2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)
Bari, Italy. October 6-9, 2019

Intent-based Robotic Path-Replanning: When to Adapt New Paths in Dynamic Environments

Meher T. Shaikh[1] and Michael A. Goodrich[2]

Abstract*— For goal-based robot navigation in a dynamic environment, human intent includes expectations about what performance objectives are satisfied by a planned path in terms of objectives to be met. If the planned path drifts from the human's intent as a result of environment changes, the path needs to be replanned. This paper presents a replanning framework with three elements: (a) the integration of fast online path-planning algorithms that generate trajectories conforming to the given intent; (b) a mathematical model that says when replanning must happen; and (c) an evaluation of events that *trigger* replanning. An interactive graphical user interface enables a human to accept or reject replanned paths when a trigger happens. A study of 50 MTurk participants is used to assess *what replanning triggers best enable a human-robot collaboration to persistently satisfy intent?**

I. INTRODUCTION

A human's intent for a robot includes the robot's activity — *what* the robot should do — as well as the objectives associated with the robot's activity — *how* the robot should do it. This paper applies this definition of intent to the problem of multi-objective robot path-replanning in dynamic environments. The human's intent is represented by a planned trajectory that reaches a desired end state while appropriately balancing tradeoffs between objectives. While the robot executes the trajectory, the environment may change causing the objective functions to change over time.

When objectives change over time, the initial chosen trajectory may fail to meet the human's intent while the robot moves to the goal. For example, suppose that the selected trajectory was to evade enemies in the environment but during execution the enemy moves really close to the initially planned trajectory. Under such conditions, an alternative path needs to identified. This is referred to as 'replanning'. Importantly, the revised plan should align with human intent. The design question is therefore, **under what circumstances should the robot switch from it's current trajectory to a replanned trajectory?** *Triggers* are events that signal the human to consider replanning. They provide an opportunity to correct a planned path to keep it aligned with intent.

This work complements our previous works that define triggers [1] and graphical user interface (GUI) for robot path planning [2]. Accordingly, the replanning system architecture has an interactive GUI, the robot, and path-planning/replanning algorithms. In this work, we extend the GUI to enable a human to manage replanning. On a trigger, the GUI communicates (a) the robot's *current location*, (b) the *current path*, (c) the *replanned path*, and (d) interface elements to switch to preferred path choices, such as pop up buttons that allow the user to either 'Stay with the current path' or 'Switch to the new path'.

We provide a mathematical model for when a robot should replan while navigating in changing environments. The model helps quantify the intent-mismatch associated with a path. The mismatch is monitored through three classes of triggers: (a) *time-based*: replanning at regular time intervals, (b) *intent-based*: replanning when the executing path no longer matches intent, and (c) *region-based*: replanning when there is reason to believe that a better path can be obtained from a different homotopy class. We evaluate these three replanning triggers using MTurk participants.

II. PROBLEM FORMALISM

A. Path-Planning Task

The path-planning task is to find a solution, σ starting from an initial state (or robot configuration) and terminating at a specified goal state (or robot configuration) bounded by time. Let X_{init} be the initial state, and let X_{goal} be the goal state. A solution or a *trajectory*, σ, is a sequence of states, $\langle x_0, x_1, \ldots, x_{n-1}, x_n \rangle$ such that $x_0 = X_{\text{init}}$ and $x_n = X_{\text{goal}}$. This paper uses the term 'trajectory' and 'path' interchangeably.

B. Multi-Objective Path-Planning

Denote the set of finite possible trajectories from X_{init} to X_{goal} as $\Sigma = \{\sigma_i\}$. Each trajectory, σ_i, is represented as a sequence of directed edges made of n vertices. Thus, the sequence of configurations replaces $\langle x_0, x_1, \ldots, x_{n-1}, x_n \rangle$ with $\langle v_0, v_1, \ldots, v_{n-1}, v_n \rangle$ where v denotes a vertex in the path. Assuming that the problem has J objectives to deal with, each σ_i is associated with a cost vector defined as $\mathbf{c}(\sigma_i) = [c_1(\sigma_i), \ldots, c_J(\sigma_i)]^T$.

Let $c_j(v_k, v_{k+1})$ denote the cost for objective J to traverse from a parent vertex, v_k, to a child vertex, v_{k+1}. The j^{th} objective cost of σ_i is the sum of the costs of the edges. Thus,

$$\forall j \quad c_j(\sigma_i) = \sum_{k=0}^{n-1} c_j[v_k, v_{k+1}] \tag{1}$$

where v_k equals the location of vertex k.

