Ensuring Flexibility and Safety through Self-Programming Modular Robots

Matthias Althoff

Technical University of Munich, 85748 Garching, Germany

Abstract

Despite enormous technological advances, investing in a robot is usually only worthwhile if it is used for a longer-term automation task. To address the untapped market of ultra-flexible automation, we have developed robot modules that can be assembled to task-specific modular robots at the customer. Previous attempts to commercialize modular robots failed because they had to be reprogrammed after each assembly. Our novel concept addressed this shortcoming by realizing the first entirely modular, self-programming robot.

1 Introduction

Robots enjoy an increasingly important role in many sectors due to challenges of population aging, labor shortages, high-risk tasks for humans, and large variances in the quality of final products. However, new application areas like construction robotics, agriculture robotics, and healthcare robotics require different robots compared to a standard industrial application as new application areas are usually unique with a different set of problems. While various single-task robots can perform very well, the lack in versatility and flexibility hinders automation in many emerging fields. Hence, we present versatile robotic modules, which can be flexibly assembled on-site into custom configurations [1]. We also present methods to find optimal task-specific robot configurations and to safely perform tasks with formal guarantees. Illustrations of different generations of our modular robots are shown in Fig. 1.

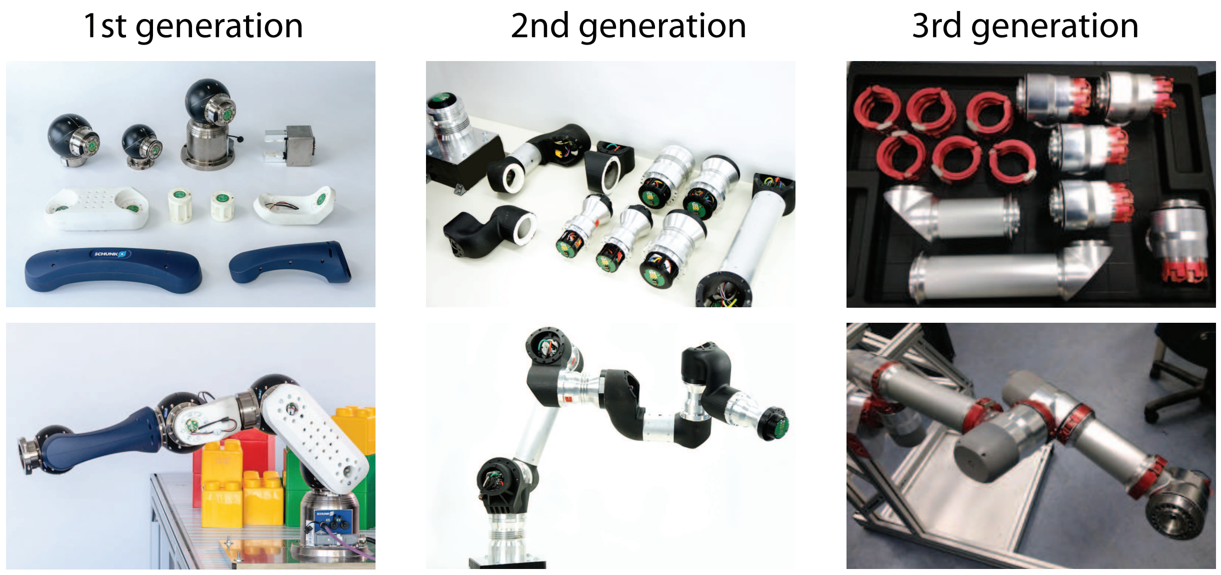

Figure 1: Three generations of modular robots developed at Technical University of Munich.

Most previous works on modular robots used small, identical cubic-shaped or prismatic-shaped modules [10, 11, 14]. The resulting robot assemblies, however, have a significantly reduced performance compared to industrial robots since efficient manipulation requires light-weight link modules and joint modules that have more torque towards the base.

2 Main Idea

To enable the robot to program itself after the modules are assembled, all relevant properties are stored in a module-integrated memory chip. After assembly, the data from the individual modules are read and assembled into a kinematic, dynamic, and geometric model of the robot. This is illustrated in Fig. 2:

(A) Self-programming: 1) Each module contains its characterization. 2) The robot is assembled. 3) The information from each module is collected to automatically build a kinematic, a dynamic, and a geometric model. 4) The robot fulfills a given task without programming by a user.

(B) Self-verification during deployment: 1) An intended trajectory is planned based on the most likely movement of the human. 2) A fail-safe trajectory is created such that the robot never intersects the reachable set of surrounding humans. 3) At the next point in time t_{k+1}, the robot updates its intended trajectory. 4) If the new plan can be verified on time, it is executed; otherwise, the fail-safe trajectory is engaged.

(C) Optimal module composition before deployment: Given 1) modules and 2) a task, we 3) compute possible compositions and 4) select the optimal composition.

