Sliding Autonomy for UAV Path-Planning: Adding New Dimensions to Autonomy Management

Lanny Lin
Vivint, Inc.
lanny@lannyland.com

Michael A. Goodrich
Department of Computer Science
Brigham Young University
mike@cs.byu.edu

ABSTRACT

Increased use of autonomy also increases the need for humans to interact with or manage autonomy. We propose a new variation of sliding autonomy useful for planning problems over a spatial region. With this approach, the user can influence the behavior of the autonomous system via spatial constraints and temporal constraints. We present a set of user interface designs to implement sliding autonomy for Unmanned Aerial Vehicle path-planning to support Wilderness Search and Rescue. Interactivities along these new dimensions allow the user to allocate degrees of authority and flexibility to the robot's algorithms. We evaluate the usefulness of the approach against manual and simple pattern path-planning methods with a user study. Results show that the sliding autonomy approach performs significantly better than the other two methods without increasing the users' mental workload, and *the performance of the human-autonomy team outperforms either human or autonomy working alone.*

Keywords

human-robot interaction; path-planning; adjustable autonomy; supervisory control; sliding autonomy

1. INTRODUCTION

With the rapid advancement in technology, people are seeing increased use of autonomy to augment human abilities and support human decision-making in many application domains (e.g., [13, 11, 32, 38]). At the same time, increased use of autonomy means increased need for humans to interact with or manage autonomy [2]. Even for so-called fully autonomous systems, human input can potentially improve the system's performance and safety [9]. The humans in such interactions manage autonomy because "only people are held responsible for consequences...and only people decide on how authority is delegated to automata" [45].

When humans manage autonomous systems, their responsibilities often include monitoring the safety of the system, supervising autonomy to achieve acceptable performance, and making sure autonomy is working toward the collective goal of the overall system. In many emerging domains, the human operators are domain experts who can use domain-specific knowledge to assist the autonomous system when it deals with changing environments, uncertainty, and case-specific scenarios. Therefore, it is necessary to design tools and interfaces that enable humans to manage the autonomous behaviors of the system efficiently and effectively; such tools can improve task performance and the experience of the human operator in human-autonomy interaction. Wilderness Search and Rescue (WiSAR) is one such domain that could benefit from autonomy management tools when a mini-UAV (Unmanned Aerial Vehicle) is used in search.

Camera-equipped mini-UAVs can be useful tools in WiSAR operations by providing aerial imagery of a search area with quick coverage of large areas, access to hard-to-reach areas, and lower cost than manned aircraft [33, 21]. UAV path-planning is important because a good path can increase the probability of finding a missing person by making efficient use of the limited flying time. Various algorithms have been developed to support UAV path-planning (e.g., [5, 29, 31]), but the question remains how best to incorporate searcher expertise in such a way that the UAV path-planning is as efficient and effective as possible without requiring the searcher to understand how the autonomy works "behind the scenes."

We propose a new autonomy management approach where the human can influence the behavior of an autonomous system along two new dimensions: 1) **Spatial Constraints**: add constraints or priorities to different spatial regions, thereby affecting how the robot plans and performs its task; and 2) **Temporal Constraints**: impose time limits for a subtask or impose ordering constraints on a subtask. We refer to this approach as *Sliding Autonomy* because, properly designed, it can allocate degrees of authority and flexibility to the robot's algorithms by adding or removing constraints. Indeed, we will explicitly use a **slider** as one GUI tool for managing UAV path-planning.

As the human adds priorities or changes constraints, the sliding autonomy tool shows instant feedback on how those changes influence the UAV's plan, which enables the searcher to perform "what-if" analysis and see how the action changes autonomous behavior. This interactive approach lets autonomous algorithms perform tasks that algorithms are good at and humans do tasks that humans are good at, but in a collaborative and interactive way that avoids the pitfalls of simple task allocation [41, 9, 25]. Properly done, the human-robot team should perform better than a human or robot working alone.

Many approaches to autonomy management already exist and are called different things, such as *supervisory control* [41], *mixed-initiative* [23], *collaborative control* [20], *adjustable autonomy* [17, 18] (also referred to as *sliding autonomy* [16] or *adaptive automation* [39, 26]). The approach we propose falls under the category of *adjustable autonomy.* The dimensions we identified are in addition to dimensions

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*
Copyright © 2015, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

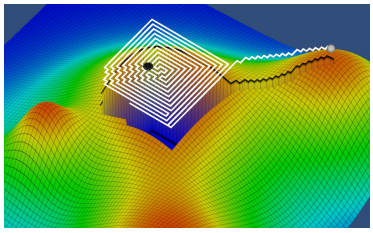

Figure 1: Sliding autonomy tool showing a 20-minute path segment with a probability distribution map (3D surface), the startpoint (UAV icon), and the endpoint (sphere on the right).

of *adjustable autonomy* identified by Bradshaw et al. so we design tools and algorithms that operate in a particular place in Bradshaw's taxonomy [8].

Previous work [32] organized the challenges of autonomy and management tool design along two dimensions: *attributes of an intelligent system* (capability, information management, performance evaluation) and *organizational scale* (individual versus group). In this paper we extend the guidelines to include attributes needed when a human-autonomy work collaboratively, and analyze how our proposed sliding autonomy approach fits in the guidelines. By applying sliding autonomy to the UAV path-planning task, we argue that this approach:

- enables the domain expert user to incorporate information only available to or understandable by the expert;
- is easy to understand;
- lets the human do what the human is good at (planning strategically and balancing performance tradeoffs) and autonomy do what autonomy is good at (planning tactically), resulting in better performance than human or autonomy working alone; and
- improves the human's experience in the human-autonomy interaction.

To evaluate the usefulness of the proposed approach, we performed a user study and compared the sliding autonomy method against two other planning methods (manual and simple pattern path-planning) in two WiSAR scenarios (one synthetic and one real). We measured each user's performance with each method on both the primary and secondary tasks. Experiment results show that the sliding autonomy method performed significantly better than the manual or simple pattern planning methods with no increased mental workload. *The performance of the human-autonomy team outperforms either human or autonomy working alone.*

2. AUTONOMY DESIGN GUIDELINES AND NEW DIMENSIONS

In this paper we extend the taxonomy described in [32] by adding attributes needed when multiple agents work collaboratively, and add new dimensions for sliding autonomy.