The multi-objective path-planning problem is to find a trajectory σ such that the resulting cost vector $\mathbf{c}(\sigma)$ satisfies some *trajectory predicate*. For example, a trajectory predicate could be to find the path that minimizes the cost for

[1]Meher T. Shaikh is a PhD student with the Department of Computer Science, Brigham Young University, mtalats@gmail.com
[1]Michael A. Goodrich is a Professor of Computer Science, Brigham Young University, mike@cs.byu.edu

978-1-7281-4569-3/19/$31.00 ©2019 IEEE

objective j, in which case the solution to the multi-objective path-planning problem would be $\sigma^* = \arg\min_{\sigma\in\Sigma} c_j(\sigma)$. Similarly, a trajectory predicate could be to find the path that uses a weighting vector $\mathbf{w} = [w_1, \ldots, w_j]^T$ to find a tradeoff among objectives that satisfies $\sigma^* = \arg\min_{\sigma\in\Sigma} \mathbf{w}^T\mathbf{c}(\sigma)$.

C. Collaborative Human-Robot Path-Planning/Replanning

Given the robot's initial configuration, a collaborative human-robot path-planning problem requires (a) a human to specify the goal state, X_{goal}, and the trajectory predicate encoded as intent, $\mathbf{h}$, and (b) a robot to generate and follow trajectory solutions, σ, that reaches a goal state and satisfies the trajectory predicate. For predicate, the human will use natural language-like descriptors like "find a safe path and a stealthy path".

For the solution, the robot generates $\Sigma = \{\sigma_i\}$. We consider trajectories that are on the Pareto front, as in other multi-objective problems [3], and the Pareto front is assumed to be convex. The human evaluates the trajectories in Σ and selects a trajectory that matches his or her *intent*. Selecting a trajectory $\sigma_h \in \Sigma$ is equivalent to selecting a desired cost vector tradeoff, $\mathbf{c}(\cdot)$, associated with the trajectory that dictates preference among the J objectives.

In a dynamic environment, the cost vector varies over time. While following the selected trajectory, the robot simultaneously computes a new trajectory, one that matches intent as the world changes. The *R*eplanned trajectory, σ_R, is the result of the robot's ongoing perception. The design question is, under what circumstances should the robot switch from it's current trajectory σ_h to σ_R? A *trigger* is an event that gives the human opportunity to switch to the replanned path, σ_R.

III. Related Literature

Following are four major areas of related work:

a) Planning/Replanning Algorithms: Researchers have created several path-planning algorithms to move a robot from a start configuration to a goal configuration, both for static and dynamic environments; see, for example, [4]–[10]. Most existing replanning algorithms find shortest paths. Others are triggered by environment changes such as the emergence of obstacles [11]–[15]. In Hyperion robot navigation [16], *progress-based* replanning was triggered if the rover did not reach the expected navigation waypoint at the scheduled arrival time. In contrast, Cummings et al. [17] studied how *time-based* replanning triggers and replanning rates affected operator performance and workload when supervising a decentralized network of heterogeneous unmanned vehicles. Yoshida et.al [13] explored replanning using two threads, one for execution and the other for planning. When a *collision* is expected along the current path, the execution thread queries the planning thread for a better plan. In this work, we use time-based, region-based and intent-based replanning for robot navigation.

b) Multi-Objective Planning: In real-time navigation, multiple objectives include path length, energy consumed, coverage, smoothness, traversal risk, safety, stealth, etc. [18], [19]. Multi-objective path-planning is typically applied to static environments [3], [20], [21]. Research on combining multi-objectives and replanning is rare [16], [19]. Work in [16] produced plans that are optimal with respect to weighted combinations of minimum plan length and energy cost. The authors of [19] view the cost of a trajectory as a function of time for traversal, traversal risk, stealth, and visibility.

We explored two path-planners; MORRF* algorithm [22] and online fast marching tree* (O-FMT*) [23]. MORRF* blends two concepts: optimal rapidly exploring random tree (RRT) [24] for efficient path finding, and a decomposition-based approach to multi-objective optimization [25]. MORRF* can be slow and is therefore not appropriate when considering replanning. To achieve faster replans, O-FMT* is evaluated. However, any fast replanning algorithm could replace O-FMT*.

c) Human Intervention: This paper combines algorithmic (re)planning and human supervision thereby placing it in the category of human supervisory control [26]. His work emphasizes *monitoring* the automatic action to detect failures followed by corrections. The trigger mechanism discussed here is analogous to the term *intervention* from Scholtz [27], which means identifying when the expected actions of the robot are not appropriate given the current situation. Thus, our notion of a trigger is closely associated with prior uses of 'intervention' and 'correction'.

d) Intent: In human-robot interaction (HRI) applications, intent is generally "owned" by the human and expressed through a command and/or correction. Commands dictate (a) *what* the robot should do and (b) *how* to do it. This intent is explicitly or implicitly communicated to the robot [28]–[31]. Commands to the robot can be in the form of plans, images, sketches, etc. that are convenient when the robot is remotely working in difficult, dangerous, and unstructured environments [32].

