Robot Arms Too Short?
Explaining Motion Planning Failures using Design Optimization

Wenxi Wu and Martim Brandão

***Abstract*— Motion planning algorithms are a fundamental component of robotic systems. Unfortunately, as shown by recent literature, their lack of explainability makes it difficult to understand and diagnose planning failures. The feasibility of a motion planning problem depends heavily on the robot model, which can be a major reason for failure. We propose a method that automatically generates explanations of motion planner failure based on robot design. When a planner is not able to find a feasible solution to a problem, we compute a minimum modification to the robot's design that would enable the robot to complete the task. This modification then serves as an explanation of the type: "the planner could not solve the problem because robot links X are not long enough". We demonstrate how this explanation conveys what the robot is doing, why it fails, and how the failure could be recovered if the robot had a different design. We evaluate our method through a user study, which shows our explanations help users better understand robot intent, cause of failure and recovery, compared to other methods. Moreover, users were more satisfied with our method's explanations, and reported that they understood the capabilities of the robot better after exposure to the explanations.**

I. INTRODUCTION

Motion planning algorithms play a crucial role in autonomous robotic systems, as they compute the trajectory for robots to accomplish tasks such as manipulation and navigation. However, these algorithms often lack interpretability and explainability, often making it difficult to understand why they fail to find solutions [1]. Understanding and diagnosing the causes of failure can be difficult and requires expertise and previous experience with robots. Automatically generated explanations of failure can address this issue, aiding both users and engineers in understanding why motion planners fail, anticipating future failures, learning the capabilities and limits of robots, or adapting robot design to prevent failure [1].

In this paper we propose a method that generates explanations of motion planning failure based on robot design. Our algorithm computes the minimum modification to the robot's design required to turn an infeasible problem into a feasible one, and thus shows users what the current design lacks. Our method simultaneously answers the questions "why did the robot fail?" and "how should the robot have been designed to avoid failure?". Therefore, our proposed algorithm outputs an explanation of the type "the planner failed to find a solution to the problem because the upper arm of the robot is not long enough". This contrastive explanation can inform users of a different robot design to contrast the failure to, which, as we will show, helps users understand the capabilities and limits of the robot. Our proposed method works by finding the minimum change to robot link lengths that makes a problem feasible, leveraging a reformulation of the design optimization problem as a normal trajectory optimization problem with virtual pneumatic joints.

This work was supported by the UKRI Centre for Doctoral Training in Safe and Trusted Artificial Intelligence [EP/S023356/1].

*Both authors are with King's College London, UK.

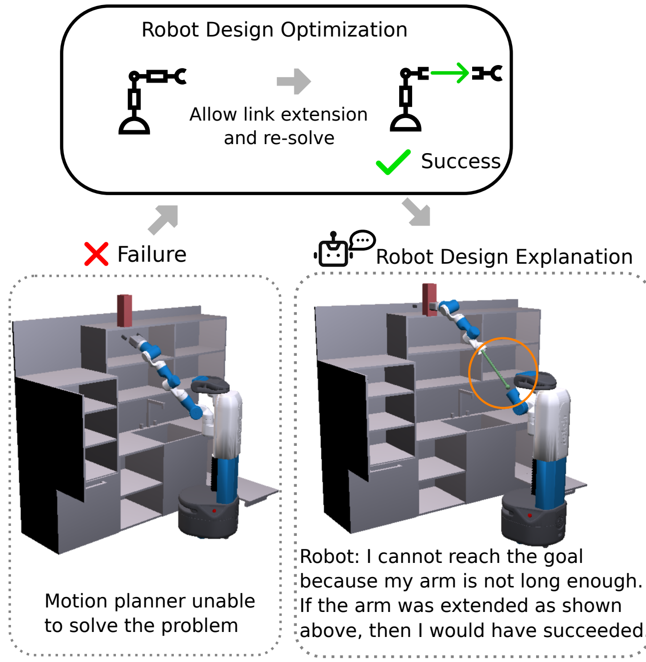


Fig. 1: Illustration of our "robot design explanation" method. The explanations show the minimum modification to the design that turns an infeasible motion planning problem into a success. When the motion planner fails, our method computes increases in link lengths that make the problem feasible, and uses them to compose an explanation as in the above example.

Our contributions are the following:

- We propose an algorithm that generates robot-design-based explanations for motion planning failures, by finding and visualizing the minimum change in robot design required to recover from the failure;
- We show that our method's explanations help users better understand robot capabilities and the cause of failure, and are perceived to be more actionable and satisfying than two other types of explanations.