2.1 Autonomy Design Guidelines

When a human-autonomy team works, the autonomous component needs to provide interfaces so the human can interactively influence the autonomous behavior (**Interactivity**); the human should be able to manage how autonomy works in order to jointly find a solution by utilizing information only available to the human and/or feed information to autonomy in a representation that the autonomy can understand (**Manageability**); and when performance is evaluated, the human operator can judge whether the individual goal aligns with the collective goal of the system [25].

This paper focuses on intelligence of collaborative agents in a path-planning problem. Our path planner interface is designed to accept human input. The human can incorporate information from various sources and influence the behavior of path-planning autonomy by allocating degrees of authority and flexibility.

2.2 Spatial Constraints

The searcher should be able to influence the behavior of path-planning autonomy by setting/changing spatial constraints, which can be in the forms of start/end points of the path segment or task-specific zones (see Section 3).

Setting an endpoint in an area is a way for the searcher to indirectly tell path-planning autonomy that the area should have higher priority. For example, if a piece of clothing is found by the ground team, the searcher can force the path-planning autonomy to go visit that area by setting an endpoint there. Because the UAV must allocate part of the fixed-length flight time to reach that area, areas that had good payoffs before can become relatively costly and, therefore, no longer attractive to path-planning autonomy. Importantly, since we have assumed a fixed flight duration, setting an endpoint not only directly causes the UAV to focus search effort around that location but also indirectly causes the UAV to avoid other areas because the budget does not allow them to be searched well. In Figure 1, the UAV icon in the middle indicates the start point of the path segment and the sphere on the right indicates the desired endpoint.

In the GUI, the endpoint can be dragged around the search region and path-planning autonomy suggests different paths accordingly. This capability enables the user to adjust how much freedom is granted to autonomy. When the endpoint is close to the start point, autonomy has greater authority and flexibility in creating paths. If the endpoint is far from the start point, authority and flexibility for autonomy is reduced because a part of path-planning is moving the UAV toward the endpoint with the shortest path.

The interactive ability to move the endpoint around and see immediately how the change affects the UAV path recommended by autonomy enables the user to perform "what-if" analysis and lets the user see the causal effect between his/her action and changes in autonomous behavior.

Spatial constraints are easy to understand, so the searcher can interpret how these constraints will affect the behavior of path-planning autonomy. By managing autonomy along this dimension, the searcher can incorporate additional information (e.g., missing person profile) into the path-planning task, improve task performance, and align the task goal with the overall goal of finding the missing person quickly.

In our user study we fixed the start point of the path to the center of the map (last know position of the missing person). The searcher can set the endpoint for the current path segment anywhere on the map, and this endpoint automatically becomes the start point for the next path segment. We disabled the ability to move an endpoint once a path segment is planned to reduce computation, but we let users reset an entire plan, effectually allowing them to try different combinations of starting and ending path segments.

2.3 Temporal Constraints

In the UAV path-planning problem, temporal constraints include a *time limit* for a subtask (path segment), *subtask ordering*, and *valid time window* (see Section 3).

With the time limit constraint, the searcher can decide on how much flight time to allocate to a path segment out of the total flight time. This enables the searcher to break the path-planning task into multiple subtasks and then plan each path segment separately. In our interface design we let the searcher control time allocation using a slider; as the searcher moves the slider, the path-planning autonomy shows how the suggested path segment changes. Similar to the spatial constraints, this instant feedback enables "what-if" analysis on the causal effect between searcher action and changes in autonomous behavior.

For example, for a 60-minute total flight with an endpoint set to the probability hill on the right (Figure 1), the searcher can move the slider to set time limits and see immediately what path segment the autonomy would suggest. The path segment shown is when 20 minutes are allocated. If the searcher is happy with the suggestion, he or she approves the path segment. The UAV moves toward the endpoint and "vacuums up" the probability along the path (how much can be vacuumed up is determined by the task-difficulty map). Then the searcher works with the autonomy to plan the path for the remaining 40 minutes. The two (or more) path segments are joined to form the final path.

By managing autonomy along the temporal constraint dimension, the searcher can break the path-planning task into subtasks and incorporate additional information into the path-planning task. The user study described in this paper includes the time limit constraint.

In the example shown in Figure 1, the path covers the middle area well before moving right. If no endpoint is set, autonomy might decide to move left, instead. Less time allocation forces autonomy to focus more on the local area; more time allocation increase authority, so autonomy has more flexibility deciding what areas to cover. Instant feedback on path changes when constraints change lets the searcher interactively review multiple options and select the path segment that fits best with his/her strategic planning. This design theoretically enables the human to plan more strategically (prioritizing areas in the entire region) while autonomy works more tactically (covering the current area well), using strengths of each when they work collaboratively. Ideally such a human-autonomy team should work better than either human or autonomy working alone.

3. RELATED WORK

Drucker defines automation as a "concept of the organization of work [19]." Goodrich and Schultz define the HRI problem as "understanding and shaping the interactions between one or more humans and one or more robots" [22]. In their 1978 seminal paper, Sheridan and Verplank propose the idea of a *level of autonomy* spectrum [42], which Parasuraman et al. extended to four different broad functions [35]. Autonomy management approaches include top-down philosophy of *supervisory control* [41], *mixed-initiative* [23] where tasks can dynamically shift when necessary, *collaborative control*, a robot-centric model [20], where the robot is treated as a peer, and *Adjustable autonomy* [18] (also referred to as *sliding autonomy* [16] or *adaptive automation* [39]) that enables the human-automation team to dynamically and adaptively allocate functions and tasks among team members. This paper proposes a new variation of *sliding autonomy* useful for planning problems over a spatial region.

Many implementations of adjustable autonomy exist. Dorais et al. discuss a framework for human-centered autonomous systems for a manned Mars mission [17] that enables users to interact with these systems at an appropriate level of control but minimize the necessity for interaction. Kaber et al. describe an experiment simulating an air traffic control task where manual control was compared to Adaptive Automation (AA) [27]. Results suggest that AA is superior to completely manual control. Adjustable autonomy can also be applied to teams of agents and humans. Tambe et al. introduced the notion of a transfer of control strategy to adjustable autonomy and evaluated the approach in a real-world, deployed multi-agent system [44]. Cummings et al. investigated the impact of automated planning rates for single operator control of multiple unmanned vehicles in a decentralized network and conclude that rapid replanning can cause high operator workload, ultimately resulting in poorer overall system performance [15]. Alan et al. carried out a field trial to study notions of *flexible autonomy* in the context of tariff switching [1].

