Progress and Challenges in Planning for a Two-Arm Robot*

Lars Karlsson[1] Julien Bidot[1] Fabien Lagriffoul[1] Alessandro Saffiotti[1] Ulrich Hillenbrand[2] Florian Schmidt[2]

The robot Justin, which has been developed at the institute of Robotics and Mechatronics at the German Aerospace Center (DLR) in Oberpfaffenhofen, is an advanced humanoid robot with two arms (7 DOF each) and four-fingered human-like hands (12 DOF each), a head with two video cameras for stereo vision, an actuated torso, and a base with four wheels mounted on extensible legs.

In this poster, we present the ongoing efforts to provide Justin with a higher degree of autonomy within the scope of the EU FP7-project GeRT (see `http://www.gert-project.eu`). Justin was until recently limited to performing tasks using hand-written programs for specific manipulation tasks and involving a priori known objects. The overall aim of the project is that Justin should be able to generalize from existing programs to perform new tasks consisting of the same types of basic operations but combined in new ways, and with new objects belonging to the same functional classes.

Among the issues addressed in the project is task planning: the problem of finding a sequence of actions (a plan) that achieves a given goal when started from some initial state. Task planning can solve tasks that involve complex causal dependencies between actions and supports high-level specifications of those tasks. However, actions and states are typically represented in logical form, which is insufficient for representing the geometrical and kinematic properties of the robot, the objects it manipulates, and the environment. Therefore, generated plans may sometimes be impossible to be executed due to kinematic constraints and/or obstacles. For this reason, we have integrated the task planner with an RRT path planner, and augmented the logical states with a geometric state component which explicitly represents the configurations of the robot and other objects.

The current planning system for GeRT [1] can compute an initial state based on perceptual data, provided it has a priori models of the objects in the scene. It then calls an extended version of a standard task planner, which can check preconditions and establish effects both in the logical and the geometric state components. Thus, during search the planner keeps track of the configuration of the robot and other objects, and can verify that viable paths exist for transit and transfer actions. In addition, the planner is capable of backtracking on the geometric level. Typically, an action such as putting down an object in a region can be performed in many ways on the geometric level, although it is still considered as the same action on the logical level. If for instance the planner chooses to put down an object in a certain position which later turns out to be an obstacle for another object, it can go back and reconsider the position of the first object. Once a plan is found, it is converted to a sequence of Python programs (each action type is associated with a code template). These are then executed on Justin. The paths computed by the RRT planner during planning are polished by a path smoother before execution. A number of runs on Justin and in simulation have been performed with different tasks involving a small number of objects (1 to 4) in order to demonstrate the feasibility of the approach and the utility of geometric backtracking [1]. These tasks include placing several cans on a tray that is just large enough to fit them all, and moving cups in the presence of a large obstacle (box) which may require changing hands.

But many challenges remain. For instance, the planner presently uses a set of grasps for each object that have been given as explicit examples for that particular object. The grasp contains information about the configuration for each finger and the pose of the tool center position (a point roughly at the wrist). Thus, with the present system the robot would not be able to grasp objects of which it does not have models and grasps sets in advance. For a novel object, we plan to use the grasps associated with a previously known object of the same functional class as a starting point; then, we warp the contact points of the known object onto the new object and compute new grasps [2], [3].

A second challenge is to be able to learn automatically planning domains from hand-written example programs. Within GeRT, an initial approach to learning bidirectional mappings between predicates and geometric states as well as actions with pre- and postconditions has been developed [4], but not yet tested in the context of the integrated task-path planner.

References

[1] L. Karlsson, J. Bidot, F. Lagriffoul, A. Saffiotti, U. Hillenbrand, and F. Schmidt, "Combining task and path planning for a humanoid two-arm robotic system," in *Proceedings of the ICAPS 2012 workshop "Combining Task and Motion Planning for Real-World Applications"*, 2012, pp. 13–20.

[2] U. Hillenbrand and M. A. Roa, "Transferring functional grasps through contact warping and local replanning," in [5].

[3] H. Ben Amor, O. Kroemer, U. Hillenbrand, G. Neumann, and J. Peters, "Generalization of human grasping for multi-fingered robot hands," in [5].

[4] C. Burbridge, Z. Saigol, F. Schmidt, C. Borst, and R. Dearden, "Learning operators for manipulation planning," in [5].

[5] *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vila Moura, Algarve, Portugal, Oct. 2012.

*The project receives funding from the European Community's Seventh Framework Programme under grant agreement no. ICT- 248273 GeRT.

[1] Center for Applied Autonomous Systems, Örebro University, Sweden
[2] German Aerospace Center (DLR), Oberpfaffenhofen, Germany