Assessing user specifications for robot task planning

Alexandru Blidaru, Stephen L. Smith and Dana Kulić

***Abstract*— As robots' capability and autonomy improve, they are expected to increasingly operate in human environments, and interact with novice, untrained users. When robots operate in human or shared environments, their tasks and behaviours need to be specified; this task is typically performed by a human operator or supervisor. The human operator may specify constraints on robot behaviour to make the robot more predictable or align its behaviour with user expectations. However, these constraints may impact robot task performance. This paper investigates how novice users generate robot specifications and proposes metrics for quantifying specification quality. The proposed approach is evaluated with a user study, where novice users provide specifications for an autonomous robot operating in a shared warehouse environment. We find that untrained users create a wide variety of behaviour-limiting specifications, that users generally have difficulty creating efficient specifications, and that they were not able to correctly assess their own performance.**

I. Introduction

Conventional robotic technology has been designed with highly trained users in mind. As robots achieve higher levels of autonomy in a variety of environments, robot designers can no longer assume a high level of expertise from all potential users, and instead they must create human robot interfaces that enable simple and effective communication between robots and untrained users.

Existing mobile autonomous robots are able to successfully plan and navigate in controlled environments [1]. Generally the robot planner is designed to find paths that minimize a certain cost function, usually the distance or time of travel. For a human working in the same environment as the robot, it might be difficult to easily understand and predict its behaviour. As a result, it is often preferred for the behaviour of the robot to be limited or structured in some manner. This reduces the complexity and number of actions that the robot can take at any given time, making its behaviour more predictable. Additionally, users might also expect the robot to follow existing conventions, such as road rules. These two factors (higher predictability and alignment of expectations) have been previously found to correlate to higher trust in robotic systems [2], [3], which in turn leads to increased usage [4], [3], [5] and increased effectiveness [6].

An easy way to constrain a robot's behaviour is to designate special-behaviour zones in the environment. The robot can then take these areas into consideration during planning, and follow the rules of the zone when inside it.

Before a robot can begin autonomous operation in a new environment, the desired zones need to be specified by human operators. This process however, can be challenging, especially for novice and untrained users. These users may not be familiar with the robot's capabilities, which can make it difficult for them to encode the expected robot behaviour. In addition, due to their inexperience, it can also be more difficult for these users to appreciate the impact of their specifications on the performance of the robotic system.

A motivating example, and the focus of our work, comes from industrial and warehouse robotics. In such environments, mobile robotic platforms are used to transport items. For example, line-following automated guided vehicles (AGVs) are used in a variety of manufacturing plants. However, AGV trajectories are rigidly defined by magnetic tape lines on the ground, and so it is hard and expensive to reconfigure a system once installed. Fully autonomous robotic platforms that are not limited to specific trajectories are a promising alternative. While AGVs only follow specific laid out paths, non-constrained mobile autonomous platforms can move anywhere in the open space and generate their paths autonomously. To ensure safety and predictability, these robots are expected to follow existing conventions, customs and rules. Examples include driving on the correct side of the road, stopping before intersections, avoiding certain areas, and following established directions of travel down narrow hallways and aisles. Operators tasked with managing the autonomous robot fleet should have the capability to easily create specifications that restrict the behaviour of the robot to enforce the desired processes and rules.

In this paper, we develop a set of metrics that allow us to evaluate and quantify the quality of user specifications. The proposed metrics are validated by applying them to several sets of user-created specifications, which were obtained through a user study where participants took the role of a mobile robot fleet manager and interacted with a simulated robotic system. Given the user specification on the environment map, a graph representing the environment and user constraints is obtained by encoding the constraints into a state lattice. The proposed metrics are applied to the generated motion graph. The metrics capture the positive (i.e. increased predictability of robot behaviour) and negative (i.e. non-optimal behaviour compared to an environment devoid of any restrictions) effects of a user specification. These metrics allow us to capture the trade-offs that users make in ensuring that the robot accomplishes its tasks while minimizing loss of performance. The ability to assess user

This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Otto Motors.

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, 200 University Ave W, Waterloo, Ontario, N2L 3G1, Canada (asblidar@uwaterloo.ca; stephen.smith@uwaterloo.ca; dana.kulic@uwaterloo.ca)

specifications could then be used to help users improve their specifications.

II. Related Work

Measures of robot performance in human-robot interaction scenarios depend on the task and type of interaction. Previous work has focused on developing frameworks and taxonomies based on the role of the human operator, and the type and extent of interaction between humans and machines, in order to group common elements of interaction and performance.

Scholtz [7] defines five models of human-robot interaction: supervisor, operator, mechanic, peer, and bystander. Of most interest to our work are the supervisor and bystander interactions; identifying the relevant interaction roles can aid in developing appropriate metrics. Someone working in a supervisory role needs to have an overview of the situation, the missions and tasks that are executed at any given time, and the current robot behaviour. On the other hand, as a bystander, it is important to understand the causes of robot behaviour, the range of behaviours, as well as predict what the robot will be doing next [7]. In our work, the supervisor role is undertaken by the user creating behavioural constraints for the robot, while bystanders interact with the robot in its workspace as it is accomplishing its task. In [8], Scholtz identifies four metrics related to the bystander role: predictability of behaviour, capability awareness, interaction awareness and user satisfaction.