The human is an integral part of the human-autonomy team. Bainbridge points out that automation requires the human operator to take additional management responsibilities [2], and Sartar identified two automation management policies: *management by consent* and *management by exception* [40]. For complex automation, the human tends to rely on his/her *mental models* [34] to manage the system.

Many adjustable autonomy design guidelines have been proposed. Dias et al. identified six key capabilities for overcoming challenges in enabling sliding autonomy in peer-to-peer human-robot teams [16]. Bradshaw et al. discuss principles and pitfalls of adjustable autonomy and human-centered teamwork, and then present study results on so-called "work practice modeling" and human-agent collaboration in space applications [10]. Bradshaw et al. propose two dimensions of Adjustable Autonomy (descriptive and prescriptive) to address the two senses of autonomy (self-sufficiency and self-directedness) [8]. They also summarized some widespread misconceptions on autonomy and listed seven deadly myths of "autonomous systems" [9, 25].

UAV technology has emerged as a promising tool in supporting WiSAR [33, 5]. The goal of our research is to support fielded missions in the spirit of Murphy's work [11]. Many path-planning algorithms in the literature address obstacle avoidance while planning a path to reach a destination using A* [37], LRTA* [24], D* [43], Voronoi diagrams [4, 3], or probability roadmaps and rapidly-exploring random tree (RRTs) [36]. Bourgault et al. [7, 6] describe how to use a Bayesian model to create paths for a single UAV or multiple coordinated UAVs to maximize the amount of probability accumulated by the UAV sensors. Clark and Goodrich present in [14] an algorithm where a user can incorporate spatial constraints such as task-specific zones (*no-fly zone*, *coverage zone*, a *sampling zone*) and temporal constraints such as subtask ordering and valid time window to path planning tasks. The algorithms we used in this paper are algorithms from [29, 31], which significantly outperform [6].

4. USER STUDY AND EVALUATION

We performed a user study to evaluate the usefulness of several aspects of the sliding autonomy approach. It follows a 2×3 within-subject design with 2 scenarios (easy vs. difficult) and 3 path-planning methods (manual, pattern, and sliding autonomy). All participants completed all 6 exercises. The order of the scenarios and planning methods was counterbalanced to reduce learning effect. We recruited a total of 26 college students (14 males and 12 females) between the age of 19 and 30 (average 22.89).

We evaluate the following hypotheses:

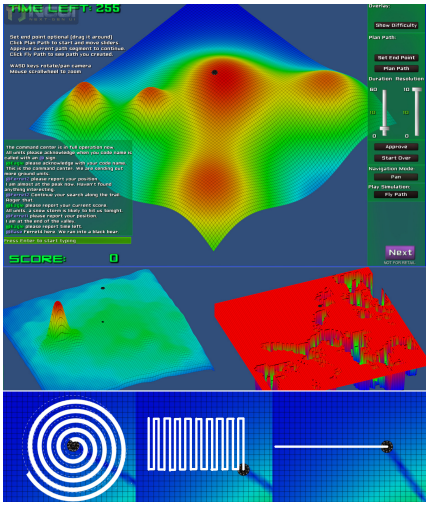

Figure 2: Top: Simulation interface with the sliding autonomy method showing the probability distribution map for scenario 1. Middle left: Probability distribution map for scenario 2. Middle Right: Task-difficulty map for scenario. Bottom: The three patterns with the pattern planning method.

H1: The sliding autonomy method performs better than either a manual method and a semi-autonomous method that uses standard search patterns to cover an area.

H2: The sliding autonomy method performs better than autonomy working alone.

H3: The sliding autonomy method does not increase the mental workload of the operator when compared against the manual and pattern methods.

4.1 Primary and Secondary Tasks

Each participant is given a probability distribution map of where are likely places to find the missing person and a task-difficulty map representing detection difficulty in different areas of the search region. The primary task is to plan a path for the UAV with fixed flight time to maximize the probability of finding the missing person.

In each exercise, each participant also performed a secondary task. This allows us to measure mental workload. In a group chat window (see Figure 2 top) when the user's code name appeared, the user had to type answers to simple questions. Roughly every 3 seconds a message was sent to the chat window, and every 5th message asked the user a simple question (4 per minute). The number of questions and the frequency remain the same across all conditions. For the same scenario and the same planning method, all users received the same set of chat messages.

We chose to use a group chat window as the secondary task because this is typical in WiSAR operations. We designed the chat messages to simulate a real WiSAR search for ecological validity. The user was asked to acknowledge connection and report path-planning status periodically.

After the demographic survey, each participant completed four 5-minute long non-skippable training sessions (one for each planning method with no task-difficulty map, and one for the manual method with a task-difficulty map) and then completed the 6 exercises. For each exercise, the participant has 5 minutes to plan the path. If the participant was happy with the path generated, he/she could finish the exercise early. We chose this design because we did not want the user to put all effort into completing the secondary task once he/she considers the primary task completed, which would skew the measurements on secondary task performance. At the end of each exercise, the participant completed a partial NASA TLX survey. Then at the very end of the user study, the participant filled out a survey describing his/her subjective preference with the three planning methods.

4.2 Simulation Environment

The user study was conducted in a 3D simulation environment (see Figure 2) where both the probability distribution map and the task-difficulty map were displayed as 3D surfaces with a color map (red means high and blue means low). The user could switch between the two maps at any time and rotate/pan/zoom a map at will. The UAV was a hexacopter capable of flying in all directions or hovering in the same spot. The UAV start point is set at the center of the search region because that was the Last Known Position (LKP) for the missing person.

With the **manual** planning method, the user flew the UAV with the arrow keys in a sped-up fashion so he/she can plan each 60-minute flight in two minutes. The user could switch between two flying modes (turn and strafe) and four camera views (global, behind, bird's eye, and free form). The user could also pause/resume the flight for the secondary task or better planning.