II. RELATED WORK

The explainability of robotics algorithms has recently been a subject of research within the Explainable AI (XAI) community [2], [3]. Algorithms for generating motion planning explanations, in particular, have recently been proposed based on optimization methods [1], [4] and sampling-based methods [5], [6]. Explaining failures is an important type of explanation in motion planning [7], especially given that robot motion planners have been shown to be brittle and can easily fail to find a solution to simple problems [8]. Explanations of failure have been shown to enhance performance in human-robot-collaboration [9], human trust in robots [10], and failure recovery [11].

Some research has recently been made on algorithms for explaining failures caused by *objects* [12], [4]. However, here we focus on explaining failures based on the robot's *design*. Moreover, our method is based on a *contrastive* explanation, that contrasts the current design to a potential alternative. Contrastiveness has been shown to be an important property of explanations in AI [13] and robotics [7]. Such kind of explanations have been proposed for robotics in the past, for example in the path-planning explanation work of Brandao et al. [14] (which computes alternative terrain label or cost assignments in maps), or the autonomous driving commentary system of Kuhn et al. [15] (which computes a potentially contrastive text sentence to narrate vehicle behaviour, such as "car is stopping because the traffic light is not green"). In this paper, we focus on algorithms for generating explanations of motion planner failure—when motion planners fail to find a feasible solution to a problem.

To communicate robot motion planning failures to users, Kwon et al. [16] proposed a method that generates motion that mimics successful execution trajectories to help users identify the source of the failure. Other work proposed repeating failed behavior [17], and others have used large language models to generate text conveying failure cause [18], [19]. One limitation of these methods is that they are not actionable (i.e. do not tell the user what should be done to avoid failure), which is another desirable property of explanations [20]. This is a limitation that our method addresses by proposing explicit changes to robot design that would make a problem feasible. Explanations from our methods could be used, for example, by robot designers or mechanical engineers in order to improve future versions of the robot design.

Robot design optimization has been used to improve metrics such as trajectory length, joint torques, manipulability [21], [22], [23], [24], as well as sensor field-of-view [25]. While previous work has focused on design optimization to improve torque performance [26], [27] and perception algorithm performance [25], to the best of our knowledge, these methods have not yet been applied to the generation of explanations.

III. BACKGROUND

The objective of robot motion planning algorithms is to generate a trajectory according to the degrees of freedom of a robot, moving from an initial configuration $q_1 \in \mathcal{C}$ to a target configuration $q_N \in \mathcal{C}$, where N represents the number of time steps. The robot's configuration space $\mathcal{C}$ is typically $\mathbb{R}^D$, where D denotes the number of degrees of freedom (e.g., joint angles). Alternatively, $\mathcal{C}$ can be a composite space, e.g., joint angle space combined with $SE(3)$. The trajectory is a sequence of waypoints $\xi = \{q_1, \ldots, q_N\}$, and the motion planning task can be formulated as an optimization problem:

$$\begin{aligned}
\underset{\xi}{\text{minimize}} \quad & f(\xi) & & (1)\\
\text{s.t.} \quad & g_i(\xi) \leq 0 & & i = 1, \ldots, n_{\text{ineq}}\\
& \phi_i^{\text{ec}}(\xi) \leq 0 & & i = 1, \ldots, n_{\text{ec}}\\
& \phi_i^{\text{sc}}(\xi) \leq 0 & & i = 1, \ldots, n_{\text{sc}}\\
& h_i(\xi) = 0 & & i = 1, \ldots, n_{\text{eq}}
\end{aligned}$$

where f represents the objective function to be optimized (e.g., trajectory length and joint velocity). ϕ_i^{ec} are inequality constraints accounting for collisions between the robot and the environment, while ϕ_i^{sc} denote robot self-collision constraints, preventing potential collisions between different parts of the robot. The terms g_i correspond to other inequality constraints on the trajectory, such as joint angle limits, and h_i represent equality constraints, which include conditions like the initial configuration or the position of a robot joint or link. These functions are scalar-valued. One conventional objective function is trajectory velocity in configuration space:

$$f(\xi) = \sum_{i=1}^{N-1} \|q_{i+1} - q_i\|^2. \tag{2}$$

Methods such as Sequential Quadratic Programming, solved by optimization solvers, have been shown to obtain locally-optimal solutions to this problem satisfying all the constraints [28].