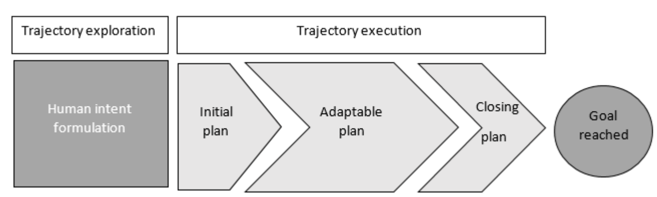


Fig. 1: Life cycle of human-robot collaboration task.

IV. Intent-Based Multi-Objective Path Planning

Fig. 1 illustrates the execution phases for intent-based planning and replanning. The process starts when the human formulates and expresses intent and ends when the goals associated with the intent are accomplished. In between, the robot follows the planned or replanned trajectory.

During execution, the robot may follow an *initial plan*, an *adaptable plan*, and a *closing plan*. In the initial plan phase, the robot follows the original planned trajectory, σ_h, transitioning from one configuration to another until replanning is triggered. In the adaptable plan phase, the robot adapts plans. The phase is metaphorically wider than the initial phase to emphasize that the robot may have to replan multiple times. When the robot is close by the goal state, the human and the robot may decide to ignore intent and choose instead a closing plan that effectually disregards nuances in intent in favor of "just reaching the goal." The remainder of this section is a review of our previous published work on multi-objective path planning [2].

A. Creating Meaningful Cost Functions

This work is motivated by path-planning in adversarial environments, hence, we consider three costs, *quickly*, *stealthily* and *safely*. The "quickly" cost is the sum of the Euclidean distances of each edge in the trajectory. The "stealthily" cost function is loosely modeled as the probability of the robot being seen by the enemy. It is the sum of costs for each point on the trajectory, computed as a function of two factors: the distance of the robot from each enemy and the visibility of the robot from all enemies. The safety of a collision-free path is the sum of the inverse distance between the robot position and the nearest obstacle in the environment. This type of "safely" objective is also referred to as "clearance", defined as the maximum possible distance from obstacles [3].

B. Normalization

The objectives used by the path planner are expressed as cost functions, $\mathbf{c}(\sigma_i) = [c_1(\sigma_i), \ldots, c_J(\sigma_i)]^T$, and these may be in incommensurate units. To consider trajectories as commensurate payoffs, each of the cost term c_j is converted to payoff term as $p_j(\sigma_i) = -c_j(\sigma_i)$ and subsequently normalized. The normalized payoff objective for trajectory σ_i is then:

$$O_j(\sigma_i) = \frac{p_j(\sigma_i) - \min_{\sigma \in \Sigma_P} p_j(\sigma)}{\max_{\sigma \in \Sigma_P} p_j(\sigma) - \min_{\sigma \in \Sigma_P} p_j(\sigma)}$$

with a corresponding payoff vector $\mathbf{O}(\sigma_i)$.

C. Intent on the Pareto Front

We are interested in Pareto optimal trajectories. Consider two non-trival objectives, O_1 and O_2 for path planning. Objectives are non-trivial if it is not possible to get the most of objective O_1 without sacrificing O_2 and vice versa. In Figure 2, objectives are encoded as payoffs, meaning higher values are preferred to lower values. Each of the red and blue circles in the figure denote a trajectory represented by its payoff vector. The extreme right blue circle in Figure 2 corresponds to a trajectory that has highest payoff for objective O_1, and similarly, the extreme left blue circle corresponds to a trajectory that maximizes O_2. All other blue circles on the blue curve indicate the best trajectories for different tradeoffs between O_1 and O_2. Notice that each of the red trajectories are "dominated", meaning, there is another trajectory in which all payoffs are higher. The blue Pareto front curve is made up of non-dominated trajectories.

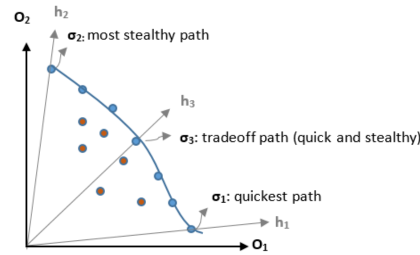


Fig. 2: Two objectives Pareto front of trajectories and intent/trajectory mapping.

For the three adverbs or objectives; 'quickly', 'stealthily', and 'safely', the intent predicate is represented as a three-element vector, $\mathbf{h} = [h_1, h_2, h_3]^T$ [2], where $j \in \{1, 2, 3\}$ and $h_j = [0, 1]$. A value of 1 indicates utmost preference of the corresponding objective, and a value of $h_j = 0$ means ignore objective O_j. For example, if $\mathbf{h} = [1, 0, 0]^T$ then the human wants trajectories that pay attention to only the first objective, and $\mathbf{h} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]^T$ means that the human wants each objective weighted equally.