With the **pattern** planning method, the user chose from spiral, lawnmower, and line patterns (see Figure 2 bottom) and joined them to form the final path. The endpoint of the previous path segment (LKP if at the very beginning) automatically became the start point of the current selected pattern. As the user moved the cursor around, the size of the pattern changed with the cursor position marking the endpoint of the pattern. The start/end points pair determined the radius of the spiral pattern, the diagonal of the rectangle for the lawnmower pattern, and the start/end points for the line pattern. Once the user was happy with the location, shape, and size of the pattern, he/she could approve the pattern with a left click. The user could also undo the last path segment (pattern) planned. This planning method was "semi-autonomous" because the patterns were generated automatically without manually setting waypoints.

With the **sliding autonomy** method (see Figure 2 top), the user can set an endpoint (optional), and then drag the left slider to change the amount of time allocated to autonomy. The path suggested by the autonomy changed as the slider moved. The slider's max value always reflected the remaining flight time (in minutes). If the user was happy with the current path segment, he/she could approve it, the UAV moved to the end of the path segment, and the process repeated until a path had been planned that accounted for all of the available flight duration. The path-planning algorithm used was the LHC-GW-CONV algorithm [29] because it is the fastest algorithm and produces superior sub-optimal performance when compared to other state-of-the-art algorithms for the problem.

With all three planning methods, the user could choose to start over at any time during the exercise, and could restart multiple times within the total time given. We recorded the best path out of all tries.

4.3 Scenarios

The user study contained two WiSAR scenarios, a synthetic case (see Figure 2 top) with no task-difficulty map (assuming uniform detection probability), and a real WiSAR scenario (see Figure 2 middle) with a task-difficulty map, in which an elderly couple was reported missing near the

Table 1: Comparing across planning methods (SE stands for standard error)

	M	P	SA	SE	Significance
% Score	59.40	72.75	94.60	1.39	$\boldsymbol{F[2,50]=223.03, p<.0001}$
Time spent	243.35	240.02	228.37	12.06	$F[2,50]=1.16, p=.32$
Try count	1.75	3.56	3.31	0.43	$F[2,50]=9.47, p=.0003$
Clicks/try	13.01	35.64	25.58	2.90	$\boldsymbol{F[2,50]=19.47, p<.0001}$
NASA TLX	61.51	49.18	48.86	2.81	$\boldsymbol{F[2,50]=14.15, p<.0001}$
% Q. missed	52.94	56.69	55.04	5.17	$F[2,50]=1.26, p=.29$
Chat latency	10.39	11.17	10.92	0.65	$F[2,50]=0.46, p=.63$

Table 2: Percent of participants outperforming autonomy with each method for each scenario

	Manual (M)	Pattern (P)	Sliding Autonomy (SA)
Sce. 1 (Low)	0%	0%	**88.46%**
Sce. 2 (High)	0%	19.23%	**92.21%**

Grayson Highlands State Park in Virginia [28][1]. Scenario 2 is clearly more complex than scenario 1 because the user also had to consider the different detection probability defined by the task-difficulty map. We refer to scenario 2 as the high-information scenario and scenario 1 as the low-information scenario. These two scenarios exhibited significantly different amounts of workload in a pilot study and gave us confidence that the results scale to different types of scenarios.

4.4 Measures

Five metrics were used for the primary path-planning task:

- **Percent score**: Computed by summing the amount of probability collected by the UAV if it followed the path planned. The user's best score for each exercise (multiple tries) was normalized by dividing the best score from all users for the same scenario. This way we could compare planning methods across scenarios.
- **Time spent**: Time spent with each exercise.
- **Try count**: How many tries in each exercise. Since the manual planning method takes much longer to plan a path than the other two methods by design, this measurement is used mainly to compare the pattern method and sliding autonomy method.
- **Mouse clicks per try**: Left mouse clicks within a try. This measurement compares pattern and sliding autonomy methods because the manual method does not require a lot of mouse clicks by design.
- **NASA-TLX raw score**: The sum of the user subjective evaluation of cognitive workload in six dimensions normalized to a 100-point scale.

Two metrics were used for the secondary task:

- **Percentage of questions missed**: What percentage of questions directed to the user were missed before the user completed the exercise? Here we did not measure the percentage of questions answered correctly because all the questions are very simple and all users answered the questions correctly.
- **Chat latency**: The number of seconds between the time a question was presented to the user and the time when the user answered the question.

[1]The probability distribution map used (Figure 2 middle left) was generated using a Bayesian model [30]. The map has been evaluated at George Mason University's MapScore web portal [12] and performed better than most other models evaluated, scoring 0.8184 on a [-1,1] scale where the higher the score the better. http://sarbayes.org/projects/. The task-difficulty map (Figure 2 middle right) was generated using vegetation density data downloaded from the USGS web site and categorized into three difficulty levels (sparse, medium, and dense, with detection probability of 100%, 66.67%, and 33.33% respectively).

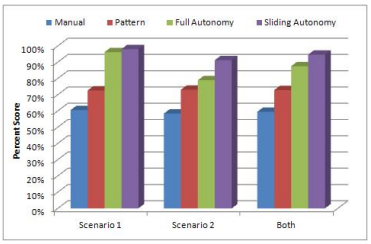

Figure 3: Three methods performance differences.

5. RESULTS AND ANALYSIS

We analyzed the user study data with a mixed measures analysis of variance (ANOVA) and report results below.

5.1 Comparing Across Scenarios

To reduce learning effect, half of the participants started on scenario 1 first and vise versa. Mouse clicks per try for the two scenarios are significantly different ($F[1,25] = 28.65, p < .0001$) indicating scenario 2 required participants to be more active than in scenario 1. This result supports observations in the pilot study that scenario 2 imposed higher workload on participants than scenario 1. User activity logs show that participants created more path segments (for pattern and sliding autonomy planning methods) in scenario 2 than scenario 1.

NASA TLX scores are also significantly different between the two scenarios ($F[1,25] = 31.35, p < .0001$). The average score difference is 9.98 (out of a total of 100 points), almost a full "pip" on the TLX survey, indicating that on average each participant felt his/her cognitive workload was much higher in the high-information scenario.

The percentage of questions missed is almost identical between scenarios (54.88% and 54.90%), and the chat latency is also very close (10.39 and 11.17 seconds). This shows that participants on average performed about the same with the secondary task across scenarios. No statistically significant differences were found across scenarios for percent score, time spent, and try count.

