agrees with our qualitative expectations in Table I: teleoperation takes longest and shows the best reliability, whereas autonomy is fastest with a much lower success rate. Sliding Autonomy lies in between the two with MISA being slightly more reliable. Note that the timing predictions are overestimates because the model considers multiple visits to a state to take the same time. For example, if a component task performed by an autonomous robot fails in a recoverable fashion, then the recovery action enters the timing calculation as another full-time attempt while in many cases the bulk of the work has already been done and the task can be completed much faster. To make more accurate prediction of completion time, the model could discount repeated visits when computing the time estimates.

With the model shown to be a simple but plausible representation of our system's behavior under Sliding Autonomy, we can exploit the structure of the Markov system further to analyze the flow of control throughout the scenario. For example, we can determine which states along the task script are visited disproportionately often. This information would

allow us to either provide the team with an expert user with particularly good performance for the problematic task, or it could focus our efforts in improving autonomous recovery strategies for failure-prone tasks. Both measures increase overall performance of the entire team.

VI. EXPERIMENTAL EVALUATION

A. Experiment Design

To compare the predictions of our model to the robot team's actual performance, we conducted a user study to quantify the effects of Sliding Autonomy on our assembly scenario. Since training of novice subjects to the required familiarity with our robots and scenario proved to be impossible within a reasonable amount of time, the results presented here are from two expert users familiar with our system.

An experimental run consisted of one attempt to assemble one side of the square structure. For each run, we recorded the total time taken and marked the run a success or failure. After each run, the operator was given a NASA TLX (Task Load Index) survey designed to measure mental workload based on a series of questions asking the user to rank components of workload (such as mental demand, frustration, etc.) on a scale from 1 to 10 [20]. The result is a score between 0 and 100 proportional to perceived workload.

B. Results

With the exception of the cells marked in bold font in Table III, our results for both users generally agreed with our early predictions in Table I. They also agree with our modeling results in Table II if we allow for the overestimate in timing mentioned above. Under Teleoperation, the team worked 1.5-2 times slower than fully autonomous, but at the same time the success rate was improved by about 30%.

TABLE III

USER PERFORMANCE RESULTS. DISCREPANCIES FROM EXPECTED RESULTS ARE HIGHLIGHTED IN BOLD. ALL DATA CORRESPOND TO ATTEMPTS TO ASSEMBLE ONE SIDE OF THE SQUARE STRUCTURE.

User 1	Mean Time to Completion [std dev]	Success Rate (trials)	TLX Workload [std dev]
Teleoperation	729 [139] sec	100% (12)	42 [10]
Autonomous	437 [94] sec	75% (24)	—
SISA	462 [63] sec	75% (20)	16 [16]
MISA	492 [140] sec	81% (16)	17 [11]

User 2	Mean Time to Completion [std dev]	Success Rate (trials)	TLX Workload [std dev]
Teleoperation	911 [193] sec	92% (12)	71 [9]
Autonomous	437 [94] sec	75% (24)	—
SISA	458 [106] sec	85% (20)	50 [14]
MISA	445 [72] sec	88% (16)	34 [11]

Fig. 5 clearly shows the trends we expected to see in our performance metrics. With some amount of autonomy as part of the control mode, the time to complete the scenario

approached the time taken by the autonomous system alone while in teleoperation mode the time was much higher. At the same time, we found that nearly any amount of user input to the system (with the exception of User 1 in SISA mode) increases the probability of success for the entire scenario. The more human input, the more reliable the performance.

Fig. 6 best exemplifies the subjectivity of the data provided by the NASA TLX survey. While User 1 generally follows the expected trend, User 2’s perception of workload increased when moving from MISA to SISA even though the operator should have to pay less attention to the robots there.