3 Optimal Configurations

One of the main challenges of reconfigurable systems is to find possible combinations of modules that are able to complete given tasks while meeting constraints. Most studies on reconfigurable systems focus on obtaining combinations of modules to achieve a given task without considering the required planning algorithms [2–6, 9, 13]. In contrast to previous works, we developed configuration synthesis methods considering constraints and planning algorithms; see, e.g. [7, 8]. This results in a mixed discrete/continuous optimization problem, which is hard to solve. We tackle this problem by finding heuristics that quickly drive the solver to reasonable solutions, since finding the optimal solution is infeasible if one applies this approach to problems of an industrial scale.

In a nutshell, the following steps are taken [8]:

1. Generate compositions with standard robot kinematics. A selection of standard kinematics is shown in Fig. 3 taken out of [8].
2. Fast feasibility checks to rule out:
    - robots that cannot reach the goal,
    - robots that cannot hold the weight,
    - robots that collide with obstacles.
3. Out of the remaining configurations, we perform optimizations and choose the best robot.

Recent work also explores reinforcement learning for the synthesis of modular robots [12].

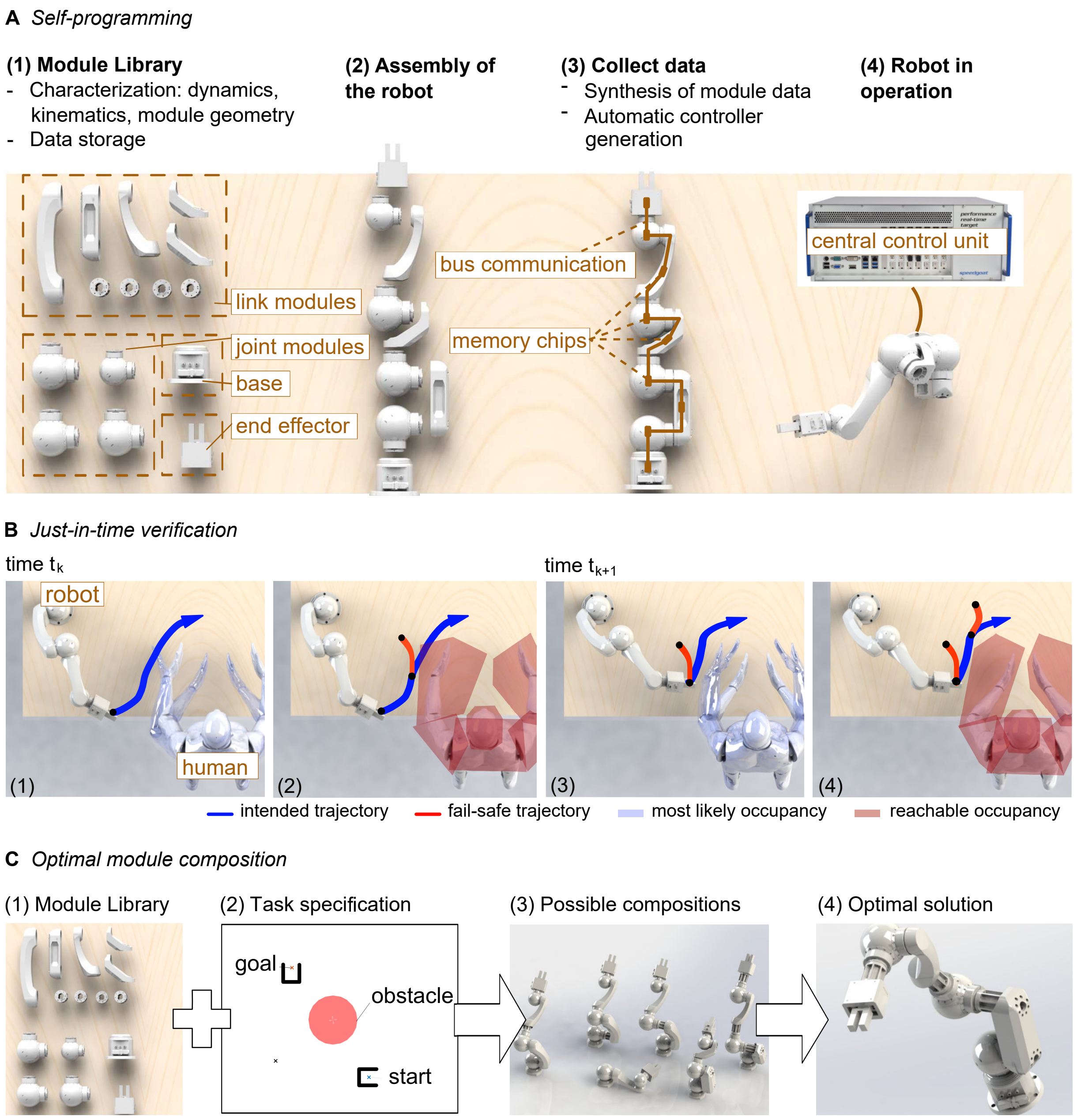

Figure 2: Self-programming, just-in-time verification, and optimal composition of a modular robot (taken out of [1]).