5.2 Comparing Across Planning Methods

For each scenario, three path-planning methods were used (manual, pattern, and sliding autonomy). Order of the methods was randomly drawn without replacement from all permutations of method order to reduce learning effect. For example, no statistically significant difference in performance is found when comparing participants who used pattern first to those using sliding autonomy first ($F[1,6] = 0.0030, p = 0.9567$). Table 1 lists comparison among these three methods.

Percent score differences across methods are statistically significant ($F[2,50] = 223.03, p < .0001$) with sliding autonomy (94.60%) performing better than pattern (72.75%) and manual (59.40%). The difference is also significant when comparing between sliding autonomy and pattern ($F[1,25] = 53.32, p < .0001$). As shown in Figure 3, this trend is clear in both scenario 1 and 2 (high vs. low-information) individually, suggesting some robustness of the result across a range of scenarios. Therefore, user study results support our first hypothesis: the sliding autonomy method performs better than either the manual method or the pattern method.

Statistically significant differences ($F[2,50] = 19.47, p < .0001$) were also found in mouse clicks per try (starting over means having another try). The manual method uses arrow keys to fly the UAV around and only uses mouse clicks when

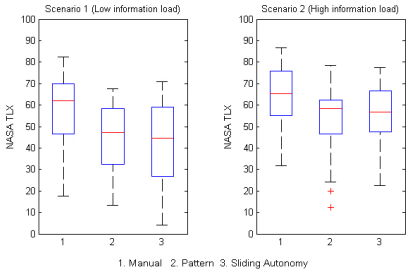

Figure 4: Box plots of the NASA TLX scores for each scenario.

switching camera modes or stopping the timer in order to perform the secondary task. By design, this method does not use a lot of mouse clicks. Pattern and sliding autonomy methods both use mouse clicks for the actual path-planning task, and the pattern method clearly generated more mouse clicks per try (35.64) than the sliding autonomy method (25.58). Two factors might have contributed to this difference: First, the pattern method allowed a participant to "undo" a path segment (in additional to reset and start over) whereas sliding autonomy did not allow this. Second, sliding autonomy allowed a participant to drag a slider, which produced different suggested paths; this accomplishes the same type of "what if" interaction as "undo" in the pattern method, but required fewer mouse clicks.

It is informative to compare these interactive planning methods with a fully autonomous path. This is useful because, due to the computational complexity of the planning problem, only suboptimal solutions can be generated by the planning algorithms. Completely autonomous path-planning (without human input) produces paths with a score of 96.13% for scenario 1 and 78.33% for scenario 2. It is instructive to compare these values to those produced by the different planning methods in the different scenarios (see Figure 3). This places the performance of full autonomy ahead of manual and pattern planning methods but behind sliding autonomy in both scenarios. This indicates that the sliding autonomy approach outperforms manual, pattern, and full autonomy approaches to the problem.

As shown in Table 2, for scenario 1, no participants were able to outperform full autonomy using manual or pattern approaches, but 23 of 26 participants (88.46%) were able to outperform full autonomy using sliding autonomy. For scenario 2, no participants were able to outperform full autonomy using manual control, but 5 of 26 participants (19.23%) and 24 of 26 participants (92.21%) were able to outperform full autonomy using pattern and sliding autonomy, respectively. Thus, results of the study support the second hypothesis: sliding autonomy methods perform better than a fully autonomous approach given state-of-the-art planning algorithms for this problem. The full autonomy we refer to here is the specific path-planning algorithm we used in the user study (LHC-GW-CONV) [29]. In Section 6.2, we discuss how the sliding autonomy approach compares to other path-planning algorithms.

NASA TLX raw scores show significant differences ($F[2, 50] = 14.15, p < .0001$) among the three methods, with the manual method showing the highest cognitive mental workload (61.51), a full "pip" more than the other two methods on the TLX survey. The average score difference between the pattern method and the sliding autonomy method is not significantly different. Figure 4 shows the box plots of the NASA TLX scores for each scenario.

For all three planning methods, participants performed about the same on the secondary task, as shown by percentage of questions missed and chat latency in Table 1. Combining this with percent score and NASA TLX we can conclude that sliding autonomy performed best without increasing participants' mental workload, which supports our third hypothesis: the sliding autonomy method does not increase the mental workload of the operator when compared against manual and pattern methods.

5.3 Additional Factors

We also performed ANOVA analysis on additional factors that might create differences: gender, experience in video games, order of the scenarios, and whether participants used full autonomy with the sliding autonomy method. No significant differences were found for these factors overall, across scenarios, or across methods. There is also no significant correlation (-0.23) between percentage of questions missed in the secondary task and the NASA TLX raw scores.

6. DISCUSSION

In this section we present observations from the user study and discuss possible explanations for the user behaviors.

6.1 Planning Methods Characteristics

Manual Method

With the manual method, the participant uses arrow keys to move the UAV around to create a path, so by design the method is very intuitive, flexible, and participants were required to plan each 60-minute flight time in two minutes. This allowed them at least two attempts during the five minutes allotted to complete the task. Many participants reported that the arrow keys were too "sensitive" and recommended slowing down the UAV.

In practice the time pressure made it too costly to start over. Although it is possible to pause the simulation to allow for participants to plan, participants reported that they did not feel that they had the the luxury to do so. Naturally, when this continuous process is interrupted by the secondary task where the participant has to pause planning and answer questions in the group chat window, frustration is high.

More physical work, higher frustration, and lower performance score are the main factors contributing to a much higher NASA TLX score for the manual method. During training, participants had one extra session with the manual method, but this method still performed the worst.

Pattern Method

With the pattern method, the participant joins a mixture of three patterns (spiral, lawnmower, and line) together to form the final path. This is more of an episodic process, so it is very easy to pause in the middle of the planning and shift attention to the secondary task. There is also less time pressure because the participant can quickly plan for the remaining time with just one big spiral (or lawnmower) in one click. Therefore, the participant has plenty of time for many tries with different strategies.

In the post-user study survey, many participants commented that with the spiral and lawnmower pattern it is really easy to run out of flight time. They suggested adding the ability to allocate time to the patterns similar to the sliding autonomy method. This means that with this method, a participant enjoys the systematic coverage of an area but has a hard time estimating how much time it takes the UAV to cover the area following the pattern. Several participants also suggested adding more patterns to the method.