Figure 2 illustrates how the normalized objective vectors, $\mathbf{O}(\sigma_i)$ and the human intent vector, $\mathbf{h}$, are represented in the same payoff space. For simplicity, this is illustrated when there are two objectives. The vectors emanating from the origin represent possible human intent vectors.

D. Matching Intent to Robot Paths

The intent predicate, $\mathbf{h}$ and the objective vectors, $\mathbf{O}(\sigma_i)$, are scaled so that a) each element h_i and $O_j(\sigma_i)$ fall between 0 and 1 and (b) $h_1 + h_2 + h_3 = 1$. Each intent component h_i is mapped uniquely to one of the RGB colors, which is equivalent to using a color palette to $R + G + B = 1$.

For a given intent, the trajectory that best matches human intent is given by $\sigma_h = \arg\max_{\sigma_i \in T} \mathrm{CS}(\mathbf{h}, \mathbf{O}(\sigma_i))$ where $\mathrm{CS}(\mathbf{h}, \mathbf{O}(\sigma_i))$ is the cosine similarity between $\mathbf{O}(\sigma_i)$ and $\boldsymbol{h}$. In other words, the trajectory vector (in payoff) that aligns closely to the intent vector is the trajectory that get associated with the intent. The following section aid in visualizing this mapping of intent and the robot path.

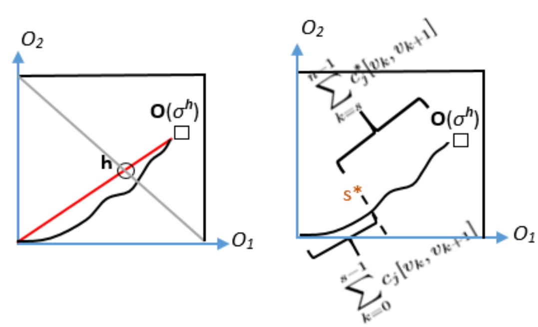


Fig. 3: Path vector $\mathbf{O}(\sigma_h)$, intent vector $\mathbf{h}$, and cost parts.

V. REPLANNING IN DYNAMIC ENVIRONMENTS

Suppose the human intended for the robot to follow a stealthy route — a plan that evades enemies. Since as the robot moves the enemy may also move, the objective costs associated with the robot's trajectory may change such that it may fail to satisfy the intent. The trajectory therefore needs to be adapted or replanned.

A. Replanning trigger

Suppose that the robot has been following the human-selected trajectory σ^h with intent $\mathbf{h}$ for some period of time and has reached vertex $v_s \in \{1, 2, \ldots, n-1\}$, where n is the number of vertices in the original path v = $\langle v_0, v_1, \ldots, v_{n-1}, v_n \rangle$. Even though s parameterizes the trajectory and isn't technically a time, we can treat it as if it is a time unit. So suppose at time s^* something happens and the costs change. For simplicity, suppose that cost function O_j has changed. Should the robot change paths?

Let the cost function after the change be denoted by O_j^* corresponding to new edge costs of $c_j^{s^*}(v_s, v_{s+1})$. The cost function of the path σ^h is adapted from Equation 1 by changing from

$$O_j(\sigma^h) = \sum_{k=0}^{n-1} c_j[v_k, v_{k+1})] \tag{2}$$

to

$$O_j^*(\sigma^h) = \sum_{k=0}^{s-1} c_j[v_k, v_{k+1}] + \sum_{k=s}^{n-1} c_j^*[v_k, v_{k+1}]. \tag{3}$$

The difference between Eq. 2 and Eq. 3 isn't the path; both use the same path $\sigma^h = \langle v_0, v_1, \ldots, v_n \rangle$. The difference is that Eq. 2 uses the original edge costs for all time and Eq. 3 uses the original edge costs up until the cost function changes, which occurs at time s, and then switches to the new cost function.

We'll use a series of figures to illustrate Equation 3. Figure 3 left shows the mapping of a trajectory with intent vector $\mathbf{h}$, but for only two objectives. The unit square represents the set of possible objective vectors; the small square is the end point of the objective vector, $\mathbf{O}$ for a the path, σ_h the red line segment emanating from the origin shows the alignment of σ_h with $\mathbf{h}$ as a result of the mapping discussed in Section IV-D; the diagonal line is the set of possible normalized intents; and the small circle represents the intent for the path σ_h. The squiggly black line below and to the right of the objective vector indicates the total costs that would accrue when the robot walks along the path σ_h if there is no change in the environment. The origin represents the beginning of the problem, before any movement is made, corresponding to $k = 0$; no costs have yet accrued. The curve terminates at the small square, indicating the cumulative cost for following the entire path, corresponding to the accumulated cost at time $k = n$.