4 Conclusions

Modular robots only realize their full potential when combining self- programming, self-verification, and the ability to find optimal configurations for a given task. When only realizing self-programming, the created robot might not realize its full potential: for example, in many applications, a robot with less than six degrees of freedom is sufficient. However, it is typically puzzling for the human mind to find the kinematics that would still realize the given task with fewer degrees of freedom. Also, since our modular robots can be easily reconfigured, the gained flexibility can only be optimally used when a safety cage is no longer required due to self-verification. The concept of self-programming modular robots is currently realized by the startup RobCo (robco.de) originating from my research group.

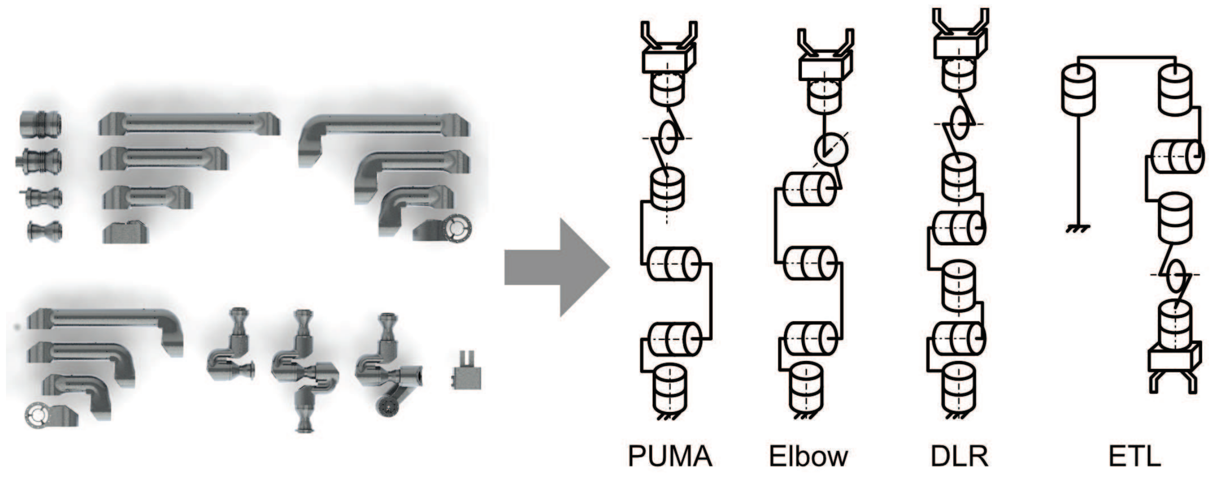

Figure 3: Standard kinematics of robots (taken out of [8]).

5 Acknowledgments

The author gratefully acknowledges financial support by the European Commission project CONCERT under grant number 101016007.

References

[1] M. Althoff, A. Giusti, S. B. Liu, and A. Pereira. Effortless creation of safe robots from modules through self-programming and self-verification. *Science Robotics*, 4(31):1–14, 2019.

[2] A. Brunete, M. Hernando, and E. Gambao. Offline ga-based optimization for heterogeneous modular multi-configurable chained microrobots. *IEEE/ASME Transactions on Mechatronics*, 18(2):578–585, 2013.

[3] I.-M. Chen and J. W. Burdick. Determining task optimal modular robot assembly configurations. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 1, page 132–137, 1995.

[4] O. Chocron and P. Bidaud. Evolutionary algorithms in kinematic design of robotic systems. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robot and Systems*, page 1111–1117, 1997.

[5] W. Chung, J. Han, Y. Youm, and S. Kim. Task based design of modular robot manipulator using efficient genetic algorithm. In *Proc. of International Conference on Robotics and Automation*, volume 1, page 507–512, 1997.

[6] S. Farritor, S. Dubowsky, N. Rutman, and J. Cole. A systems-level modular design approach to field robotics. In *Proc. of IEEE International Conference on Robotics and Automation*, volume 4, page 2890–2895, 1996.

[7] E. Icer, H. A. Hassan, K. El-Ayat, and M. Althoff. Evolutionary cost-optimal composition synthesis of modular robots considering a given task. In *Proc. of the IEEE International Conference on Robotics and Automation*, page 3562–3568, 2017.

[8] S. B. Liu and M. Althoff. Optimizing performance in automation through modular robots. In *Proc. of the International Conference on Robotics and Automation*, page 4044–4050, 2020.

[9] C. J. J. Paredis and P. K. Khosla. Synthesis methodology for task based reconfiguration of modular manipulator systems. In *Proc. of the 6th International Symposium on Robotics Researc*, 1993.

[10] J. Seo, J. Paik, and M. Yim. Modular reconfigurable robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):63–88, 2019.

[11] K. Stoy, D. Brandt, and D. J. Christensen. *Self-Reconfigurable Robots: An Introduction.* 2010.

[12] J. Whitman, R. Bhirangi, M. Travers, and H. Choset. Modular robot design synthesis with deep reinforcement learning. In *Proc. of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, page 10418–10425, 2020.

[13] G. Yang and I.-M. Chen. Task-based optimization of modular robot configurations: minimized degree-of-freedom approach. *Mechanism and Machine Theory*, 35(4):517–540, 2000.

[14] M. Yim, W.-m. Shen, B. Salemi, et al. Modular self-reconfigurable robot systems. *IEEE Robotics & Automation Magazine*, 14(1):43–52, 2007.