The pattern method is the only one that allows a participant to "undo" a plan. This ability increased the number of mouse clicks per try for the pattern method, and likely made the participants' feel that the pattern method was easier compared to the other two. Although undo doesn't make sense in manual method, making the function available in sliding autonomy might have mitigated the effect. Another interesting observation is that participants seemed to be overly optimistic about their performance using the pattern method. For example, although sliding autonomy created better paths than pattern in all scenarios for all participants, 46.15% of participants (as measured in the NASA TLX with the performance dimension) and 26.92% (as reported in the post study survey) reported that the pattern method created best paths.

Sliding Autonomy Method

Similar to the pattern method, the sliding autonomy method is also episodic. Therefore, stopping in the middle of the planning to answer questions for the secondary task was easy. Since it only takes a few clicks to let autonomy plan a path for the remaining time, there is not much time pressure and the participant can have many tries.

Because the participant does not know how the autonomy works behind the scenes, many participants were surprised by the path recommended by autonomy and felt that autonomy did not do what they wanted it to do. For example, when a participant sets the endpoint in region A, autonomy might plan a path that spends most of its time in a seemingly unrelated region B and only goes toward region A at the end of the path, because such a path is more efficient (scores higher). In such cases, the slider becomes the only tool that lets the participant "force" autonomy to do what the participant wants, and path-planning turns into a fight between the human and autonomy. However, the instant feedback (displaying the path and the predicted "vacuuming effect") does help the participant figure out why autonomy would suggest something different, and some participants were glad that autonomy suggested better paths they had not considered.

Most participants were generally happy with the path segment recommended by autonomy covering a local region, even when the region is in an irregular shape (not a circle or rectangle). Many participants also expressed that they did not have enough control over the path generation and recommended adding the ability to include constraints such as "middle points" where the path segment has to go through these middle points. In fact, such "middle points" can already be achieved with the current method by setting multiple endpoints, effectually creating a multi-segment approach. Several participants complained that this method does not have the undo function. This is an oversight on our end. Experiment results show that sliding autonomy still performed the best even without the undo function. Therefore, the most important conclusions from the experiment are not invalidated by this. With both the pattern and sliding autonomy methods, many participants expressed the desire to be able to modify the path after it is generated.

User Preferences

In scenario 2 where a task-difficulty map was used, most participants switched between map views. The probability map used is similar to a uni-modal distribution (see Figure 2 middle left). For the first part of the planning, they viewed the probability distribution map and "covered" the high mode. They then switched the view to the task-difficulty map for the remaining time, only occasionally switching back to the probability distribution map view. This pattern of behavior was seen in each of the three planning methods. Some participants suggested showing both maps side by side or have a way to combine the two maps into one. These ideas are worth exploring in future user studies.

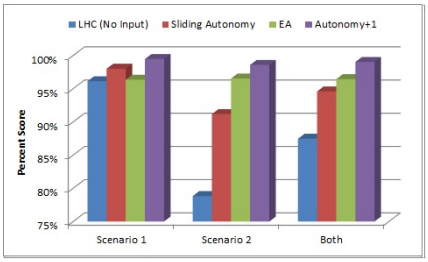

Figure 5: Comparing sliding autonomy performance against various markers.

In the post-user study survey, the majority of the participants think manual is the easiest to learn (53.85%), pattern is the easiest to use (57.69%), and sliding autonomy performed the best (65.38%). However, most participants preferred the pattern method (69.23%) out of all three. We believe the lack of undo function and operator-induced oscillation when moving the slider had negative impacts on participants' opinion of the sliding autonomy method. This is relevant for the design of future sliding autonomy systems, suggesting that some combination of pattern-based planning and sliding autonomy, augmented with the ability to undo decisions and flexibly alter or constrain paths, will produce a high-performing GUI with high user acceptance.

6.2 Reliance on Autonomy

We have claimed that a human interacting with an autonomous algorithm via sliding autonomy outperforms full autonomy, but this claim naturally depends on the quality of the autonomous algorithm. The algorithm we used was selected from a comparison of various algorithms in prior work [29, 31] because it worked in real-time and produced high-quality paths, but there exist other algorithms that produce higher-quality paths if we allow more time for path-planning. It is useful to compare performance of the sliding autonomy algorithm with these other algorithms.

As a basis for comparison, we consider an evolutionary algorithm (EA) that takes the output of several real-time planning algorithms, including the one we used in the user study, as seeds for the evolutionary process. Thus, the EA approach takes high-quality solutions and then adds further optimizations. As shown in Figure 5, when such optimization is applied to scenario 1 and scenario 2, the optimization produces a path that is only slightly worse than sliding autonomy for scenario 1 and a path that is much better than sliding autonomy in scenario 2. This suggests that better path-planning might be more important than interactive path-planning.

Because we are arguing that human-plus-autonomy is better than either alone, we explored how the number of human inputs can affect the output of the sliding autonomy approach. Results indicate that the sliding autonomy algorithm we used in the user study can generate high-quality paths with only one point of human input (specifying an endpoint). We call this approach the *Autonomy+1* approach. Note that we arbitrarily picked one point to represent minimal human input. This does not imply that Autonomy+2 or Autonomy+3 will perform worse.

Table 3: Percentage of participants outperforming autonomy performance markers

	Full Autonomy	EA	Autonomy+1
Scenario 1 (Low)	**88.46%**	**88.46%**	7.69%
Scenario 2 (High)	**92.21%**	26.92%	15.38%

Using the best score out of 3 tries (roughly equal to the average number of tries in the user study), we computed the percent score for this Autonomy+1 human input approach: 99.47% for scenario 1 and 98.58% for scenario 2. Using the EA and Autonomy+1 scores as additional markers, we plotted participants' average performance in each scenario against these markers. Figure 5 shows the result.

First, sliding autonomy (human-autonomy team) outperformed nominal autonomy in both scenarios. Sliding autonomy also outperformed EA in scenario 1 (low-information). In scenario 2, the performance of sliding autonomy is not very far from EA (5.36%), and the difference is even smaller (1.85%) when averaged over both scenarios. However, the most interesting observation is that Autonomy+1 actually outperformed all others in both scenarios (99.47% for scenario 1 and 98.58% for scenario 2). Although a few participants did score higher than Autonomy+1, the difference is less than 1.5%. Table 3 lists what percentage of participants outperformed full autonomy, EA, and Autonomy+1.