What happens if the robot starts moving along σ_h, and environment changes resulting into costs changes? Figure 3 right illustrates the two parts of Equation (3). The total cost of the path turns into the sum of the cost of the path segment up to time s and the cost of the path segment after time s. Note that for this figure, the cost function didn't actually change so the the squiggly line stays the same. The next figure illustrates what happens if the cost function changes.

Figure 4 (a) depicts what happens when the cost functions changes at time s. The squiggly line before time s is precisely what it was in Figure 3 right, but the squiggly line changes after time s because objective costs have changed. As a result, the objective vector has shifted down and to the right.

Figure 4 (b) illustrates that, because the objective vector $\mathbf{O}^*(\sigma_h)$ has changed, the intent associated with the path σ_h has changed. Since we assumed that the human's intent was indicated by the small circle on the diagonal line intersecting the red line, the original path σ_h no longer matches this human intent. Instead, the objective vector now matches another intent indicated by the small circle at the intersection of now a blue line and the new path/objective vector. Should this be a trigger for replanning? Yes, here is the situation when we need to replan a new path.

Note that, we only compute a new path such that the current path and the new path, also called as the *replanned path*, are identical to σ^h up to time s; after time s the replanned path may differ from σ_h.

The problem is illustrated in Figure 4 (c). The robot has followed the original blue path up until time s. At time s, a new path needs to be computed that would match the original intent — the intent at the intersection of the red line and the diagonal in Figure 4 (b). Thus, at time s, the human needs to decide whether it wants the robot to continue along the original blue path or switch to a new brown path that builds from the original blue path.

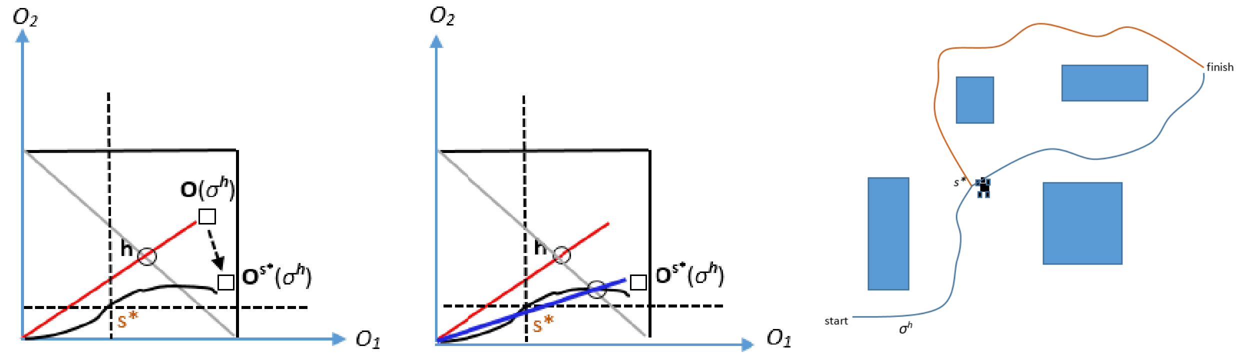
B. When to Replan

Previous work identified multiple triggers when a human may replan [1]. This paper evaluates three triggers: *time-trigger* , *intent-mismatch trigger*, and *homotopy trigger*. At each of the trigger, the human is presented with a replanned trajectory and allowed to choose between the replanned trajectory and the original trajectory.

The time trigger signals the human to check if something is wrong at regular time intervals. Most of these checks may result in the human concluding that the path still matches intent, with an occasional need to replan detected.

The intent-mismatch trigger is analogous to system alerts on human-machine systems. These alerts seek the human's attention if something goes wrong with respect to expectations, and uses the cosine similarity distance between the path objective vector and the intent vector (red vector in Figure 4 (b)). The intent-mismatch trigger indicates that the current path no longer satisfies human intent.

Given that a path replanner is always running in the background, the homotopy trigger signals the human when the replanned path is in a different homotopy class compared to the current path, giving the human the opportunity to switch to the new path that resembles a detour.



(a) Cost changes in path vector $\mathbf{O}(\sigma_h)$ (b) Intent vector before s and after s (c) Branch out from the original path

Fig. 4: Changing costs resulting into intent mismatch.

We hypothesize that the intent-mismatch and homotopy triggers help the operator to replan at critical times/events that should improve the performance of human-robot collaborative tasks. A natural limitation of this assumption is whether the replanned trajectory matches human intent.

VI. EVALUATION OF TRIGGERS

To determine the usefulness of these triggers, we recruited Amazon Mechanical Turk (MTurk) workers to answer questions in a survey format on each of the three trigger mechanisms. When a trigger occurs, the robot stops to seek advice from the human.