The metrics for measuring human-robot interaction and performance also depend on the specific tasks to be accomplished. In [9], Steinfeld et al. propose a framework that organizes these metrics by breaking up a task into the underlying components and then comparing the human-robot interaction (HRI) of different applications and tasks. Steinfeld et al.'s framework describes these metrics through five task categories: navigation, perception, management, manipulation, and social. The ones relevant to our activities are navigation and social. As part of navigation, the authors proposed two subcategories of measurements: effectiveness-related, which measure how well the task is completed; and efficiency-related, which determine the time needed to complete the task. The suggested effectiveness measurements include percentage of navigation tasks successfully completed, area coverage and any deviations from the planned route. For efficiency, proposed measures include time to task completion, as well as operator time for the task. For social tasks, it is suggested that interaction characteristics such as style or context, persuasiveness, trust, engagement and compliance be measured in order to ensure a high level of success in social interactions.

In many cases, in order to achieve better performance, it might be of interest to place constraints on robot behaviour, which has the effect of altering the robot's operating environment. A relevant area of related work considers metrics of the complexity of an environment, which have been found to correlate with robot performance [10], [11]. Different methods of calculating environmental complexity have been suggested. Crandall and colleagues [10], [12], [13], [14] propose that complexity be computed by estimating the branching factor and amount of clutter in the robot environment. Another approach uses techniques based in information theory to determine the robot environment complexity. In [11], Lampe et al. propose measuring the entropy based on obstacle density to compute the global complexity of the environment. In [15], [16], Anderson and Yang use the number of accessible neighbors at every node to calculate the entropy of the environment. Additionally, in [16] a secondary complexity measure is proposed based on the distribution of obstacles and the size of the compressed description of obstacle distribution. More recent work, in [17], has extended the measurement of complexity from a binary function of local obstacles to a continuous function, which allows for the consideration of dynamic obstacles.

For sufficiently complicated tasks and scenarios, it can be difficult for humans to accurately communicate their preferences related to robot behaviour. The knowledge of how a user ranks the constraints they have created can help improve the robot's performance in the constrained environment. In [18], Wilde et al. describe an approach to learn user preferences by ranking alternate paths, from which the importance of the different constraints is learned. Sadigh et al. propose a similar approach in [19], where the weight space of user preferences is based on a probability distribution.

III. Scenario Description

We first describe the scenario that is used to elicit the user specification and generate the graph for robot motion planning. We consider a simulated warehouse setting, where a robot is tasked with material transport tasks. Each task consists of a starting and goal location in the environment.

To begin the specification process, the user is provided with a map of the target environment. An example of a warehouse environment is illustrated in Fig. 1. The user then uses a graphical interface to specify behaviour constraints for the robot, such as *No-Go* zones and *Roads*. Two types of *Roads* were allowable, one-way and two-way *Roads*. The set of constraints that a user creates on this environment is referred to as a *specification*.

Once the user specification is complete, the constraints contained in the specification are used to create the motion graph for the robot. This is done by first creating a uniformly-distributed 4-connected directed grid graph, then deleting all edges connecting vertices that lie inside obstacles. The motion graph for an environment prior to the inclusion of any constraints is designated as the "baseline" graph. The motion graph of a given specification is then created by modifying the edges of the baseline graph as follows: 1) each edge that is incident with a vertex in a *No-go* zone is deleted, 2) each edge that is incident with a vertex in a *Road* zone that is directed in the opposite direction of the *Road* is deleted (red edge in Fig. 2), and 3) each other edge incident with a vertex in a *Road* will have its weight modified. If the edge connects two vertices that belong to the same *Road* zone that are in the specified direction of the road (edge X to X_1 in Fig. 2),

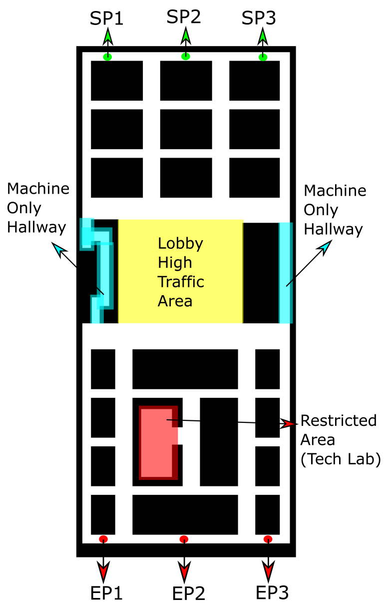


Fig. 1: Map of the target environment. Empty space is in white and occupied space is in black. The green dots represent the 3 possible starting locations while the red dots represent the 3 possible end locations. The central empty area is the "Lobby" of the warehouse, while the two hallways to the left and right of the "Lobby" are machine only areas.

