Feasibility Checks for Safe Shared Control with Variable Autonomy in Assistive Robotics

Samuel Bustamante*, Ismael Rodríguez*, Gabriel Quere, Daniel Leidner, Jörn Vogel, Freek Stulp

***Abstract*— In robotics, especially on systems that interact with humans, there is an increasing need for adaptable levels of autonomy. These include direct control, shared control, supervised autonomy, and full autonomy. A concrete example of this is the wheelchair-based robot EDAN, where a user may want to initiate the opening of a door in shared control, but then let the system autonomously cross the doorway.**

It is essential that robot guidance in shared control is robust and helps the human to prevent errors, rather than causing them by guiding the user towards solutions that are ultimately infeasible due to obstacles or limited manipulability. In this letter, we describe a proof-of-concept for fast and iteratively-refined feasibility checks in the context of EDAN. We conduct exploratory experiments in a grasping action with obstacles.

I. Motivation

EDAN (our *electromyography-controlled daily assistant*) uses Constraint-Action Templates (CATs,[1]), an action representation that provides the user with supervised autonomy, shared control (using an interface like a 3D joystick or electromyography [2]), or a mixture of both [3]. In particular, CATs contain declarative knowledge (PDDL) to plan a task with atomic actions, as well as procedural knowledge (operations) on how to execute them. CAT operations produce manifolds constraining the space of motion. As an example, a cylindric container can, in principle, be approached from an arbitrary angle, as illustrated in Fig. 1. The `_container.pick` action manifold therefore leads to feasible grasps for arbitrary angles. CATs exploit such symmetries for shared control, by allowing the user to choose from where the grasp is performed while steering the robot.

However, the formulation of CATs in [1] is prone to errors. In principle, the container can be approached from any angle, but in practice, these feasible approaches also depend on obstacles and manipulability restrictions of the robot arm. In some scenarios the task may not be feasible *at all*. The aim of this paper is to take such factors into account by doing feasibility checks on trajectories wrt. manipulability and obstacles, to ensure that shared control guidance will function in practice, not only in principle. We conduct an exploratory experiment for a partially-feasible task, illustrated in Fig. 1 (right). Our approach allows to augment new constraints to keep the user in a safe zone and to ease the operation of the system. We believe these constraints will help users as they do not need to take care of collisions with the environment, specially with noisy or low-bandwidth interfaces like electromyography.

* Equal contribution. All authors are with the German Aerospace Center (DLR), Robotics and Mechatronics Center (RMC), Münchner Str. 20, 82234 Weßling, Germany. This work is partly supported by the German Research Foundation (DFG) within the Collaborative Research Center EASE (SFB 1320) and the Bavarian Ministry of Economic Affairs, Regional Development and Energy (StMWi) by means of the project SMiLE2gether (LABAY102). The authors would like to thank Annette Hagengruber and Peter Lehner for continual support and motivating discussions.

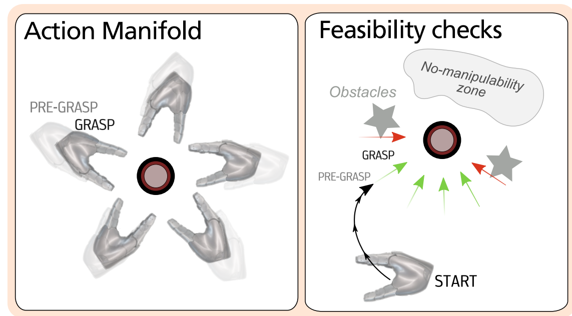

Fig. 1. Feasibility checks in a grasping task. **Left:** CATs define a manifold of motion for the task. **Right:** We aim to add constraints to this manifold to create a zone safe of obstacles and manipulability issues.

II. Related Work

Feasibility checks have been explored in other domains such as manufacturing. Iteratively-refined feasibility checks on task and motion plans [4] increase the robustness of task execution, but have not yet been used for shared control. In comparison, feasibility checks for shared control should be as fast as possible, since the human is waiting; additionally robot decisions should be legible.

Previous work in that domain include [5] for an optimization-based motion planner to plan a shared control task and [6] for a reactive framework that adapts constraints online to prevent a collision. Our work extends this line of research by explicitly planning restrictions in a manifold of motion. [7] also does so, but only final grasp poses are checked and not paths in between.

III. Feasibility Checks

We add feasibility checks using a kinematic simulation of EDAN. With these we can compute constraints to action manifolds from any starting situation, like when the user starts a task with a UI. They could also be pre-computed when the robot estimates a task is likely to be started.

Overview: An overview of the approach is shown in Fig. 2. CATs are used for hybrid planning: first, a symbolic task planner decides the overall sequence of actions; then, a motion-constraints planner defines so-called *action manifolds*. That means, it does not output a single path, as conventional motion planners do, but a set of constraints in task space that can be used to generate motions in shared control, supervised autonomy, or both [3], [1]. Its input is an

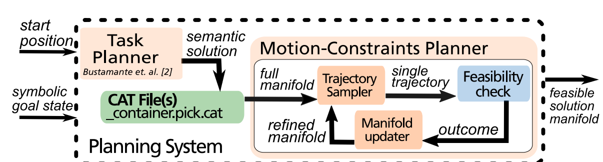

Fig. 2. Our concept for a hybrid planning system.