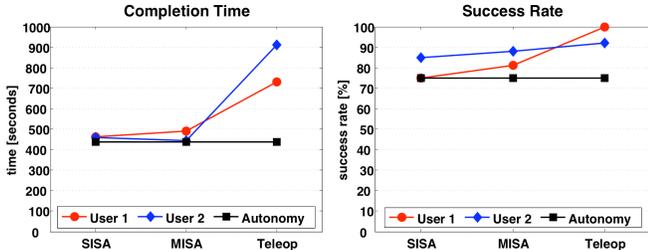


Fig. 5. Comparison of our two subjects’ independent runs and the fully autonomous system. Completion time and success rate generally follow our expectation.

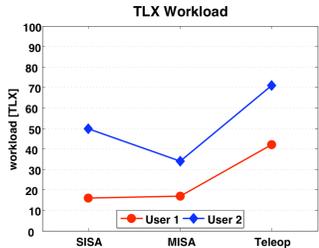


Fig. 6. Comparison of our two subjects’ perception of workload. These results are necessarily subjective and depend heavily on the both the task and the user.

The histograms of User 1’s performance (Fig. 7) show again that the autonomous modes are clearly faster than teleoperation. The fastest run time was recorded under autonomous operation at just over 300 seconds. At the same time, a large portion of the teleoperation runs took almost 900 seconds, much longer than the slowest run during an autonomy trial. The multi-modal groupings correspond to assembly attempts with varying numbers of failures. The left-most grouping represents smooth runs without failures, and the failure count increases for each grouping to the right.

VII. DISCUSSION

In general we found that our hypotheses regarding our team’s performance in both Sliding Autonomy modes accurately predicted their true behavior. While the simplicity of our current model forces us to accept certain inaccuracies, particularly in the prediction of performance time, it allows us to perform meaningful analysis of the flow of control within the team of robots and human operators.

Our results clearly show the inherent trade-off of speed versus robustness at the two extremes of the autonomy spectrum. If we can accept an increase in the time needed to complete the scenario of 50-100%, a skilled operator can teleoperate

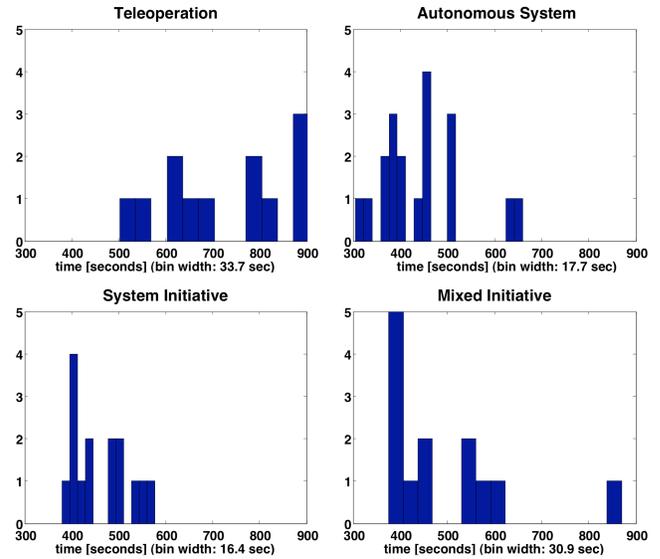


Fig. 7. Histograms comparing User 1’s performance when using the different Sliding Autonomy modes. The horizontal axis marks completion time in seconds, and the height of the bars shows the number of experimental runs for each time interval.

the entire team with minimal failures. The dramatic increase in mental workload for that operator, however, reduces the practical usefulness of this approach significantly.

As we have shown, introducing Sliding Autonomy as a mode of operation can overcome this dilemma. As shown in Fig. 5, any amount of autonomy reduces completion time and human assistance generally increases reliability. User 1’s low performance in SISA mode can be explained by his comparatively lower performance models (see Fig. 3) that caused the autonomous system to remain in control for much of the scenario. Taking advantage of both opposing trends, Sliding Autonomy enables a team of robots to work both efficiently and reliably by incorporating the human operator’s intuition to quickly understand problematic conditions and initiate recovery measures. With tedious and non-critical tasks still being performed autonomously, Sliding Autonomy also provides a clear benefit to the mental demand on the operator.