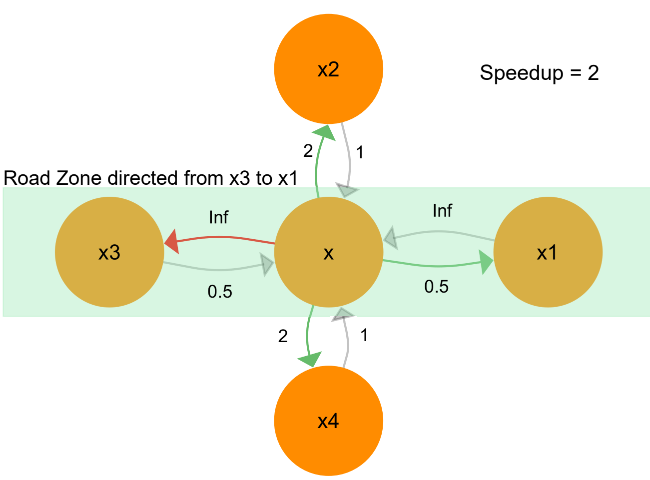


Fig. 2: Examples motion graph generation for a vertex in a *Road* zone from left to right. The edges modified at vertex X are in color. The edge between X and X_3 is going against the *Road* direction and is deleted (cost becomes infinite, highlighted in red). Edges between X and X_2, and X and X_4 are edges leaving the *Road*; their cost is increased (highlighted in green). The edge between X and X_1 is fully contained inside the *Road*, and it is in the forward direction so its cost is decreased (highlighted in green).

then it will have its weight decreased through the following equation:

$$W_{\text{new}}(e) = W_{\text{base}}(e)/\texttt{SF},$$

where $W_{\text{new}}(e)$ is the modified weight of the edge, $W_{\text{base}}(e)$ is the base edge weight, and $\texttt{SF}$ is the *Speedup factor*, which is greater than or equal to 1. Alternatively, if the edge connects two vertices, one that belongs to a *Road* zone, and one that does not (edge X to X_2, and X to X_4 in Fig. 2), then it will have its weight increased as

$$W_{\text{new}}(e) = W_{\text{base}}(e) \times \texttt{SF}.$$

This means that there is no additional cost to join a *Road*, but once inside, it is cheaper to continue down the *Road* than to exit it. Two-way *Road* zones are treated as if they were two one-way *Roads* (of opposite directions) adjacent to each other. In addition to position, the motion graph also encodes heading information, which allows for the differentiation between multiple shortest paths by the number of turns that each of them contains. This is done by associating a very small cost (much smaller than any non-turn edge costs) to all turn actions. The cost of a shortest path will then continue to be primarily dependent on the non-turn edges, but between two equally short shortest paths, the one with fewer turns will have a smaller cost. This was done to penalize unnecessary turning motions. The graphs obtained in this manner are then used to determine the quality of the specification through the application of our metrics described below.

IV. Algorithm and Metrics

We now describe the metrics used to evaluate the quality of a given specification, which is the set of user-created behavioural constraints. We propose three metrics: Entropy, Shortest Paths Coverage Area ($\texttt{SPCA}$), and Loss of Efficiency ($\texttt{LoE}$). The first two metrics capture the positive effects of a specification, related to its ability to constrain the robot behaviour to be more predictable and better conform to user expectations. The last metric, $\texttt{LoE}$, captures the negative effects related to the performance losses encountered by the robot (in particular, the increase in distance traveled by the robot), due to the limits placed on its behaviour. These metrics are computed on the graph representation described in Section III and depend on the speedup factor $\texttt{SF}$ used in modifying the graph's edge weights. To investigate the relationship between these metrics, a specific $\texttt{SF}$ needs to be selected. For this purpose, we have chosen a $\texttt{SF}$ of 2. However, we found the results in the following sections hold for $\texttt{SF}$ values between 2 and 10 (an $\texttt{SF}$ of 1 provides no incentive for *Road* zone use).

A. Positive Effects

The intent of a user specification is to constrain the robot behaviour to make the robot's actions more predictable, and better align them with a user's expectations.

In the context of warehouse robotics, increasing robot predictability primarily refers to making it easier for workers to determine the path that the robot will take. This applies to both operator (supervisor) interactions during tag creation, and to interactions the robot will have with warehouse staff (bystanders) while it navigates the warehouse to accomplish its goals.

A second intent of a specification is to align the behaviour of the robot with user expectations. In many cases, the closest existing technologies to mobile robots are AGVs and forklifts, which are typically constrained to operate on fixed roads, and obey road rules. We hypothesize that autonomous robots that limit their coverage area to follow *Roads* will be better aligned with user expectations.

