

Circumventing Robots' Failures by Embracing Their Faults: A Practical Approach to Planning for Autonomous Construction

Stefan Witwicki and Francesco Mondada

Laboratoire de Systèmes Robotiques (LSRO)

Swiss Federal Institute of Technology (EPFL)

CH-1015 Lausanne, Switzerland

{stefan.witwicki, francesco.mondada} @epfl.ch

Abstract

This paper overviews our application of state-of-the-art automated planning algorithms to real mobile robots performing an autonomous construction task, a domain in which robots are prone to faults. We describe how embracing these faults leads to better representations and smarter planning, allowing robots with limited precision to avoid catastrophic failures and succeed in intricate constructions.

Introduction

Along with recent algorithmic advances in planning comes the expectation that bigger and more realistic problems can and will be solved by autonomous agents. However, when those agents take the form of robots interacting with the physical world, results tend to fall short of expectations. Faults inevitably enter into the robots' behavior. For mobile robots that are autonomously constructing a wall, building blocks are sometimes incorrectly grasped or imprecisely placed (Wisner et al. 2012). And one may be tempted to blame the faults on the robot's hardware or its low-level controllers, or to dismiss them as rare implications of imperfect environmental conditions.

Here, we advocate a different perspective. Why not accept the robots as imperfect agents? Why not acknowledge that, no matter the investment in expensive high-precision sensors, regardless of the meticulous efforts spent in designing robust controllers, when we put them into complex real-world environments, robots can and will make mistakes. In adopting this somewhat unconventional view of robots, it becomes necessary to explore alternative approaches to the classical application of automated planning.

Autonomous Wall Construction

Displayed in Figure 1 is the the marXbot (Bonani et al. 2010), a small mobile robot with a modular, reconfigurable design, allowing it be optionally equipped with a variety of different modules supporting various sensory and actuation capabilities, some of which are highlighted. These modules are seamlessly integrated by a CAN bus connected to an ARMv6 board running Linux. The magnetic gripper module is particularly useful for autonomous construction;

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

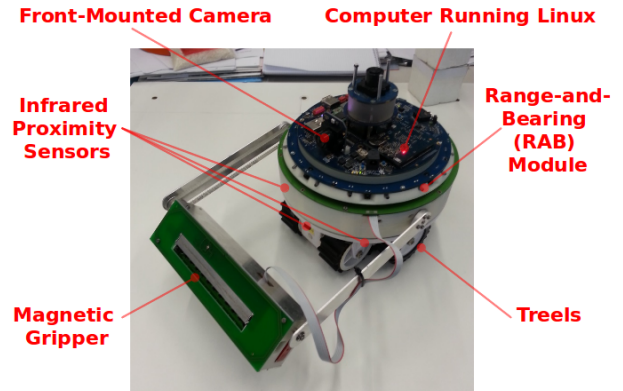


Figure 1: A marXbot equipped for autonomous construction.

it is designed with three degrees of freedom (lift, rotation around the robot, and tilt of the end-effector), and it has a switchable magnetic device that allows the robot to grasp and release building blocks to which we have affixed thin metallic strips. In addition, it has two linear arrays of short-range IR sensors mounted on the front and the bottom of the end-effector, providing primitive vision of the object being manipulated. By rotating the gripper to its side and moving forward with its treels, the marXbot can perform lateral scans and align with whatever it is constructing.

Here, the task is to autonomously construct a wall (Figure 2) from a blueprint using small (6cm) cubes. This task is made challenging by the fact that, in contrast to related work (Wisner et al. 2012; Petersen, Nagpal, and Werfel 2011), the construction blocks are not self aligning. Moreover, the marXbot's primary mode of viewing the wall is highly sensitive to lighting conditions, is imprecise outside of a very limited range, and is prone to error due to infrared dispersion. In light of these quirks, and despite our best efforts in designing robust controllers, faults tend to arise during the construction process. For instance, the robot fails to align with the wall because one block is crooked, holes appear in the wall due to an accumulation of misalignments, or a block is placed in a way that disturbs other blocks on the wall. Our objective is for robots to use planning to anticipate and correct such faults, thereby preventing catastrophic failures.



Figure 2: A wall of height 3 cubes and length 3 cubes.

Hierarchical Representation for Robot Control

In support of a variety of different planning frameworks, we have developed a hierarchical feature representation for the autonomous construction problem, partially illustrated in Figure 3. At the lowest levels are the raw sensor readings. These serve as inputs to sensory processing routines running on the microcontrollers, in turn forming feature abstractions that feed into a hierarchy of subtasks. Moving up this hierarchy, subtasks compose into sophisticated activities such as picking up a block or building a wall. Subtasks may contain their own (internal) features; external features encode subtask dependencies and conditions over which the robot’s objectives may be defined.