IV. METHOD

Let P be a motion planning problem that is infeasible, i.e. for which no solution can be obtained. We assume there is a design-related aspect of the robot that could be changed to make the problem feasible, specifically the length of its links, and which could be used as an explanation for the failure. For example, the explanation "the problem is infeasible because the robot's upper arm is too short" highlights which limits of the robot prevent it from completing the task. Building on findings from seminal eXplainable Motion Planning work [1], we assume this type of explanation could be useful either for mechanical engineers considering re-designs, or for lay users to better understand the robot's limits.

Our goal is then to convert an infeasible motion planning problem P into a feasible one P', by making minimum changes to the lengths of the links of the robot. We approximate the extension of the links' lengths by inserting virtual prismatic joints before each link in the kinematic chain, for all links, and then finding a solution to the new problem. In the context of this paper, therefore, the only difference between P and P' is the robot model, which has additional virtual joints in the latter case, and additional costs and constraints that minimize the amount of change between the two models.

Enabling Every Link to Extend: We insert a virtual prismatic joint j_v before every link whose parent is a revolute joint. This extends the configuration space to $X = \mathcal{C} \times \mathbb{R}^V_{0+}$, where V is the number of virtual pneumatic joints (i.e. virtual links allowed to extend). In practice, we use a script that takes a robot model in Unified Robotics Description Format (URDF), searches for revolute joints, and then (since URDF only allows link-joint connections) inserts a 0-length virtual link and a virtual pneumatic joint after each revolute joint found.

Optimization Problem: In order to compute a minimal change to the robot design that makes a problem feasible, we solve a new optimization problem. The new problem has both ξ and $\vartheta \in \mathbb{R}^{VN}$ as variables, and is defined as follows:

$$\begin{aligned} \underset{\xi,\vartheta}{\text{minimize}} \quad & f(\xi) + f(\theta) + \sum_{i=1}^{N} (\alpha f_1(\vartheta_i) + \beta f_2(\vartheta_i)) \qquad (3) \\ \text{s.t.} \quad & g_i(\xi) \leq 0 \qquad i = 1, \ldots, n_{\text{ineq}} \\ & \phi_i^{\text{ec}}(\xi) \leq 0 \qquad i = 1, \ldots, n_{\text{ec}} \\ & \phi_i^{\text{sc}}(\xi) \leq 0 \qquad i = 1, \ldots, n_{\text{sc}} \\ & h_i(\xi) = 0 \qquad i = 1, \ldots, n_{\text{eq}} \\ & \phi_{\text{ext}}(\vartheta) = 0 \end{aligned}$$

where f is a velocity cost as before, f_1 and f_2 are costs applied to virtual joints at each waypoint, and ϕ_{ext} is a constraint on virtual joints.

Extension Cost (f_1): We use this cost to limit the overall extension of links. Let $j = [j_1, ..., j_V]$ be the vector of virtual joint values, then we set:

$$f_1(j) = \sum_{k=1}^{V} j_k, \qquad (4)$$

since $j_k \in \mathbb{R}_{0+}$ are constrained to be non-negative by definition.

Length Distribution Cost (f_2): Most robots have redundant degrees of freedom, and therefore it will often be possible to make a problem feasible both by extending one link by a large amount, or multiple links by a smaller amount. We assume a larger extension on a single link produces clearer explanations for observers, and therefore devise a cost function which incentivizes solutions with fewer extended virtual joints:

$$f_2(j) = -\max\{j_1, \ldots, j_V\}. \qquad (5)$$

Since the $\max$ function is not differentiable, we use log-sum-exp as an infinitely differentiable approximation

$$f_2(j) = -\log\left(\sum_{k=1}^{V} e^{j_k}\right). \qquad (6)$$

There is a trade-off between f_1 and f_2, and in our experiments we prioritize f_1 as it penalizes the overall extension of the robot, and use f_2 to decrease the number of extended links. Therefore, we set $\alpha \gg \beta$.

Last-timestep Extension Constraint: For visualization purposes (i.e. when displaying the solution to this optimization problem *as a form of explanation*), one may choose to show the link extension required to solve the problem in all waypoints, or only at the last waypoint. We found last-waypoint-only visualizations to work better as explanations. In this setting, we construct a trajectory that attempts to reach the goal with the original design as much as possible, and only at the last waypoint will show necessary link extensions in case they would be needed to solve the problem. To achieve this effect, we set ϕ_{ext} so as to make all virtual joint values equal to zero at all waypoints except the last $\vartheta_i = 0, \forall i = 1, ..., N-1$. However, our approach is general and different constraints could be set so as to display the explanation in various ways.