What Figure 5 suggests is that the sliding autonomy method does not need a lot of human input to perform really well. Instead of spending the effort creating many path segments and setting many endpoints, it may be more effective to search in the right region by setting just a few constraints. However, 88.68% of the participants gave more than 1 input when they used sliding autonomy (81.13% for 2 inputs and 69.81% for 3 inputs). In the post-user study survey, only 46.15% of the participants acknowledged trying full autonomy with the sliding autonomy method: they did not specify any endpoints and simply relied on the autonomous path-planner to do all the planning. When using the sliding autonomy method, a good strategy is to start with full autonomy (as the worst scenario) and then see how additional human input can improve the path, but this leads to questions of over- and under-reliance on autonomy [9].

6.3 Why a Human-Autonomy Team Performs Better?

User study results show that the human-autonomy team outperformed both human or autonomy working alone, but how were they able to achieve this? We hypothesize that this is because the sliding autonomy approach enabled the human to focus on what the human is good at and autonomy to focus on what autonomy is good at. Bradshaw et al. point out [9]: "Humans, though fallible, are functionally rich in reasoning strategies and their powers of observation, learning, and sensitivity to context." Our observation suggests that a human may be better equipped than autonomy to think strategically and to recognize bad path segments.

The sliding autonomy method lets the user plan at a higher abstract level by specifying priorities in search subregions and how well each sub-region should be covered. Autonomy, on the other hand, can generate a path that covers a sub-region (or some nearby sub-regions) precisely and quickly, and can handle all kinds of irregular sub-region shapes. Therefore, the sliding autonomy method combines the strengths of both human and autonomy.

Observations from the user study suggest that humans are very good at recognizing bad moves in solutions suggested by path-planning autonomy. The sliding autonomy approach enables the human to select from a bunch of suggested paths.

6.4 Why Similar Secondary Task Performance in All Three Methods?

The pattern and sliding autonomy methods are episodic, suggesting that it is easier for the user to pause planning and shift attention to the secondary task of answering questions in the group chat window. However, user study data show that there is no significant difference in secondary task performance across all three path-planning methods.

The manual method requires a lot of continuous keyboard interaction (great physical demand and temporal demand) to move the UAV around. However, it does not actually require much mental demand and effort because the planning process is more sporadic and spontaneous. Observations show that if a mistake is made, because there is no way to correct it, the user quickly stops worrying about it and moves on. The low mental demand and effort make monitoring the group chat window an easy task, even though some users complained that switching back and forth between primary task and secondary task is very frustrating.

With the pattern and sliding autonomy methods, path-planning is more like piecing together a puzzle. The user appears to be deeply drawn into problem-solving, constantly comparing tradeoffs, which actually requires more mental involvement. With the sliding autonomy method, the user is interacting with complicated algorithms, so while planning a path, the user is also trying to build a mental model of how autonomy works. As a result, the user actually paid less attention to the secondary task. Fighting with autonomy when human and autonomy had disagreements also drew user attention away from the group chat window. But when the group chat window catches the user's attention, he/she can perform the secondary task leisurely.

7. CONCLUSIONS

We propose a new autonomy management approach, a variation of sliding autonomy, which lets the user influence the behavior of the autonomous system along two new dimensions: spatial constraints and temporal constraints. We present interface designs that let the user allocate degrees of authority and flexibility to the robot's algorithms through interactivities along these new dimensions. Experiment results show that the sliding autonomy method performs significantly better than either the manual or pattern path-planning method without increasing the user's mental workload, the human has a better interaction experience, and human-autonomy collaboration outperforms either human or autonomy working alone.

We used algorithms in [31] as seeds to the EA algorithm. These algorithms are our early attempts at incorporating strategic planning in autonomy. Further investigation on how improved strategic planning algorithms might change the dynamics of the human-autonomy collaboration is a natural extension of the present work.

Acknowledgments

This work was partially supported by the NSA under grant number 0534736 and 0812653, and under the Robotics Collaborative Technology Alliance supported by the ARL. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsoring organizations.

REFERENCES

[1] A. Alan, E. Costanza, J. Fischer, S. D. Ramchurn, T. Rodden, and N. R. Jennings. A field study of human-agent interaction for electricity tariff switching. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 965–972. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[2] L. Bainbridge. Ironies of automation. *Automatica*, 19(6):775–780, 1983.

[3] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. McLain, and M. Goodrich. Autonomous vehicle technologies for small fixed-wing UAVs. *AIAA Journal of Aerospace Computing, Information, and Communication*, 2(1):92–108, 2005.

[4] S. Bortoff. Path planning for UAVs. In *Proceedings of the American Control Conference*, volume 1, pages 364–368. IEEE, 2000.

[5] F. Bourgault, T. Furukawa, and H. Durrant-Whyte. Coordinated decentralized search for a lost target in a Bayesian world. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[6] F. Bourgault, T. Furukawa, and H. Durrant-Whyte. Optimal search for a lost target in a Bayesian world. In *Field and Service Robotics*, pages 209–222. Springer, 2006.

[7] F. Bourgault, A. Goktogan, T. Furukawa, and H. Durrant-Whyte. Coordinated search for a lost target in a Bayesian world. *Advanced Robotics*, 18(10):979–1000, 2004.

[8] J. Bradshaw, P. Feltovich, H. Jung, S. Kulkarni, W. Taysom, and A. Uszok. Dimensions of adjustable autonomy and mixed-initiative interaction. *Agents and Computational Autonomy*, pages 235–268, 2004.

[9] J. Bradshaw, R. Hoffman, M. Johnson, and D. Woods. The seven deadly myths of "autonomous systems". *Intelligent Systems, IEEE*, 28(3):54–61, 2013.

[10] J. Bradshaw, M. Sierhuis, A. Acquisti, P. Feltovich, R. Hoffman, R. Jeffers, D. Prescott, N. Suri, A. Uszok, and R. Van Hoof. Adjustable autonomy and human-agent teamwork in practice: An interim report on space applications. *Agent Autonomy*, pages 243–280, 2003.

[11] J. Casper and R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 33(3):367–385, 2003.

[12] E. Cawi, N. Jones, and C. R. Twardy. MapScore: Probability map evaluation for search & rescue. In *Virginia Search & Rescue Conference*. Conference Presentation presented at the Virginia Search & Rescue Conference, Holiday Lake, VA, 2012.