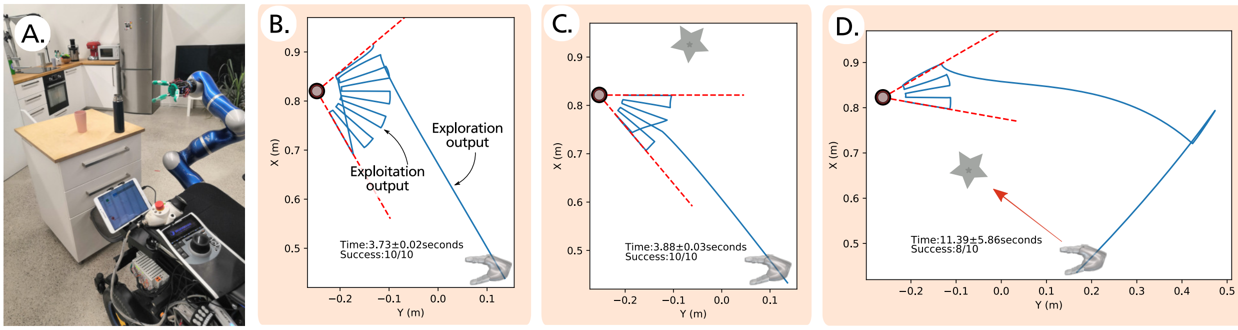

Fig. 3. Feasibility checks featuring simulated trajectories in the context of EDAN (**A.**). We tested different scenarios with (**B.**) no obstacles, (**C.**) an obstacle (gray star) in the periphery of the manifold, and (**D.**) an obstacle in the direct way to the target. We report the motion planning times ($\mu \pm \sigma$) over 10 trials as well as the number of times the algorithm found a manifold; the resulting ranges are shown in a red dotted line.

object-centric definition of the action provided by a CAT file like `_container.pick`. The planner samples and tests single trajectories in simulation, obtaining a boolean indicating if the trajectory is feasible and does not result in collisions. This entails a feedback algorithm, updating the manifold after every check and improving the specificity of future samples.

Implementation on a grasping task: In order to keep planning times low, the trajectory sampler contains two phases: first, in an *exploration* phase it aims to connect the starting position with action-related poses from the manifold, such as pre-grasps and grasps. The manifold is sampled exhaustively (and updated after each sample) until *one feasible trajectory* is found. If it exists, the exhaustive search stops, and the planner (in an *exploitation* phase) searches locally around it, attempting to connect further pre-grasp and grasp poses only. This continues until limits are found, and the plan is final if the range is larger than a pre-set threshold. If not, the exploration phase is restarted. Example trajectories are shown in Fig. 3B/C.

Geometric backtracking: Connections in either phase are attempted using a fast local numerical method for the inverse kinematics that can compute neither reconfigurations nor large motions. This may work in many situations, but may fail in complex scenarios. Upon failure, our planner restarts the exploration phase using a global inverse kinematics search, which will take longer time and could force the robot to reconfigure from its initial position (as in Fig. 3D).

IV. Experiments

The task consists in obtaining the feasible action manifold to grasp the red mug placed in front of EDAN, as shown in Fig. 3A. We considered three scenarios with a bottle as an obstacle, also shown in Fig. 3B-D. The starting position of EDAN and the mug (but not the obstacle) were fixed, and the objects were localized with perception. Finally, the motion plans were generated and executed in simulation.

Discussion: As shown in Fig. 3, feasible action manifolds could be generated. The approach generalizes from a purely manipulability manifold (Fig. 3B) to regions that avoid obstacles (C. and D., subsets of B.). The motion planning is relatively fast, converging in less than 4 seconds in B. and C. In D. the robot exhausts the local options and thus requires a plan with reconfiguration, which takes longer to compute. However, such a situation could have lead to a task failure directly without a feasibility check [3]. It highlights the advantages of geometric backtracking: a reconfiguration is only sought after if a local search is not successful; this decreases planning time in common situations such as B. while maintaining robustness for scenarios such as D.

V. Next Steps

In the future, we plan to test our approach in more complex scenarios, such as tasks composed of a sequence of actions. While we considered a grasping task for simplicity, CATs support other daily living actions with extended durations [1]. We will do experiments with target users with motor impairments, exploring how the robot may convey its current strategy and explain the reasoning behind its plans. As it remains a question whether users will accept larger planning times as a trade-off for more robustness, we will also explore planning strategies that do not interrupt the robot movement.

Finally, we plan to extend the feasibility checks by adding the capability to backtrack not only in the geometric level but also on the symbolic task plan, which gives the robot the chance to replan actions when one is completely unfeasible. For example, in a cluttered environment EDAN may decide to move the wheelchair to a better position or to schedule a pick-and-place action for an obstacle. Based on this we may create semantic guidelines on how the robot should behave in certain scenarios and reuse this information for future plans (similar to in [8] for the manufacturing domain).

References

[1] S. Bustamante et. al., "CATs: Task Planning for Shared Control of Assistive Robots with Variable Autonomy," *Accepted to the 2022 IEEE International Conference on Robotics and Automation (ICRA)*.

[2] J. Vogel et. al., "An ecosystem for heterogeneous robotic assistants in caregiving: Core functionalities and use cases," *IEEE Robotics Automation Magazine*, vol. 28, no. 3, pp. 12–28, 2021.

[3] S. Bustamante et. al., "Toward seamless transitions between shared control and supervised autonomy in robotic assistance," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3833–3840, 2021.

[4] I. Rodríguez et. al., "Iteratively Refined Feasibility Checks in Robotic Assembly Sequence Planning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1416–1423, Apr. 2019.

[5] A. Dragan and S Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, Jun. 2013.

[6] S. Iregui et. al., "Reconfigurable constraint-based reactive framework for assistive robotics with adaptable levels of autonomy," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7397–7405, 2021.

[7] K. Muelling et. al., "Autonomy infused teleoperation with application to brain computer interface controlled manipulation," *Autonomous Robots*, vol. 41, no. 6, pp. 1401–1422, Aug. 2017.

[8] I. Rodríguez et. al., "Pattern recognition for knowledge transfer in robotic assembly sequence planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3666–3673, 2020.