1) Entropy: Yang and Anderson [15] proposed to use entropy to measure the complexity of an environment, by quantifying the number of decisions that a robot has to make to traverse the environment. We propose to use the environmental entropy to quantify how a specification affects the complexity of the robot's decision making. For this, given a node x_i, we define its set of neighboring nodes as $\mathcal{N}(i)$. Following Yang and Anderson [15], the entropy of a given node in the motion graph can be calculated using

$$H(x_i) = -\sum_{j \in \mathcal{N}(i)} p(x_i, x_j) \log_2 p(x_i, x_j),$$

where given a node x_i, x_j is a neighbor of x_i, and $p(x_i, x_j)$ is the "probability" of the robot transitioning from node x_i to node x_j, i.e., traversing the edge (x_i, x_j). The transition probability captures a localized view of the graph, as it assumes that at a first approximation, transition decisions are based on the edge weights of each outgoing edge from a given vertex. The following formula is used to calculate the transition probability corresponding to each edge

$$p(x_i, x_j) = \frac{\frac{1}{W(x_i, x_j)}}{\sum_{k \in \mathcal{N}(i)} \frac{1}{W(x_i, x_k)}},$$

where $p(x_i, x_j)$ is the probability associated with transitioning from edge x_i to x_j, $W(x_i, x_j)$ is the weight of the edge (x_i, x_j), and $W(x_i, x_k)$ is the weight of the edge (x_i, x_k), where x_k is a neighbour of x_i.

For nodes that have no neighbors (outdegree of 0), the entropy is zero, while for a node where all outedges have an equal weight, the probability to move to any of them will be equal, and the entropy of that node will be equal to 2. When the outedges do not all have the same weights, entropy is reduced as the robot is more likely to traverse the higher probability edge. As an example, consider the set of nodes presented in Fig. 2, where we are interested in calculating the entropy of node x. In Fig. 2, nodes x_3, x, and x_1 are part of a *Road* zone directed from left to right. Applying this formula results in $p(x, x_1)$ being equal to $0.\overline{6}$, while $p(x, x_2)$ and $p(x, x_4)$ are equal to $0.1\overline{6}$ and $p(x, x_3)$ is equal to 0.

To compute the overall entropy of the environment, the entropies of all nodes are summed:

$$H = \sum_{i=0}^{n} H(x_i),$$

where n is the number of nodes in the graph, $H(x_i)$ is the entropy of each node in the graph, and H is the total entropy of the graph.

2) Shortest Paths Coverage Area: The second metric that captures the positive impact of constraints in a specification is related to the area covered by the shortest paths with the minimum number of turns between start and goal points.

In an environment with no restrictions, there will generally be a large number of candidate shortest paths between two points. Adding restrictions on robot behaviour, however, limits the number of shortest paths in the environment. By reducing the number of viable paths between the start and end points, the area covered by those paths is then also potentially reduced. A simple example are two rooms in an environment connected by two separate hallways, resulting in two different paths between the rooms. Placing a *No-go* zone on one of the hallways will result in only one path remaining between the two rooms.

There are two ways that the coverage area can be calculated and interpreted. The first is to count the number of nodes that are traversed by any of the shortest paths. For this, we define S and G as the set of start and goal nodes, respectively. We then define the following indicator function:

$$I(x_i) = \begin{cases} 1, & \text{if } x_i \in P(s, g) \text{ for some } s \in S, \text{ and } g \in G, \\ 0, & \text{otherwise,} \end{cases}$$

which outputs 1 if the node x_i is part of any of the shortest paths $P(s, g)$. To obtain the shortest paths coverage area, SPCA, of a specification, we sum up the result of the indicator function for each of the n nodes of the environment:

$$\texttt{SPCA} = \sum_{i=1}^{n} I(x_i).$$

The shortest paths in the graph are found using Dijkstra's algorithm.

In the second approach, we first compute all shortest paths for each $s \in S$ and $g \in G$. Let the number of such paths be $m(s, g)$. Then the total number of shortest paths over all tasks is $M = \sum_{s \in S, g \in G} m(s, g)$. Then, for each node x_i, we let $C(x_i)$ be the number of those M shortest paths that pass through node x_i. Finally, we define

$$A(x_i) = \frac{C(x_i)}{M}.$$

This quantity gives the fraction of shortest paths from S to G that pass through x_i.

It is important to highlight that a robot will not necessarily pass through all of the areas indicated by the shortest paths coverage, as the shortest paths coverage area is the result of the union of individual paths, and the robot will only navigate on one of those paths. We consider all possible shortest paths in the SPCA metric, since selecting among the shortest paths is an additional optimization problem outside the scope of this work.

B. Negative Effects

The negative effects of a specification aim to capture the drawbacks of constraining robot behaviour. The major drawback of the user constraints comes from increasing the length of shortest paths between start and end points compared to the baseline motion graph.