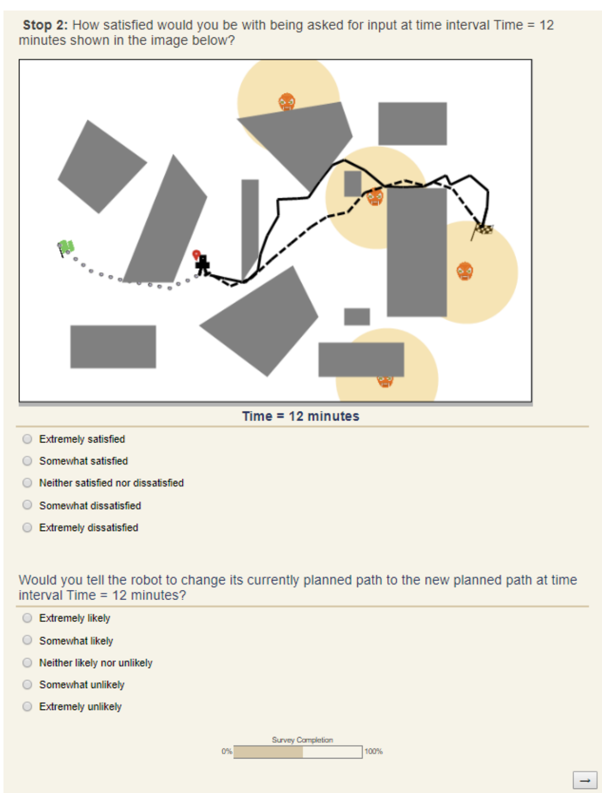


Fig. 5: Example survey question, obstacles, enemies, original dashed path, and replanned solid path.

For evaluating each trigger type, we used three time lapse images of trigger occurrences of a simulated robot moving from start to goal using the interface illustrated in Figure 5. Each trigger image shows the environment with (a) the robot's *current location*, (b) the *current path* — the path which the robot is currently following shown by a dashed pattern, (c) the *new path* that the robot has recalculated and thinks is better — shown by a solid pattern, and (d) the enemy location (orange entities).

For each image, participants were asked two types of questions. The first question type, Q1 category, asked for a participant's opinion on whether s/he was *satisfied on being asked for advice by the robot* at that particular walk juncture. The second question type, Q2 category, asked for a participant's opinion of whether s/he would *recommend* the robot to change the path. Fig. 5 shows an example of the two questions related to a time trigger image.

The response to each of the question is evaluated on a 5-point Likert scale. For the first question, the scale went from *"Extremely satisfied"* indicated with 1 to *"Extremely dissatisfied"* with 5. Responses collected closer to 1 for this question would mean that the trigger under investigation captured critical juncture when replanning was desired. For the second question, response scale goes from *"Extremely likely"* as 1 to *"Extremely unlikely"* as 5. Responses closer to 1 for this question would mean that the robot could be guided to a better path than the one it is following using that particular trigger mechanism.

A. Data

Given the three trigger types, three images in each type, and two questions on each image, we had 18 questions for each participant to answer. The trigger types were presented in pseudo random order to avoid bias towards any trigger type. 50 MTurk workers participated, P=50. After completing an IRB-approved consent form and reading through training, each of participant provided 18 responses. We report results for all 50 participants.

B. Hypothesis

a) Time Trigger: Since, time triggers may or may not capture intent-mismatch, we hypothesize that participants will not favorably view time triggers.

b) Intent-Mismatch Trigger: Based on intent-mismatch theory, we hypothesize that participants will express satisfaction regarding the occurrence of the trigger. Further, we hypothesize that most participants will recommend the robot to switch to the new path.

c) Homotopy Trigger: Since homotopy triggers offers an intent-based path from a different homotopy class, we hypothesize that the participants may want to consider alternate route, a detour, if available shown by this trigger.

d) Replanned path: Since the replanned path is derived from an algorithm that is seeking to most closely match human intent, we hypothesize that the new path will almost always better match intent than the current path.

e) Correlation between Q1 and Q2 responses: We expect a *correlation* to be evident between *Q1 category* and *Q2 category* questions responses. For example, if the participant was extremely satisfied that the robot stopped at an alert because the current path was violating intent, then s/he would most likely recommend the robot to change its path, unless the new path is equally bad or worse.

VII. RESULTS

A. Summary Statistics

Image Sequence	Mean	Std Dev	Std Error
Time_1_Q1	2.04	1.07	0.15
Time_1_Q2	1.92	1.43	0.2
Time_2_Q1	3.42	1.2	0.17
Time_2_Q2	2.92	1.29	0.18
Time_3_Q1	3.32	1.33	0.19
Time_3_Q2	3.62	1.03	0.15

(a) Time trigger image sequence. Average Mean: 2.87

Image Sequence	Mean	Std Dev	Std Error
Alert_1_Q1	2.3	1.07	0.15
Alert_1_Q2	2.54	1.43	0.2
Alert_2_Q1	2.32	1.35	0.19
Alert_2_Q2	1.96	1.19	0.17
Alert_3_Q1	1.58	1.09	0.15
Alert_3_Q2	1.34	0.87	0.12

(b) Alert trigger image sequence. Average Mean: 1.82

Image Sequence	Mean	Std Dev	Std Error
Detour_1_Q1	1.74	1.12	0.16
Detour_1_Q2	1.62	1.26	0.18
Detour_2_Q1	1.64	1.17	0.17
Detour_2_Q2	1.44	1.07	0.15
Detour_3_Q1	2.2	1.21	0.17
Detour_3_Q2	2.24	1.45	0.21

(c) Detour trigger image sequence. Average Mean: 1.81

TABLE I: Different trigger type response statistics.