Algorithm 1: Robot Design Optimization

Data: trajectory initialization ξ, robot model r, start configurations s, cost function f, penalty factor k, penalty coefficient μ

for *joint = 1,2, ... maxJointNumber* **do**
  if *joint is revolute* **then**
    $r \leftarrow r \cup$ insert virtual link and prismatic joint;
for *penaltyIteration = 1, 2, ...* **do**
  for *optimizerIteration = 1, 2, ...* **do**
    $\xi, \vartheta \leftarrow$ SQP iteration to (3) ;
    if *converged* **then**
      break ;
  if *constraints satisfied* **then**
    return ξ, ϑ, r ;
  else
    $k \leftarrow \mu * k$;

Computing an explanation: Pseudo-code summarizing our method for computing an explanation of failure is shown in Alg. 1. First, we compute a new robot model with additional virtual links and joints, which allows the robot to extend its links. Then, we solve the new problem using TrajOpt [28], which is based on SQP and penalty iteration. The result is a trajectory and new robot model, which will be shown to a user to show design-related reasons for the planner's failure in the original problem. It will show a trajectory moving the robot as close as possible to its goal, and then an extension of its links required to make the problem feasible. An accompanying text template can be filled to display an automated text message, such as: "The planner failed because the links highlighted in green are not long enough, as shown in this animation". We show examples in Fig. 3, which will be discussed in the next section.

V. RESULTS

To evaluate the performance of our method, we first generated a set of unsolvable problems for which an explanation of failure needs to be computed. We used MotionBenchMaker [29] to generate a set of unsolvable problems in the kitchen and shelf scenarios from the library. The problems represent a variety of failure modes: from out-of-reach grasping locations to obstacle occlusions, as shown in Fig. 2.

We compare our method (**"robot design explanation"**) with two baselines:

- **Robot incapability expression**, a method proposed by Kwon et al. [16], which relaxes goal constraints into cost functions in order to display motion that makes as much progress towards the goal as possible while respecting physical and collision constraints. Explanations generated by this method are implicit in the generated motion, though they can be accompanied with a textual explanation of the form: "the planner failed because the goal object was out of the robot's reach. The video shows how the robot tries its best to reach there but fails".
- **Joint limit explanation**, a method that checks whether a problem becomes feasible once joint-limit constraints are removed from the optimization problem. We propose this method as a comparison where all the joint limits can be removed to show users the effect of these limits. Explanations take the following form: "The planner failed because of limits on the angle/distance of joints X. If these limits were lifted then the planner would have solved the problem as shown in the following animation", and the method can display the computed trajectory which does not respect joint limits.

We solve all optimization problems using TrajOpt [28] with Gurobi as the solver. We use a constraint tolerance of 10^{-4}, and consider that TrajOpt has converged when the cost improvement between two iterations is lower than 10^{-4}. When no solution is found after 50 iterations, it is considered a failure. Our experiments use $\alpha = 13$ and $\beta = 2$.

A. Qualitative results

Fig. 3 shows examples of robot design explanations in 4 scenes in which the robot needs to grasp the red box on top of the cupboard. The robot is unable to reach the goal pose in its original design, and our explanations show which links are responsible for the failure because they are too short. The example images show green arrows on top of robot links that, if modified to be longer, would enable the robot to complete the task.

Fig. 4 compares the explanations generated by our method and the two baseline methods for the same problem. Robot incapability expression shows that the planner fails because it cannot reach the target, by displaying motion of the robot attempting to reach the goal as much as it can, but failing. The joint-limit explanation shows that the planner fails because the robot is constrained by limits on the torso joint, and displays the motion that the robot would do if the torso's prismatic joint was not limited. Our robot design explanation, on the other hand, shows that the planner fails because the arm is not long enough, and displays how arm links could be extended to complete this task. If the robot had been designed that way, then it would have been able to complete the task, as shown in Fig. 4 (c). Both the joint-limit and robot-design explanations are "actionable", a desirable property of explanations [20], in the sense that they inform the user about what could be done to make the problem feasible. However, design explanations directly inform how the design can be changed (i.e. by extending a link's length), while joint-limit explanations may be harder to implement and visualize as they require thinking about changes in the mechanical design that would allow to extend the joint limits.

Although each of the three explanation methods focuses on a different aspect of the problem that is related to infeasibility, they all generate *faithful* explanations, i.e. all explanations are simultaneously true. The planner failed both because it cannot reach the goal, because its joint limits keep it from reaching the goal, and because the length of some of its links is too short to be able to reach the goal. However, as we will see in Section V-B, users may find some explanations more or less satisfying, actionable, and leading to a better understanding of robot's capabilities than others.