Flexible Models of Robot State. The advantage of such a comprehensive representation is that it accommodates all conceivable features that a robot might use to make its decisions. *State* can be defined to include some or all of the features from the hierarchy. And, although computation arguments might favor traditional state encodings that only employ the highest-level features, there can be benefit to reaching deeper into the hierarchy when reasoning about faults.

Modeling Faulty Actions. The rectangles in Fig. 3 serve as *actions*. Motor control actions reside at the lower levels, called by more abstract actions such as moving forward ($m\bar{f}$). Above, task-level actions comprise controllers designed specifically to produce effects such as aligning with a cube. Intended effects are not always produced, however. We find that, in order to move beyond such faults, it is crucial for the robot to be able to infer that a fault has occurred (e.g., that the robot has not successfully aligned with a cube) and what type of fault occurred (e.g., the robot over-rotated such that it no longer sees the cube). We therefore include such features. We also add task-level actions that the robot can take to correct for faults (such as *find-cube*).

Planning to Build a Wall

Towards a robust wall-building controller, we have adopted a planning methodology that overcomes faults by exploiting our representation. We now highlight the key aspects:

- Given a high frequency of failures experienced with classical planning (Wisner et al. 2012), our problem is more aptly framed as one of FOND conditional planning (Muisse, Belle, and McIlraith 2014), wherein we associate possible fault effects with each task-level action. Through

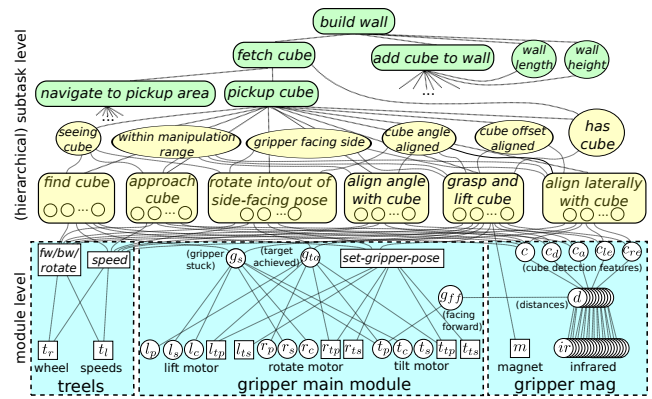


Figure 3: A hierarchical representation for control.

experimentation on the real robot, we have incrementally built up a library of faults, adding a new entry whenever the robot’s plan encounters an unmodeled action effect.

- We additionally record how often faults occur, maintaining probability distributions that can serve as a basis for selecting among plan alternatives. To leverage these, the robot solves a factored MDP (Hoey et al. 1999) derived directly from the FOND planning model and fault library. We mitigate the higher cost of MDP planning by decomposing the overall wall construction into a repeated execution of subplans for *fetch-cube* and *add-cube-to-wall*.
- Repeated subplan executions present the opportunity for intermittent re-planning, as probabilities are updated and as the fault model is successively improved. In effect, over the course of a single scenario, the robot can learn to build more efficiently by embracing the experience of its faults.
- Ongoing work strives to further improve planning models by mining lower-level features that aid in fault prediction.

Acknowledgements

This work was partially supported by projects “ASCENS” (EU 257414) and “H2Swarm” (European Science Foundation, Swiss National Science Foundation grant 134317).

References

- Bonani, M.; Longchamp, V.; Magnenat, S.; Rétornaz, P.; Burnier, D.; Roulet, G.; Bleuler, H.; and Mondada, F. 2010. The marxbot, a miniature mobile robot opening new perspectives for the collective-robotic research. In *Int. Conf. on Intelligent Robots and Systems*.
- Hoey, J.; St-Aubin, R.; Hu, A.; and Boutilier, C. 1999. Spudd: Stochastic planning using decision diagrams. In *UAI’99*, 279–288.
- Muisse, C.; Belle, V.; and McIlraith, S. A. 2014. Computing contingent plans via fully observable non-deterministic planning. In *The 28th AAAI Conference on Artificial Intelligence*, 2322–2329.
- Petersen, K.; Nagpal, R.; and Werfel, J. 2011. Termes: An autonomous robotic system for three-dimensional collective construction. *Proc. Robotics: Science & Systems VII*.
- Wisner, S.; Hitz, G.; Bonani, M.; Gribovskiy, A.; Pradalier, C.; and Magnenat, S. 2012. Autonomous Construction of a Roofed Structure: Synthesizing Planning and Stigmergy on a Mobile Robot. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.