Development of a Benchmarking Framework for Evaluation of Modular Robots

Background

Modular industrial robots have great potentials to optimize industrial automation. They have various advantages compared to their non-modular counterparts: Modular concepts as presented in [1] make it possible to flexibly adapt the physical structure of a robot and, thus, can be optimally configured to meet the requirements of a task, can easily be specialized to automate tasks no standard kinematic can solve, and can be adapted to new upcoming tasks. In order to automatically evaluate the advantages of modular robots, to compare them against other industrial robots and to evaluate different sets of robot modules, a benchmarking framework is a valuable tool.

This thesis is offered in cooperation with the startup FLEXMATE.

Description

The aim of this thesis is the development of an automated benchmarking framework for evaluating industrial robots against multiple use cases. It contains several components:

1. Three databases, one for use cases, one for industrial robots, and one for robot modules;
2. a web-based front end;
3. interfaces which allow for several optimization functions for robot path planning and for the robot composition to be tested;
4. a benchmarking suite; and
5. a module for summarizing, visualizing and saving the results.

The framework to be developed must support four application scenarios: (a) different industrial robots and (manually) given modular robot compositions are benchmarked against each other, (b) different optimization algorithms for path planning and robot composition optimization are benchmarked, (c) an optimal composition of a modular robot for a certain use case is calculated, (d) a set of robot modules is evaluated against multiple use cases and a common score is given. The user must be able to select the robots, robot modules and the use cases which should be taken into account, run the benchmarking and receive the results. Figure 1 shows a pool of robot modules and one possible robot composition which is assembled from them. Figure 2 shows a visualization of the main application of the benchmarking framework. Since the framework is to be modular, all the components from figure 2 can be evaluated separately as well.

Figure 1: left: one pool of robot modules, which can be assembled to various robot compositions; right: one possible robot composition out of the set on the left.

Technische Universität München

Fakultät für Informatik

Lehrstuhl für Echtzeitsysteme und Robotik

Supervisor:
Prof. Dr.-Ing. Matthias Althoff

Advisor:
Paul Maroldt, M.Sc.
Constantin Dresel, M.Sc.

Research project:
Modular Robotics

Type:
Bachelor's Thesis/Master's Thesis

Research area:
Robotics, Benchmarking, Optimization

Programming language:
MATLAB, C++

Required skills:
Programming, Robotics

Language:
English

Date of submission:
21. Januar 2020

For more information please contact us:

Phone: +49.89.289.18144

E-Mail: paul.maroldt@tum.de

Internet: www6.in.tum.de

Technische Universität München

Fakultät für Informatik

Lehrstuhl für Echtzeitsysteme und Robotik

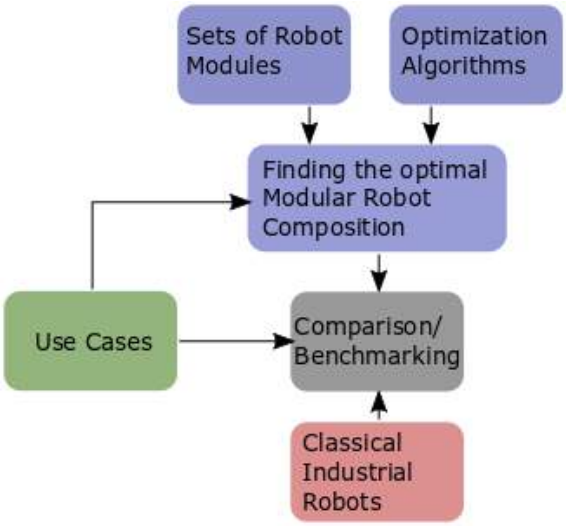

Figure 2: Visualization of the frameworks applications: An optimal robot composition is found based on the available set of robot modules and a specific use case using optimization algorithms for path planning and calculating robot compositions. This composition is then compared to classical industrial robots in the context of the use case.

For this thesis, existing optimization algorithms developed at the chair must be integrated. The whole framework must be tested with various use cases including scenarios for concrete renovation of a pole and a wall. The calculation and the visualization should be implemented using MATLAB or C++. The existing algorithms are implemented in MATLAB or C++.

Tasks

- Familiarizing with modular robotics, robot benchmarking and the existing code.
- Research state of the art methods for benchmarking.
- Definition of the database structure.
- Definition of the framework structure and the necessary interfaces to other program parts (i.e., optimizers, database, etc.).
- Implementation of the framework and integration of all existing components.
- Building a simple user interface for using the framework.
- Showing functionality by running the use cases of concrete renovation of a pole and a wall.
- Evaluating several robot module pools and sub-pools.
- Documentation of results.

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):eaaw1924, 2019.