Additionally, some explanations may also be applicable to more problems than others. For example, for the Fetch robot in our experiments, the joint-limit explanations extend the configuration space by a much smaller amount than design explanations. Although removing joint limits can increase reachability on the vertical axis, as seen on Fig. 4(b), reachability in other axes does not get enhanced. Design explanations, on the other hand, allow links to extend and thus for the robot to reach in locations that are further away.

B. User evaluation

We evaluated our explanations with a user study with 20 participants. 70% of the participants were holders of an undergraduate degree in Computer Science or Engineering, 60% of the participants were female and 40% were male. Participants saw 9 problems (6 on kitchen scenes and 3 on shelf scenes) where the robot fails to grasp the goal object. For each problem participants saw the 3 different explanations of failure in random order. Each explanation showed both the template text and an animation (e.g. "the planner failed because links X are not long enough, as shown in the following animation").

1) Effectiveness and actionability: In order to measure whether explanations helped users understand the intention of the robot (what it was trying to do), the cause of failure (why it failed), and the recovery approach (what it would take to make it succeed), we asked participants to rank the following three statements using a 5-Likert scale ("strongly disagree - 0 " to "strongly agree - 5"):

1) It was clear that the robot failed because [cause]
2) It was easy to tell the robot was trying to do [goal]
3) It was clear how [changes] can be made to enable the robot to succeed in this task.

The user responses from the explanation survey in Fig. 5 shows the average score of how much the participants agree or disagree with the statement in Likert scale. We ran a Shapiro-Wilk test to check the normality of the data, and found that answers to the above questions were not normally distributed. Therefore we used Wilcoxon rank-sum test, which is a non-parametric statistical significance test. We use star symbols

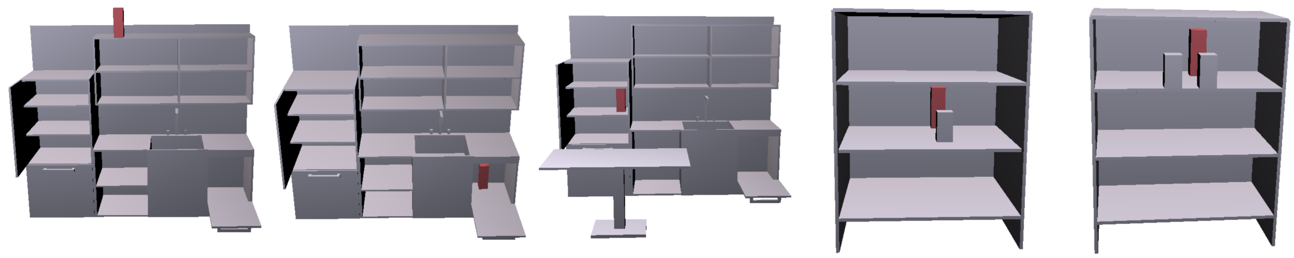
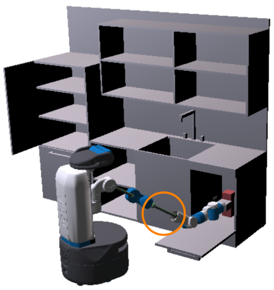
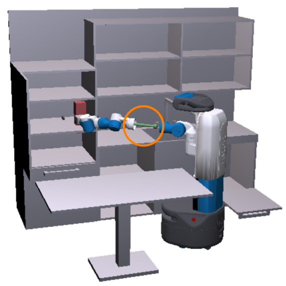


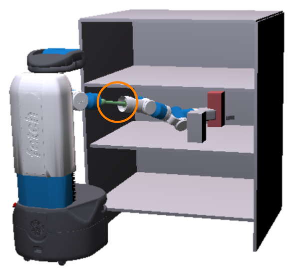
Fig. 2: Examples of infeasible problems for the kitchen scene and shelf scene. The problems require the robot to move its end-effector to the red box.