The subjective nature of the TLX results makes a direct user-to-user comparison meaningless. In addition, there are at least two factors contributing to a user’s perception of workload that are not well captured within the TLX framework: boredom of the operator and frequency of the system’s calls for help. If a strong user working in SISA mode is asked for help so frequently that effective multi-tasking is impossible, the constant need to regain situational awareness can be much more stressful than continuously monitoring the robots’ progress, as in MISA mode. After the experiment, User 2 confirmed that this was the case during SISA trials.

We are currently in the process of moving to a more complex assembly scenario with a more flexible ordering of component tasks and the potential for parallelism in the assembly process. We will have to extend our models to be able to support such features, and we are investigating how Sliding Autonomy can be used to prioritize requests for assistance

from different robots working on different component tasks in order to optimize overall team performance. Part of this new scenario will also be a planning/scheduling system to task and retask robots and operators. We are investigating ways of integrating Sliding Autonomy concepts into the planning stage as well with the goal of taking advantage of performance gains similar to those reported here during task execution.

VIII. CONCLUSION

As a hybrid of autonomy and teleoperation, we have shown that Sliding Autonomy combines the benefits of each mode individually and enables our team of robots to operate both efficiently and reliably at the same time. By exploiting the strengths of each approach, this solution neither overly burdens the operator nor requires unmanageable complexity of the autonomous robots to realize its gain in performance.

We showed that the difference between SISA and MISA operation (i.e. the amount of attention an operator pays to the system) does not noticeably improve the team's overall efficiency for our scenario, but that a human's constant attention raises the success rate of completing the scenario. The availability of an operator serves as a safety net for the majority of failures the robots cannot recover from autonomously.

Purely from a system performance point of view, the choice between SISA and MISA depends heavily on the system as a whole. If the operators are comparable in skill level with the autonomous system, and the system is able to perform significant portions of the task on its own, then the humans can productively multi-task when operating under SISA. Clearly, if the autonomous system were unable to detect most failures, MISA would be the preferred method in all cases in order to compensate for the autonomous system's lack of reliability. With the system able to reliably detect failure conditions, the slightly lower success rate of SISA compared to MISA may be outweighed by the fact that a few operators can oversee several different teams at the same time. For very skilled users, however, their abilities often lead to them being continuously asked for help because the system defers to the operator when it fails rather than trying over again.

For scenarios where optimizing both efficiency and reliability is not absolutely essential, a Sliding Autonomy system needs a way to adjust its balance of optimizing performance at all cost versus frequency of interruption of its operators. This mechanism is needed to avoid putting skilled operators at the mercy of the system to either explicitly spend all their time monitoring the operation in MISA mode or to be unable to perform parallel work in SISA mode due to constant calls for help. If the user has the ability to specify his willingness to be interrupted, it can be directly incorporated into the decision process of who will be assigned the next task.

Our user experiment has allowed us to validate our modeling approach to Sliding Autonomy. The ability to evaluate overall system behavior based on data collected on a component task basis enables this approach to model and analyze a Sliding Autonomy system before it is implemented. The model also allows us to determine problem areas in an existing system and focus our efforts to improve its overall performance.

ACKNOWLEDGMENT

This work has been supported by NASA grant NNA04-CK90A. The experiments were conducted together with Brennan Sellner and Laura Hiatt. In addition, the authors would like to thank the many individuals who contributed to the project over the years: Rob Ambrose, David Apfelbaum, Jon Brookshire, Rob Burrige, Brad Hamner, Dave Hershberger, Myung Hwangbo, Simon Mehalek, Metrica/TRACLabs, Josue Ramos, Trey Smith, Pete Staritz, and Paul Tompkins.