1) Loss of Efficiency: The LoE metric seeks to determine the relative increase in length of the shortest path between a pair of points on the map between a motion graph obtained from a specification and the baseline motion graph, using the following formula:

$$\texttt{LoE}(x_i, x_j) = \frac{d_s(x_i, x_j) - d_b(x_i, x_j)}{d_b(x_i, x_j)},$$

where x_i and x_j are two nodes of the graph, $d_s(x_i, x_j)$ is the distance between x_i and x_j on the specification motion graph, and $d_b(x_i, x_j)$ is the distance between the two nodes on the baseline motion graph with no specifications applied. Where there is no path between two nodes, the distance between them is set to be equal to n, the total number of nodes in the graph, when calculating the Global LoE, and Inf when calculating the task LoE. This calculation can be applied to a single task (a single pair of start and end positions), a set of tasks, or to all pairs of nodes in the graph. Based on this metric, we can then calculate the mean LoE of the entire specification.

V. Experimental Design

To evaluate the proposed metrics and determine how novice users provide specifications, we implemented and tested an interface that allowed users to create specifications. In the user study, users were presented with a sample specification scenario and asked to generate a specification.

A. Interface Design

The floor plan of the environment used in the study, as well as the simulated environment itself can be observed in Fig. 1. The central area of the environment was described as a "Lobby" with a high amount of foot traffic and several static obstacles, while the areas above and below the "Lobby" were described as being storage/shelving areas. Additionally, a "Tech Lab" area which the robots were not allowed to enter was defined. As an alternative to navigating through the lobby, two hallways were indicated to be machine-only, currently in use by forklifts and other similar equipment. Two different hallway sizes were used: narrow and wide.

The robot tasks were simple pick-up and drop-off tasks. More specifically, the robot might need to traverse from any of the three starting locations (SP1, SP2, SP3) to any of the three end locations (EP1, EP2, EP3). To keep the scenario simple, the robots were not required to navigate back to their original starting positions.

Two constraints were made available. The first was a *No-Go* Zone which specified an area of the map where the robot was not allowed to enter or navigate through. The second constraint was a *Road*, of which there were two types: one-way and two-way. *Road* zones could be used to limit navigation in the indicated direction and encourage robots to take certain paths through the environment. While the width of the *Road* was customizable, the default width allowed for a one-way *Road* to perfectly fit in the narrow hallways and for the two-way *Road* to fit in the wide hallways of the environment.

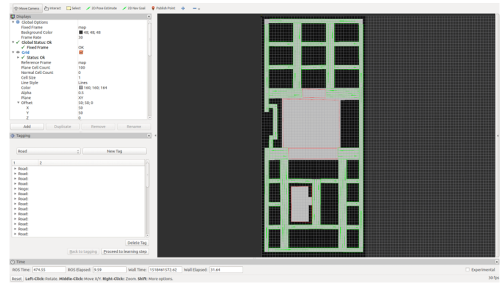


Fig. 3: RVIZ-based interface used in the study

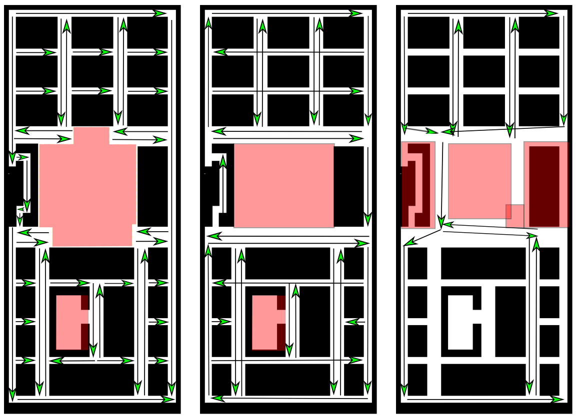
The developed application allows a user to easily command a warehouse robotic system by marking areas of the map with tags that serve to constrain the robot's behaviour in that specific area. The application was designed as a plug-in for RVIZ, Robot Operating System's (ROS) 3D visualizer. This allowed us to take advantage of the ROS framework and simulators to create more realistic environments. In addition, designing the application as a plug-in for RVIZ also made the interface more compact, by combining data streams and reducing the number of windows that a user had to manage, which have been previously linked to increased robot-performance [20], [21]. The two resulting windows that the user works with are the RVIZ window that contains the interface, and a view of the Gazebo simulator which shows the virtual world that the robot operates in. The robot used in the study was a simulated Clearpath Husky robot, which could be tele-operated through the use of an Xbox game controller.

B. Study Design

1) Participants: We recruited 8 University of Waterloo students via mailing lists. Out of these participants, 6 were undergraduate students, while 2 were graduate students.

2) Procedure: The study was approved by the Office of Research Ethics at the University of Waterloo. Each study session took approximately 1 hour. After being informed about the study and providing consent, participants were provided with a brief introduction and an overview of the study and their tasks.

Participants then proceeded to the training phase, where they were allowed to familiarize themselves with the interface. They were introduced to the RVIZ and Gazebo interfaces, and the allowable tags and their properties were described. The participants were then provided with hands-on time with the system, and were instructed to create at least one tag of each type, and to attempt to tele-operate the robot through the simplified training environment. Once the participants indicated that they were finished with the



(a) Specification 2 (b) Specification 4 (c) Specification 5

Fig. 4: Sample user specifications

training section of the study, they would proceed to the actual experiment scenario.