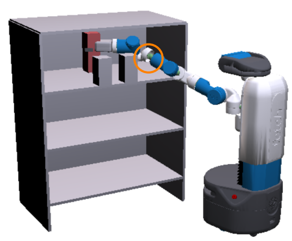
(a) kitchen scene (easy)



(b) kitchen scene (hard)

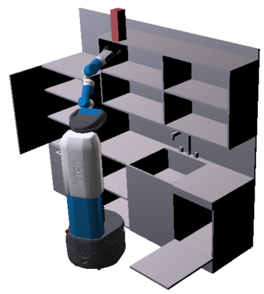


(c) shelf (easy)

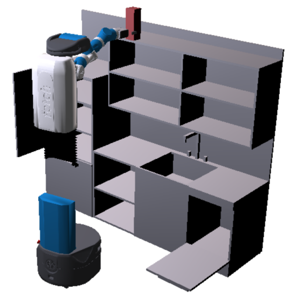


(d) shelf (hard)

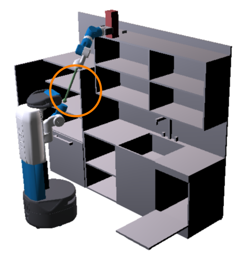
Fig. 3: Examples of robot design explanations in 4 tasks: a) grasping an object in the dishwasher; b) in a cupboard with a table in the middle; c) behind an obstacle on a shelf; d) behind two obstacles on a shelf. Explanation: "The robot failed to complete this task because the arm was not long enough. The video shows how the arm can be re-designed longer (the green arrow circled) to help it succeed.".



(a) Robot incapability expression [16].



(b) Joint-limit explanation.



(c) Robot design explanation (ours)

Fig. 4: Examples of the three types of explanations considered. (a) "Robot incapability expression" [16]: the robot tries to lift its gripper as high as possible to approach the goal object as an imitation of a successful trajectory; (b) "Joint-limit explanation": Because all limits on joints are removed, the prismatic joint on the torso lifts it up to reach the object; (c) "Robot design explanation" (ours): the robot arm is extended to reach the goal, highlighting a design-based reason of failure.

in figures to indicate statistical significance using a post-hoc Wilcoxon Signed Rank test with a Bonferroni correction due to multiple comparisons between 3 conditions. The original significance level is $\alpha = 0.05$, which after a Bonferroni correction leads to a significance level of $\alpha_{corrected} = .0167$ (i.e. we use * to indicate $p < 1.67 \times 10^{-2}$ in Fig. 5 and 6).

Fig. 5 shows that the perception of clarity of the cause of failure was not significantly different between the three methods, even if our method has a slightly higher average. Participants found that design explanations made it easier to understand what the robot was trying to do (average 4.32, "agree"), with a score significantly higher than incapability expressions and joint-limit explanations ($p = 0.0008$ and 0.0012, respectively). We believe that the low result for incapability expression originates from the end-effector not reaching the goal object by design, which may make it hard to understand what the robot was trying to do. The actionability of our method (i.e. clarity of changes that can be made to make the problem feasible), was also significantly higher than both baselines ($p = 0.0010$ compared with incapability expression and 0.0007 compared with joint-limit explanation). We believe this is because the incapability expression method does not indicate how to recover from the failure, while joint-limit explanations can be hard to understand especially for revolute joints (i.e. it is hard to tell whether and by how much a joint limit is being extended).

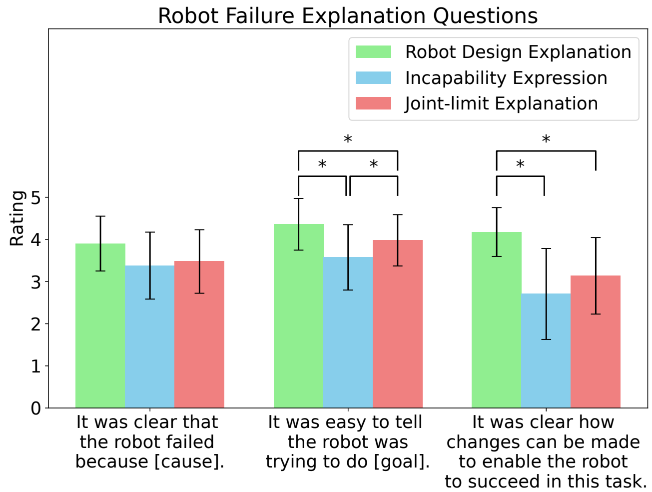


Fig. 5: User ratings of explanations on failure cause, robot intention and recovery methods.