REFERENCES

- [1] M. A. Goodrich, D. R. Olsen, J. W. Crandall, and T. J. Palmer, "Experiments in Adjustable Autonomy," in *Proceedings of the IJCAI Workshop on Autonomy, Delegation and Control: Interacting with Intelligent Agents*, 2001.
- [2] G. Dorais, R. Banasso, D. Kortenkamp, P. Pell, and D. Schreckenghost, "Adjustable Autonomy for Human-Centered Autonomous Systems on Mars," Presented at the Mars Society Conference, 1998.
- [3] D. Kortenkamp, R. Burrige, P. Bonasso, D. Schreckenghost, and M. Hudson, "An Intelligent Software Architecture for Semiautonomous Robot Control," in *Autonomy Control Software Workshop, Autonomous Agents 99*, 1999.
- [4] D. Rus, B. Donald, and J. Jennings, "Moving Furniture with Teams of Autonomous Robots," in *Proceedings of 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems 95*, vol. 1, 1995.
- [5] L. Chaimowicz, T. Sugar, V. Kumar, and M. Campos, "An architecture for tightly coupled multi-robot cooperation," in *Proceedings of ICRA 2001*, Seoul, Korea, May 21-26, 2001.
- [6] A. Stroupe, T. Huntsberger, A. Okon, and H. Aghazarian, "Precision Manipulation with Cooperative Robots," in *Multi-Robot Systems: From Swarms to Intelligent Automata*, L. Parker, F. Schneider, and A. Schultz, Eds. Springer, 2005.
- [7] J. Brookshire, "Enhancing Multi-Robot Coordinated Teams with Sliding Autonomy," Master's thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2004.
- [8] R. R. Burrige, J. Graham, K. Shillcutt, R. Hirsh, and D. Kortenkamp, "Experiments with an EVA Assistant Robot," in *Proceedings of iSAIRAS 2003*, Nara, Japan, 2003.
- [9] R. B. Gillespie, J. E. Colgate, and M. Peshkin, "A General Framework for Cobot Control," in *International Conference on Robotics and Automation*, Detroit, MI, 1999.
- [10] W. Wannasuphprasit, P. Akella, M. Peshkin, and J. E. Colgate, "Cobots: A Novel Material Handling Technology (best paper award)," International Mechanical Engineering Congress and Exposition, Anaheim, ASME 98-WA/MH-2, 1998.
- [11] P. Scerri, D. Pynadath, and M. Tambe, "Towards Adjustable Autonomy for the Real World," *Journal of AI Research*, vol. 17, 2003.
- [12] T. Fong, C. Thorpe, and C. Baur, "Robot, Asker of Questions," *Robotics and Autonomous Systems*, vol. 42, 2003.
- [13] R. Bostelman, J. Albus, N. Dagalakis, and A. Jacoff, "RoboCrane Project: An Advanced Concept for Large Scale Manufacturing," in *Proceedings of the AUVSI Conference*, Orlando, FL, July 1996.
- [14] "Metrica/TRACLabs Arm Website," Company website. [Online]. Available: "http://www.traclabs.com/robotarm.htm"
- [15] R. Simmons, J. Fernandez, R. Goodwin, S. Koenig, and J. O'Sullivan, "Lessons learned from xavier," *IEEE Robotics and Automation Magazine*, vol. 7, no. 2, pp. 33-39, June 2000.
- [16] B. Sellner, R. Simmons, and S. Singh, "User Modelling for Principled Sliding Autonomy in Human-Robot Teams," in *Multi-Robot Systems: From Swarms to Intelligent Automata*, L. Parker, F. Schneider, and A. Schultz, Eds. Springer, 2005.
- [17] "SpaceMouse manual," Company website. [Online]. Available: "http://www.3dconnexion.com/spacemouseplus.htm"
- [18] B. Sellner, L. Hiatt, R. Simmons, and S. Singh, "Attaining Situational Awareness for Sliding Autonomy," in *Proceedings of HRI 2006*, 2006.
- [19] P. Bonacich, "Textbook on Mathematical Sociology, Chapter 9: Markov Chains - II." [Online]. Available: <http://www.sscnet.ucla.edu/soc/faculty/bonacich/chapt9-2.pdf>
- [20] S. G. Hart and L. E. Staveland, *Human Mental Workload*. Amsterdam: North-Holland, 1988, ch. Development of the NASA-TLX (task load index): Results of Empirical and Theoretical Research, pp. 139-183.