As the beginning of the experiment phase, the participants were provided with a floor plan of the environment (illustrated in Fig. 1), as well as additional contextual descriptions of the various areas of the environment. Based on these two items, they were instructed to create tags in order to ensure that the robot could safely reach its intended goals from any of the three starting points. Additionally they were informed that there was no limit to the number of tags that they could create, and that they should continue until they were satisfied with their efforts. Once the participants indicated that they had finished tagging the environment, they were asked to tele-operate the robot from SP1 to EP2 in Figure 1. The experiment phase of the study ended when the tele-operated robot successfully reached EP2.

Finally, following the end of the experiment the participants were asked to fill out a questionnaire and verbally expand on their answers if desired. The questions were split into two types. The first set of questions were related to system usability and how easy/difficult it was to use the developed interface. The second set asked the participants to assess their own performance in accomplishing the task, and included the question:

- How WELL do you think you specified the robot task?

It is important to note that participants were not provided with the metrics proposed in this paper while generating the specification.

VI. Results and Discussion

A. Specification Variety

Even though the participants were provided with identical task instructions, a large variety of specifications were observed, both in number of tags and in the area that they occupied. The average number of tags contained in a specification was 23.12, with a standard deviation $\sigma = 11.31$, and on average, tags covered 53.48% of the map ($\sigma = 16.38\%$).

In addition to variations in the number and area coverage of tags, there was also a large variety in terms of how the tags

TABLE I: Mean and standard deviation values for each of the performance metrics

SPCA	Entropy	Task LoE	Global LoE
190 ($\sigma = 100$)	551 ($\sigma = 169$)	16 ($\sigma = 9$)	4172 ($\sigma = 1795$)

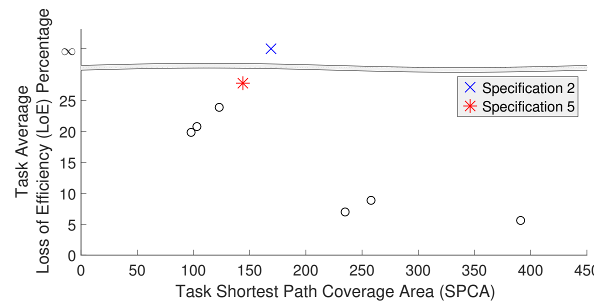


Fig. 5: Average LoE versus SPCA. In this figure, the average LoE was calculated over the 9 pair of start and end points that specified the robot task.

are configured and specified. In the example specifications in Fig. 4, we can see that Roads differ in direction, coverage and the amount of overlap between different Road zones. Similarly, the extents and locations for No-Go zones differ between specifications.

The specifications indicated that the extent to which participants considered *global* performance (i.e., performance outside of the specified start and end positions) also differed. The global performance of a specification would be useful if the robot were to perform tasks other than the ones described in the study, or if the robot was disturbed and then had to re-plan from an unexpected area of the environment. One example of differences between the participants in terms of global considerations comes in the form of the *No-Go* tags that some specifications have on the Lab Tech room, despite the fact that based on the robot's given task set, it should have no reason to navigate through that room.

The different specifications resulted in considerable differences in the performance levels, as seen in Table I. Note the mean and standard deviation calculation for the Task Average LoE does not include the outlier specification observed in Fig. 5 for reasons detailed in Section VI-B.

B. Task-specific Performance

We first analyze how participants traded-off between positive and negative effects for task-specific performance, i.e., when only the performance on the specified task is considered. The relevant metrics are the Task SPCA, illustrated in Fig. 7, and Task Average LoE (increase in distance travelled for task completion).

Fig. 5 immediately indicates one large outlier, Specification 2. In this specification, the robot cannot reach all of the end points from all of the starting points (e.g. EP3 from SP3) which results in a score of infinity. Our LoE metric was able to identify the failure of Specification 2, which was not

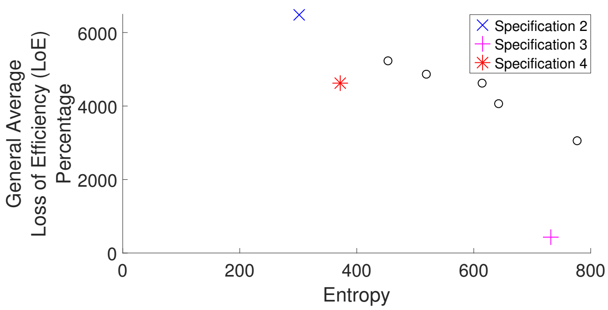
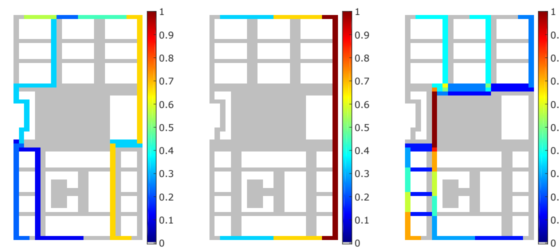


Fig. 6: Average LoE versus Entropy. In this figure, Average LoE was calculated over all points in the map.