Table I shows the summary response statistics with 50 participants for the three evaluated trigger types. The 'Mean' column conveys the importance of each trigger type. For *mean*, we were expecting that an appreciated trigger would have response values between 1 (Extremely satisfied/likely) and 2 (Somewhat satisfied/likely) both for both Q1 and Q2 categories. The intent-mismatch trigger and the homotopy trigger have average means of 1.82 and 1.81 respectively. These means indicate that the responses lie between 'Extremely satisfied and Somewhat satisfied' and 'Extremely likely and Somewhat likely' for the Q1 and Q2 category questions, respectively. The average means provide evidence that support the hypothesis that robot seeking human advice at these triggers was appreciated by the participants.

By contrast, the average mean of 2.87 of the time trigger indicate that the participants were less appreciative of regular checks of robotic paths. A value of 3 for a response indicate neutral feedback for a trigger occurrence. These results provide evidence that although monitoring the navigation regularly is important it may not be an important reason to ask a human about whether the robot should change paths.

B. Comparing Trigger Types

Table II shows the significant differences between the means of the three triggers for $Q1$ category. Significance was computed using the Least Squares Means method using Tukey adjustments on participants. The asterisk $*$ denotes significant differences. There were significant differences between time and intent-mismatch trigger. Similarly, time trigger differed significantly from homotopy trigger. However, there was no significant difference between the means of intent-mismatch and the homotopy trigger. Similar significance pattern was observed for $Q2$ category responses (separate table not shown).

Trigger	Trigger	Std Error	t value	Adj P
Alert	Detour	0.155	1.33	0.38
Alert	Time	0.155	-5.54	$< .0001$*
Detour	Time	0.155	-6.87	$< .0001$*

TABLE II: Triggers comparison.

C. Change Path Recommendation

Table III shows the mean recommendation values obtained from 50 participants for Q2 category at each trigger juncture. That is, the statistics about the preference of participants recommending the robot to switch to the new/replanned path. Based on the 'Mean' column in the table, participants recommended changing path for alerts and detours more compared to the time trigger junctures.

Image Sequence	Mean	Std Dev	Std Error
Time_Image1_Q2	1.92	1.43	0.2
Time_Image2_Q2	2.92	1.29	0.18
Time_Image3_Q2	3.62	1.03	0.15
Alert_Image1_Q2	2.54	1.43	0.2
Alert_Image2_Q2	1.96	1.19	0.17
Alert_Image3_Q2	1.34	0.87	0.12
Detour_Image1_Q2	1.62	1.26	0.18
Detour_Image2_Q2	1.44	1.07	0.15
Detour_Image3_Q2	2.24	1.45	0.21

TABLE III: Mean recommendation at different triggers.

D. Correlation between Q1 and Q2 responses

Table IV shows the correlation between Q1 and Q2 responses. Pearson correlation coefficient given with *System Analysis System* tool was adopted to determine if there existed any linear relationship between Q1 and Q2 responses. A high correlation is evident between the two response categories for all trigger images except for 'Image 1' of time trigger. This is indicated by the significant p-values in the table. Hence, we can conclude that if one is extremely satisfied with the robot pausing at a trigger and seeking advice, s/he is more likely to recommend the robot to switch paths and vice-versa. The significance pattern shows that Q2 responses closely follow the response pattern of Q1 type except for a few deviations in the time trigger.

Trigger Type	Metrics	Image 1	Image 2	Image 3
Time Trigger	Correlation	0.27	0.66	0.55
	p-value	0.058	$< .0001^*$	$< .0001^*$
Alert Trigger	Correlation	0.34	0.54	0.63
	p-value	0.015^*	$< .0001^*$	$< .0001^*$
Detour Trigger	Correlation	0.3	0.36	0.63
	p-value	0.032^*	0.011^*	$< .0001^*$

TABLE IV: Q1/Q2 responses: Pearson Correlation.

ACKNOWLEDGMENT

This work was partially supported by the U.S. Army RCTA program and by the U.S. Office of Naval Research under grant N000141613025. All results and conclusions are the responsibility of the authors and do not necessarily reflect the opinions of the funding agencies.