[13] A. Chun. Optimizing limousine service with AI. In *Proceedings of the Innovative Applications of Artificial Intelligence Conference*. AAAI, 2010.

[14] S. Clark and M. Goodrich. A hierarchical flight planner for sensor-driven UAV missions. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*, pages 509–514. IEEE, 2013.

[15] M. L. Cummings, A. Clare, and C. Hart. The role of human-automation consensus in multiple unmanned vehicle scheduling. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 2010.

[16] M. Dias, B. Kannan, B. Browning, E. Jones, B. Argall, M. Dias, M. Zinck, M. Veloso, and A. Stentz. Sliding autonomy for peer-to-peer human-robot teams. In *Proceedings of the Intelligent Conference on Intelligent Autonomous Systems*, page 332, 2008.

[17] G. Dorais, R. Bonasso, D. Kortenkamp, B. Pelland, and D. Schrechenghost. Adjustable autonomy for human-centered autonomous systems on Mars. In *The First International Conference of the Mars Society*, 1998.

[18] G. Dorais and D. Kortenkamp. Designing human-centered autonomous agents. In *Advances in Artificial Intelligence. PRICAI 2000 Workshop Reader*, pages 321–324. Springer, 2001.

[19] P. Drucker. *The Practice of Management*. Harper Paperbacks, NYC, NY, Oct. 2006.

[20] T. Fong, C. Thorpe, and C. Baur. Collaborative control: A robot-centric model for vehicle teleoperation. In *AAAI Spring Symposium: Agents with Adjustable Autonomy*, 1999.

[21] M. Goodrich, B. Morse, D. Gerhardt, J. Cooper, M. Quigley, J. Adams, and C. Humphrey. Supporting wilderness search and rescue using a camera-equipped miniUAV. *Journal of Field Robotics*, 25(1-2):89–110, 2008.

[22] M. Goodrich and A. Schultz. Human-robot interaction: A survey. *Foundations and Trends in Human-Computer Interaction*, 1(3):203–275, 2007.

[23] M. Hearst. Mixed-initiative interaction. *Intelligent Systems, IEEE*, 14(5):14–23, 1999.

[24] J. Howlett, T. McLain, and M. Goodrich. Learning Real-Time A* path planner for unmanned air vehicle target sensing. *Journal of Aerospace Computing, Information, and Communication*, 3(3):108–122, 2006.

[25] M. Johnson, J. M. Bradshaw, P. J. Feltovich, R. R. Hoffman, C. Jonker, B. van Riemsdijk, and M. Sierhuis. Beyond cooperative robotics: The central role of interdependence in coactive design. *Intelligent Systems, IEEE*, 26(3):81–88, 2011.

[26] D. Kaber, J. Riley, K. Tan, and M. Endsley. On the design of adaptive automation for complex systems. *International Journal of Cognitive Ergonomics*, 5(1):37–57, 2001.

[27] D. Kaber, M. Wright, L. Prinzel III, and M. Clamann. Adaptive automation of human-machine system information-processing functions. *Human Factors*, 47(4):730, 2005.

[28] R. Koester. *Lost Person Behavior: A search and rescue guide on where to look - for land, air and water*. dbS Productions LLC, Charlottesville, VA, August 2008.

[29] L. Lin and M. Goodrich. UAV intelligent path planning for wilderness search and rescue. In *Proceedings on the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 709–714, 2009.

[30] L. Lin and M. Goodrich. A Bayesian approach to modeling lost person behaviors based on terrain features in wilderness search and rescue. *Computational and Mathematical Organization Theory*, 16(3):300–323, 2010.

[31] L. Lin and M. Goodrich. Hierarchical heuristic search using a Gaussian Mixture Model for UAV coverage planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2014.

[32] L. Lin, M. Roscheck, M. Goodrich, and B. Morse. Supporting wilderness search and rescue with integrated intelligence: Autonomy and information at the right time and the right place. In *Proceedings of the AAAI Conference on Artificial Intelligence, Special Track on Integrated Intelligence*, 2010.

[33] R. Murphy, E. Steimle, C. Griffin, C. Cullins, M. Hall, and K. Pratt. Cooperative use of unmanned sea surface and micro aerial vehicles at hurricane Wilma. *Journal of Field Robotics*, 25(3):164–180, 2008.

[34] D. Norman. *Some Observations on Mental Models*. Lawrence Erlbaum Associates, Mahwah, NJ, 1983.

[35] R. Parasuraman, T. Sheridan, and C. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 30(3):286–297, 2000.

[36] P. Pettersson and P. Doherty. Probabilistic roadmap based path planning for an autonomous unmanned helicopter. *Journal of Intelligent and Fuzzy Systems*, 17(4):395–405, 2006.

[37] M. Quigley, B. Barber, S. Griffiths, and M. Goodrich. Towards real-world searching with fixed-wing mini-UAVs. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[38] B. Robins, K. Dautenhahn, and P. Dickerson. From isolation to communication: A case study evaluation of robot assisted play for children with autism with a minimally expressive humanoid robot. In *Proceedings of the International Conferences on Advances in Computer-Human Interactions*, pages 205 –211, 2009.

[39] W. Rouse. Adaptive aiding for human/computer control. *The Journal of the Human Factors and Ergonomics Society*, 30(4):431–443, 1988.

[40] N. Sarter. Making coordination effortless and invisible: The exploration of automation management strategies and implementations. In *CBR Workshop on Human Interaction with Automated Systems*, 1998.

[41] T. Sheridan. *Telerobotics, automation, and human supervisory control*. The MIT Press, Cambridge, MA, 1992.

[42] T. Sheridan and W. Verplank. Human and computer control of undersea teleoperators. Technical report, MIT, Cambridge, MA, 1978.

[43] A. Stentz. Optimal and efficient path planning for partially known environments. In *Intelligent Unmanned Ground Vehicles*, pages 203–220. Springer, 1997.

[44] M. Tambe, P. Scerri, and D. V. Pynadath. Adjustable autonomy for the real world. *Journal of Artificial Intelligence Research*, 17(1):171–228, 2002.

[45] D. Woods and E. Hollnagel. *Joint cognitive systems: Patterns in cognitive systems engineering*. CRC Press, Boca Raton, FL, 2006.