(a) Specification 2 (b) Specification 4 (c) Specification 5

Fig. 7: Shortest Paths Coverage Area of 3 specs, illustrating the differences between specs. The colors encode the ratio of shortest paths that pass through each point on the map

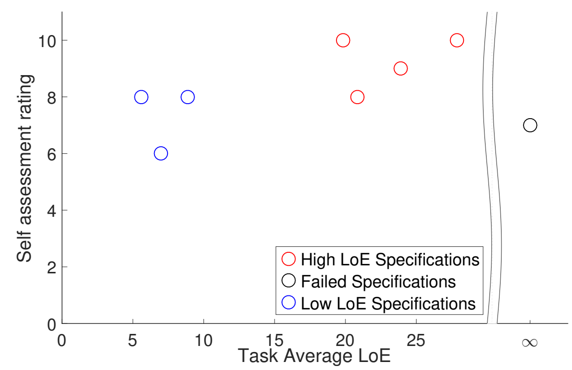


Fig. 8: The participants' self-assessed rating of how well they specified the robot task, shown against the Task Average LoE scores of their specifications

immediately obvious by looking at Fig. 4a. The rest of the specifications averaged a task LoE score of 17%, with three of them attaining a task LoE score of under 10%. It should be noted that the three specifications with a low LoE score have higher SPCA scores than the rest. In general, specifications that have a larger SPCA have smaller efficiency losses.

C. Global Performance

Although participants were told of only one set of tasks that the robot needs to accomplish, it is useful to examine how performance is impacted if a robot was required to solve additional tasks using the same specification, or if disturbances forced the robot away from paths connecting the task start and end points. Since we are interested in task agnostic performance, Entropy and Global Average LoE will be used as the metrics representing the positive and negative effects of a specification globally.

While a few participants created specifications with low levels of Task Average LoE, that was not the case for General Average LoE. As Fig. 6 indicates, only one specification obtained a relatively low Global LoE score(430%), while the rest obtained much higher scores (average 4700%). This leads us to believe that participants did not generally consider the global implications of their specifications, and that they primarily constructed them to serve the sample task provided in the study. This is further evidenced by observing that some specifications (e.g Specification 5) completely block the flow of traffic from the lower side of the map to the top side in Fig. 1, which results in high efficiency losses.

At the global level, it can be difficult for a human operator to predict how even small changes in the specification can affect the performance of the robot. This is seen in Fig. 6 with Specifications 2 and 4, where despite having similar entropy levels (predictability of behaviour), there is a very large difference in their Global LoE values, which may not be apparent by simply looking at the specifications (Fig. 4).

D. User Questionnaire

After completing the specification, the participants were asked to assess their task performance.

The responses, comparing user assessment of their performance at specifying the robot task against Task LoE, is shown in Fig. 8. Surprisingly, the three participants who created the specifications with the smallest Task LoE scores, and thus the fewest negative effects on performance, rated their performance below the average of the participants who authored the high Task LoE specifications (7.3 versus 8.8 rating). While the participant who created a specification that failed to meet the task objectives (Specification 2), rated their performance slightly under the average rating of the top performers, their rating was still high, despite failing to meet the robot task.

Since the users' self-assessment ratings do not match their specification's Task LoE performance, it was hypothesized that the users might have instead rated the global performance of their specifications. However, upon investigating how the users' self-assessment rating matches the Global LoE, we identified a similar relationship to that in Fig. 8. That is, participants with a lower Global LoE rated themselves lower than participants with a higher Global LoE. This indicates that it was difficult for participants to accurately rate the performance of their specifications.

E. Discussion

Our research investigates the quality of user specifications by measuring the positive and negative effects of specifications.

Our results found that the choice of tags can have a large impact on the performance and usability of the robot system, and that novice users who are unfamiliar with the specification system may not be able to fully appreciate the impact of specification choices on subsequent robot performance, as even small changes in the specification can lead different robot behaviour. Additionally, one participant created a specification that prevented the robot from accomplishing its tasks, and failed to recognize this during the study. Our metrics were able to identify this failure. This indicates that they could be used as a verification tool during the specification design process. However, these results need to be confirmed with a larger number of participants.

These findings motivate the need for an interface that can guide users towards obtaining better specifications; a direction of future work is to integrate performance metrics into a preference learning system similar to the ones in [18], [19]. Such a system would allow us to better align the behaviour of the robot with user expectations while minimizing performance loss, based on the feedback obtained from the user. In addition, it would also allow the user to better understand the effects that their configuration has on robot performance, and improve their ability to accurately rate their performance.

Currently meta-information regarding the environment, such as the lobby being an area with high pedestrian traffic, is not used in any measure of performance. Including this type of information could further improve our metrics. Our initial specification system included only two types of constraints, *Roads* and *No-Go Zones*. The expressiveness of the specification could be improved with additional types of constraints, such a *Slow* zone where the robot's speed is reduced, or a *Stop sign* zone where the robot comes to a halt, before proceeding along its path. These additional tags would allow users to encode a more complex set of behaviours into the robot, but might further increase the specification task difficulty. As part of future work, these metrics could also be provided as feedback to the users as they are generating a specification, to investigate how users create and modify a specification when they become explicitly aware of the quality of their work.

