

A Software Structure for Control and Monitoring of Flexible Automation

Industrial robots and manipulators are still mostly used for repetitive tasks in never-changing and over-engineered environments despite robots being generic machines. Most of our knowledge about automation and our approach to automation systems are born out of the car industry and large-scale manufacturing. Such systems are designed for highly structured environments and target many years of operation. The systems traditionally favor precision, repeatability and cycle times but lack flexibility, sensing, and easy programming. Today, robots are slowly spreading beyond their roots in industrial manufacturing and are making an entry into other fields such as health-care, education, and homes. These robots are in many ways based on a rather different methodology, since they are designed to work in non-structured environments and with little repetition. Working with robots for small-batch productions and low-volume manufacturing put us somewhere in-between the highly- and non-structured environments. It allows us to benefit from some structure and repeatability known from large-scale manufacturing, but we still need to learn how to handle a larger degree of unknowns and uncertainties to make the system feasible.

This thesis deals with the use of robots in industrial assembly and manufacturing tasks. The ultimate goal is to make robotics viable for small-batch productions such as those often found in small- and medium-sized enterprises. In many ways, we are predisposed to expectations on what constitute assembly systems. Expectations formed by large-volume and high-production assembly systems. This thesis challenge the preconceptions that automation is required to run flawlessly and without errors. We investigate a reversible programming approach and design concepts which help increase robustness and reliability of robotic setups and we present a reversible programming language for robotic assembly sequences. Programs written in this language can be executed both forwards and backward. Forward execution results in assembly, whereas backward execution leads to disassembly. This allows for a trial-and-error approach to automatically correct faults by backtracking in the program and repeating the assembly. Thereby effectively allowing many errors in robotic assembly to corrected automatically. The language supports both reversible and nonreversible features. Some task such as inserting a screw can both be done and undone, other task such as painting can be redone but never undone and finally some task such as inserting and removing a nail are conceptually inverse but requires different forward and backward instructions. The programming language supports such different kinds of reversibility and the thesis discusses reversibility in manufacturing tasks and its applicability for industry.

This thesis mainly worked with the notion that uncertainties and variations should be addressed through software to increase the feasibility of automation significantly. Programming a robot to run an assembly sequence once is fairly easy whereas programming a robot to run continuously is difficult. Apart from reversibility the thesis investigates the possibilities of automatic error detection in robotic assembly. Based on identifying irregularities in the robot-motor-currents errors and collisions are detected such that the robot can be stopped in a timely manner and the error can be corrected. Lastly the thesis presents a concept fast configuration and set-up of new assembly tasks where a search for robust and uncertainty-tolerant robot motions is conducted using simulation and an explicit model of uncertainties.