REFERENCES

[1] M. T. Shaikh and M. A. Goodrich, "When does a human replan? exploring intent-based replanning in multi-objective path planning," in *Unmanned Systems Technology XX*, vol. 10640, p. 106400G, International Society for Optics and Photonics, 2018.

[2] M. T. Shaikh and M. A. Goodrich, "Design and evaluation of adverb palette: A gui for selecting tradeoffs in multi-objective optimization problems," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pp. 389–397, ACM, 2017.

[3] M. Davoodi, F. Panahi, A. Mohades, and S. N. Hashemi, "Multi-objective path planning in discrete space," *Applied Soft Computing*, vol. 13, no. 1, pp. 709–720, 2013.

[4] L. Kavraki, P. Svestka, and M. H. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*, vol. 1994. Unknown Publisher, 1994.

[5] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.

[6] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[7] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[8] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*, pp. 396–404, Springer, 1986.

[9] A. Stentz *et al.*, "The focussed d^* algorithm for real-time replanning," in *IJCAI*, vol. 95, pp. 1652–1659, 1995.

[10] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1478–1483, IEEE, 2011.

[11] M. Otte and E. Frazzoli, "Rrtx̂: Real-time motion planning/replanning for environments with unpredictable obstacles," in *Algorithmic Foundations of Robotics XI*, pp. 461–478, Springer, 2015.

[12] N. Ganganath, C.-T. Cheng, and K. T. Chi, "Rapid replanning of energy-efficient paths for navigation on uneven terrains," in *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*, pp. 408–413, IEEE, 2015.

[13] E. Yoshida, K. Yokoi, and P. Gergondet, "Online replanning for reactive robot motion: Practical aspects," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 5927–5933, IEEE, 2010.

[14] M. Wzorek, J. Kvarnström, and P. Doherty, "Choosing path replanning strategies for unmanned aircraft systems," in *Twentieth International Conference on Automated Planning and Scheduling*, 2010.

[15] D. Ferguson, N. Kalra, and A. Stentz, "Replanning with rrts," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1243–1248, IEEE, 2006.

[16] P. Tompkins, A. Stentz, and D. Wettergreen, "Mission-level path planning and re-planning for rover exploration," *Robotics and Autonomous Systems*, vol. 54, no. 2, pp. 174–183, 2006.

[17] M. L. Cummings, A. Clare, and C. Hart, "The role of human-automation consensus in multiple unmanned vehicle scheduling," *Human Factors*, vol. 52, no. 1, pp. 17–27, 2010.

[18] K. Fujimura, "Path planning with multiple objectives," *IEEE Robotics & Automation Magazine*, vol. 3, no. 1, pp. 33–38, 1996.

[19] J. Van Den Berg, D. Ferguson, and J. Kuffner, "Anytime path planning and replanning in dynamic environments," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 2366–2371, IEEE, 2006.

[20] K. Jeddisaravi, R. J. Alitappeh, and F. G. Guimarães, "Multi-objective mobile robot path planning based on a* search," in *Computer and Knowledge Engineering (ICCKE), 2016 6th International Conference on*, pp. 7–12, IEEE, 2016.

[21] L. Mandow, J. P. De la Cruz, *et al.*, "A new approach to multiobjective a* search.," in *IJCAI*, vol. 8, 2005.

[22] D. Yi, M. A. Goodrich, and K. D. Seppi, "MORRF*: Sampling-based multi-objective motion planning," in *Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 1733–1739, AAAI Press, 2015.

[23] B. Chandler and M. A. Goodrich, "Online rrt and online fmt: Rapid replanning with dynamic cost," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pp. 6313–6318, IEEE, 2017.

[24] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI*, vol. 104, p. 2, 2010.

[25] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[26] T. B. Sheridan, *Telerobotics, automation, and human supervisory control*. MIT press, 1992.

[27] J. Scholtz, "Theory and evaluation of human robot interactions," in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pp. 10–pp, IEEE, 2003.

[28] A. Bauer, D. Wollherr, and M. Buss, "Human–robot collaboration: a survey," *International Journal of Humanoid Robotics*, vol. 5, no. 01, pp. 47–66, 2008.

[29] M. Awais and D. Henrich, "Human-robot collaboration by intention recognition using probabilistic state machines," in *Robotics in Alpe-Adria-Danube Region (RAAD), 2010 IEEE 19th International Workshop on*, pp. 75–80, IEEE, 2010.

[30] C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin, "Effects of nonverbal communication on efficiency and robustness in human-robot teamwork," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 708–713, IEEE, 2005.

[31] R. A. Knepper, C. I. Mavrogiannis, J. Proft, and C. Liang, "Implicit communication in a joint action," in *Proceedings of the 2017 acm/ieee international conference on human-robot interaction*, pp. 283–292, ACM, 2017.

[32] T. B. Sheridan, "Human–robot interaction: status and challenges," *Human factors*, vol. 58, no. 4, pp. 525–532, 2016.