VII. Conclusion

In this paper, we introduced a set of measurements describing the effects, both positive and negative, of behavioural constraints applied to a robot. These metrics measure the behavioural predictability of the robot, the extent to which robot behaviour meets a user's expectations, and the loss of efficiency due to following the user specifications. A user study was conducted where novice users generated specifications for a set of warehouse robot tasks. Our metrics illustrate that users given the same task description generate very different specifications, and that differences between specifications have significant impact on robot predictability, legibility and efficiency which may not be readily apparent to novice users. Users were not necessarily capable of accurately assessing their performance in creating these specifications.

References

[1] J. Minguez, F. Lamiraux, and J.-P. Laumond, "Motion Planning and Obstacle Avoidance," in *Springer Handbook of Robotics*. Springer, Cham, 2016, pp. 1177–1202.

[2] M. Desai, K. Stubbs, A. Steinfeld, and H. Yanco, "Creating Trustworthy Robots: Lessons and Inspirations from Automated Systems," *Proceedings of the AISB Convention: New Frontiers in Human-Robot Interaction*, Apr. 2009.

[3] B. M. Muir and N. Moray, "Trust in automation. Part II. Experimental studies of trust and human intervention in a process control simulation," *Ergonomics*, vol. 39, no. 3, pp. 429–460, Mar. 1996.

[4] R. Parasuraman and V. Riley, "Humans and Automation: Use, Misuse, Disuse, Abuse," *Human Factors*, vol. 39, no. 2, pp. 230–253, Jun. 1997.

[5] M. A. Goodrich and D. R. Olsen, "Seven principles of efficient human robot interaction," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, Oct. 2003, pp. 3942–3948 vol.4.

[6] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "Situation Awareness, Mental Workload, and Trust in Automation: Viable, Empirically Supported Cognitive Engineering Constructs," *Journal of Cognitive Engineering and Decision Making*, vol. 2, no. 2, pp. 140–160, Jun. 2008.

[7] J. Scholtz, "Theory and evaluation of human robot interactions," in *36th Annual Hawaii International Conference on System Sciences*, Jan. 2003.

[8] J. Scholtz and S. Bahrami, "Human-robot interaction: Development of an evaluation methodology for the bystander role of interaction," in *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, Oct. 2003, pp. 3212–3217 vol.4.

[9] A. Steinfeld, T. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich, "Common Metrics for Human-robot Interaction," in *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, ser. HRI '06. New York, NY, USA: ACM, 2006, pp. 33–40.

[10] J. W. Crandall, M. A. Goodrich, D. R. Olsen, and C. W. Nielsen, "Validating human-robot interaction schemes in multitasking environments," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 35, no. 4, pp. 438–449, Jul. 2005.

[11] A. Lampe and R. Chatila, "Performance measure for the evaluation of mobile robot autonomy," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2006, pp. 4057–4062.

[12] J. W. Crandall, "Towards developing effective human-robot systems," Ph.D. dissertation, Brigham Young University. Department of Computer Science, 2003.

[13] J. W. Crandall and M. A. Goodrich, "Measuring the intelligence of a robot and its interface," in *Proc. of PERMIS*, 2003.

[14] ——, "Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 2002, pp. 1290–1295.

[15] G. Yang and G. T. Anderson, "An experimental study of environmental complexity as seen by robots," in *IEEE International Conference on Systems, Man, and Cybernetics*, Oct. 2011, pp. 3102–3106.

[16] G. T. Anderson and G. Yang, "A proposed measure of environmental complexity for robotic applications," in *IEEE International Conference on Systems, Man and Cybernetics*, Oct. 2007, pp. 2461–2466.

[17] S. H. Young, T. A. Mazzuchi, and S. Sarkani, "A Framework for Predicting Future System Performance in Autonomous Unmanned Ground Vehicles," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1192–1206, Jul. 2017.

[18] N. Wilde, D. Kulic, and S. L. Smith, "Learning User Preferences in Robot Motion Planning through Interaction," in *IEEE International Conference on Robotics and Automation*, 2018.

[19] D. Sadigh, A. Dragan, S. Sastry, and S. Seshia, "Active Preference-Based Learning of Reward Functions," in *Robotics: Science and Systems (RSS)*, Jul. 2017.

[20] A. Norton, W. Ober, L. Baraniecki, E. McCann, J. Scholtz, D. Shane, A. Skinner, R. Watson, and H. Yanco, "Analysis of human–robot interaction at the DARPA Robotics Challenge Finals," *The International Journal of Robotics Research*, vol. 36, no. 5-7, pp. 483–513, Jun. 2017.

[21] A. Steinfeld, "Interface lessons for fully and semi-autonomous mobile robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, Apr. 2004, pp. 2752–2757.