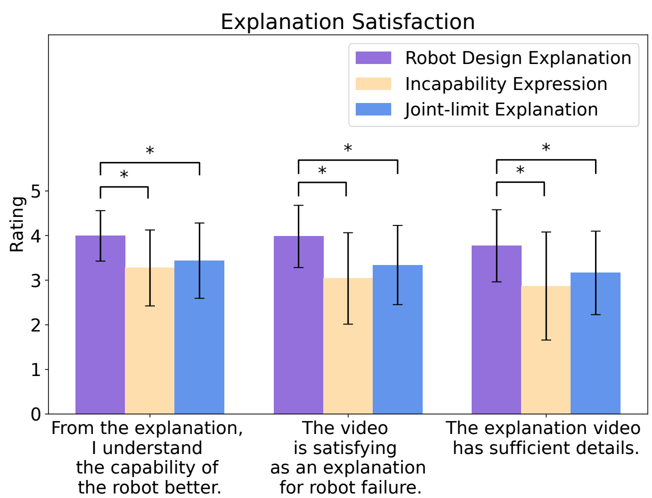


Fig. 6: User ratings of satisfaction on the 3 types of explanations.

2) Satisfaction: We also evaluated users' satisfaction with the explanations. To measure satisfaction we adapted the XAI Explanation Satisfaction Scale questions proposed by Hoffman et al. [30] to our setting. Concretely, we used the "understanding", "satisfying" and "sufficient details" questions of the scale, formulated as follows:

1) From the explanation, I understand the capability of the robot better.
2) The video is satisfying as an explanation for robot failure.
3) The explanation video has sufficient details.

Fig. 6 shows the user satisfaction with the three types of failure explanations. Overall, participants found robot design explanations more satisfying than the other two. Our explanations had higher scores than others on all questions. First, the figure shows that the robot design explanation helped users understand the capabilities of the robot better than the other two methods (statistically significant with $p = 0.0020$ and 0.0024). Second, users found our method the most satisfying explanation among the three (statistically significant with $p = 0.012$ and 0.0009). This is because the robot design explanation shows the lack of robot capability in an intuitive way. It contains information of the robot's intention, failure cause, and recovery method.

Finally, the amount of detail in the explanation was also scored higher on average for our method. The difference was statistically significant when comparing to incapability expression ($p = 0.0024$) and joint-limit explanations ($p = 0.0014$). Although both the robot design and joint-limit explanations show a similar amount of information (degree of change that can be made), the type of changes is different (joint-limit or link-length change).

Users could also write open-text feedback at the end of the survey. Participants mentioned that sometimes it can be difficult to know where the target object is, and whether the robot has succeeded or failed to reach it. Therefore, we plan to improve communication of the goal and constraint satisfaction in future studies. Some also suggested that indicating the grasping radius of the gripper could make it easier for them to identify the capability to grasp the object when the gripper is in the vicinity of the object.

VI. CONCLUSIONS

We proposed a new type of motion planning explanation based on robot design. Our algorithm computes the minimum change in robot design that converts a failed task into a success. In particular we formulate the explanation problem as a new trajectory optimization problem with virtual pneumatic joints that allow robot links to "extend", and cost functions that promote small extensions over few links and an intuitive visualization. We conducted a user study to compare our explanation method with two baselines, one of them from the recent literature. The results showed that our method helped users better understand the robot's capabilities and cause of failure. Furthermore, users were more satisfied with the explanations generated by our method than those from the baselines.

One limitation of this work is that the extended links do not have geometry, therefore potentially colliding with obstacles. In the future, we would also like to extend the current method to more complex design changes. As suggested from user feedback on the survey, it is also important to combine explainability methods with good communication methods, such as verbal communication, visual indicators, or interactive tools that can be used alongside explanations to further improve the user understanding of robot behavior and capabilities.

REFERENCES

[1] M. Brandão, G. Canal, S. Krivić, and D. Magazzeni, "Towards providing explanations for robot motion planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 3927–3933.

[2] A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli, "Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–18. [Online]. Available: https://doi.org/10.1145/3173574.3174156

[3] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253519308103

[4] Q. Liu and M. Brandão, "Generating environment-based explanations of motion planner failure: Evolutionary and joint-optimization algorithms," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 15 263–15 269.

[5] K. Hauser, "The minimum constraint removal problem with three robotics applications," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 5–17, 2014. [Online]. Available: https://doi.org/10.1177/0278364913507795

[6] K. Molloy, L. Denarie, M. Vaisset, T. Simeon, and J. Cortés, "Simultaneous System Design and Path Planning: A Sampling-based Algorithm." *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 375–387, Mar. 2019. [Online]. Available: https://laas.hal.science/hal-01834414

[7] M. Brandao, M. Mansouri, A. Mohammed, P. Luff, and A. Coles, "Explainability in multi-agent path/motion planning: User-study-driven taxonomy and requirements," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2022, p. 172–180.

[8] W. Wu, F. Pierazzi, Y. Du, and M. Brandão, "Characterizing physical adversarial attacks on robot motion planners," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 14 319–14 325.

[9] A. D. Dragan, S. Bauman, J. Forlizzi, and S. S. Srinivasa, "Effects of robot motion on human-robot collaboration," in *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2015, pp. 51–58.

[10] S. Ye, G. Neville, M. Schrum, M. Gombolay, S. Chernova, and A. Howard, "Human trust after robot mistakes: Study of the effects of different forms of robot communication," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2019, pp. 1–7.

[11] D. Das, S. Banerjee, and S. Chernova, "Explainable ai for robot failures: Generating explanations that improve user assistance in fault recovery," in *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 351–360. [Online]. Available: https://doi.org/10.1145/3434073.3444657

[12] M. Diehl and K. Ramirez-Amaro, "A causal-based approach to explain, predict and prevent failures in robotic tasks," 2022. [Online]. Available: https://arxiv.org/abs/2209.05255

[13] T. Chakraborti, S. Sreedharan, and S. Kambhampati, "The emerging landscape of explainable automated planning & decision making," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 4803–4811, survey track. [Online]. Available: https://doi.org/10.24963/ijcai.2020/669

[14] M. Brandão, A. Coles, and D. Magazzeni, "Explaining path plan optimality: Fast explanation methods for navigation meshes using full and incremental inverse optimization," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 31, no. 1, pp. 56–64, May 2021. [Online]. Available: https://ojs.aaai.org/index.php/ICAPS/article/view/15947

[15] M. A. Kühn, D. Omeiza, and L. Kunze, "Textual explanations for automated commentary driving," in *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2023, pp. 1–6.

[16] M. Kwon, S. H. Huang, and A. D. Dragan, "Expressing robot incapability," in *2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2018, pp. 87–95.

[17] M. Nicolescu and M. Mataric, "Learning and interacting in human-robot domains," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 31, no. 5, pp. 419–430, 2001.

[18] Z. Liu, A. Bahety, and S. Song, "Reflect: Summarizing robot experiences for failure explanation and correction," 2023. [Online]. Available: https://arxiv.org/abs/2306.15724

[19] C. Tagliamonte, D. Maccaline, G. LeMasurier, and H. A. Yanco, "A generalizable architecture for explaining robot failures using behavior trees and large language models," in *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1038–1042. [Online]. Available: https://doi.org/10.1145/3610978.3640551

[20] M. Brandao, G. Canal, S. Krivic, P. Luff, and A. Coles, "How experts explain motion planner output: a preliminary user-study to inform the design of explainable planners," in *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, Aug 2021, pp. 299–306.

[21] M. Toussaint, J.-S. Ha, and O. S. Oguz, "Co-optimizing robot, environment, and tool design via joint manipulation planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6600–6606.

[22] J. Whitman and H. Choset, "Task-specific manipulator design and trajectory synthesis," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 301–308, 2019.

[23] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence*, vol. 267, pp. 1–38, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370218305988

[24] M. Lei, E. Romiti, A. Laurenzi, and N. G. Tsagarakis, "Task-driven computational framework for simultaneously optimizing design and mounted pose of modular reconfigurable manipulators," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 4563–4570.

[25] M. Brandao, R. Figueiredo, K. Takagi, A. Bernardino, K. Hashimoto, and A. Takanishi, "Placing and scheduling many depth sensors for wide coverage and efficient mapping in versatile legged robots," *The International Journal of Robotics Research (IJRR)*, vol. 39, no. 4, pp. 431–460, 2020.

[26] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Computational co-optimization of design parameters and motion trajectories for robotic systems," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1521–1536, 2018. [Online]. Available: https://doi.org/10.1177/0278364918771172

[27] O. Taylor and A. Rodriguez, "Optimal shape and motion planning for dynamic planar manipulation," *Autonomous Robots*, vol. 43, pp. 327–344, 2019.

[28] J. Schulman, J. Ho, A. X. Lee, I. Awwal, H. Bradlow, and P. Abbeel, "Finding locally optimal, collision-free trajectories with sequential convex optimization." in *Robotics: science and systems*, vol. 9, no. 1. Berlin, Germany, 2013, pp. 1–10.

[29] C. Chamzas, C. Quintero-Peña, Z. Kingston, A. Orthey, D. Rakita, M. Gleicher, M. Toussaint, and L. E. Kavraki, "Motionbenchmaker: A tool to generate and benchmark motion planning datasets," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, p. 882–889, Apr. 2022.

[30] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, "Metrics for explainable ai: Challenges and prospects," 2019. [Online]. Available: https://arxiv.org/abs/1812.04608