Towards Multi-functional Robot-based Automation Systems

Andreas Angerer[1], Michael Vistein[1], Alwin Hoffmann[1], Wolfgang Reif[1], Florian Krebs[2] and Manfred Schnheits[2]

[1]*Institute for Software & Systems Engineering (ISSE), University of Augsburg, Augsburg, Germany*

[2]*Center for Lightweight Production Technology, Institute of Structures and Design, German Aerospace Center, Stuttgart, Germany*

Keywords: Cooperating Robots, Industrial Automation and Robotics, Robot Programming, Modelling and Simulation.

Abstract: Multi-functional robot cells will play an important role in smart factories of the future. Equipped with flexible toolings, teams of robots will be able to realize manufacturing processes with growing complexity. However, to efficiently support small batch sizes and a multitude of process variants, powerful software tools are required. This paper illustrates the challenges that developers face in multi-functional robot cells, using the example of CFRP production. The vision of a new programming environment for such future flexible automation systems is sketched.

1 INTRODUCTION

According to the International Federation of Robotics (2014), the automotive industry is currently the largest operator of industrial robots. This is mainly due to the large batch sizes and, thus, a high number of repetitive tasks which allows for a more economic integration of robots compared to other industries. However, there is a trend to apply robotic systems also for small batch production and for complex manufacturing processes (cf. euRobotics aisbl, 2014). Especially with the *Internet of Things and Services* and strategic initiatives such as *Industry 4.0* in Germany (cf. Kagermann et al., 2013), the idea of smart factories with intelligent machinery – referred to as cyber-physical systems (cf. Geisberger and Broy, 2012) – and a highly customized production is emerging. The products incorporate the knowledge of how they need to be processed and, moreover, they communicate directly with the machinery. As cyber-physical systems, they know their skills and offer them as services in a smart factory.

From our point of view, multi-functional robot cells will play an important part in future smart factories. Industrial robots are flexible machines that can perform a broad variety of tasks using different end-effectors. Moreover, dynamic teams of cooperating robots can together handle complex tasks if required. Until today, there is a strong focus on knowledge-based manufacturing systems or cyber-physical systems consisting of a single robot. However, when regarding dynamic teams of robots, the available skills depend on the composition of the team. Cyber-physical systems must be able to occasionally form a new system with an extended set of skills and services which poses new challenges for research.

The production of *carbon fiber-reinforced plastics* (CFRP) is an interesting example where multi-functional robot cells are important. CFRPs are becoming more and more important for many modern products, e.g. for airplanes or helicopters, but also for the automotive industry. Today, the production is often done manually, which is a very tedious and strenuous task, therefore it is worthwhile to automate this process. However, the dimensions of CFRP parts can vary tremendously depending on the product: Some aircraft structures can be only a few centimeters in size whereas others can be in the range of several dozens of meters. Thus, a single industrial robot in a static, fixed configuration will not always be sufficient for handling the parts.

This paper introduces the challenges as well as a possible road map for modeling, programming and simulating dynamic robot teams in multi-functional cells. Section 2 delivers a detailed view of both the production process for CFRP and the multi-functional robot cell built at the Center for Lightweight Production Technology (ZLP) of the German Aerospace Center (Deutsches Zentrum für Luft- und Raumfahrt, DLR) in Augsburg. In Section 3, the current ap-

Angerer A., Vistein M., Hoffmann A., Reif W., Krebs F. and Schönheits M..
Towards Multi-functional Robot-based Automation Systems.
DOI: 10.5220/0005573804380443
In *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2015), pages 438-443
ISBN: 978-989-758-123-6

Table 1: Cutpiece distribution for the demonstration process.

Cutpiece class	**#**
Skin "tile" (regular shape)	40
Skin "tile" (irregular on margins)	16
Frame thickness extension (regular shape)	40
Frame thickness extension (irregular)	16
Stringer base (large)	48
Stringer base (small)	48
Sum	**208**

proaches are described and challenges for application development in such large scale environments are identified. Novel approaches for efficient software development for large teams of robots and peripheral devices are introduced in Section 4. The paper is concluded with Section 5.

2 CASE STUDY

To motivate the challenges that have to be faced in multi-functional robot cells, we would like to address the automated production of CFRP structural components for aerospace applications. Many processes for the production of CFRPs exist such as advanced fiber placement, resin transfer molding, or vacuum assisted resin infusion (VARI). An overview over these techniques can be found in Baker et al. (2004). In the following, the VARI process will be shortly introduced to understand the process requirements. A first automated solution is described afterwards.

2.1 Vacuum Assisted Resin Infusion

The VARI process basically uses dry textile cutouts laid into a mold at specific positions. Afterwards the mold is sealed with an airtight foil and other auxiliary materials. The whole layup is then evacuated and subsequently infused with resin. To fully leverage the weight saving potential of CFRPs, fibers are mainly oriented along the major axis of tensile loads. In other words, more material is placed in areas where high loads are expected and less in other areas. Ultimately, this leads to a very complex design of cutouts – often referred to as cutpieces – in which many cutpieces are unique within the layup. The whole layup is documented in a plybook[1] which specifies the shape of each cutpiece and where it should be placed in the layup. In aerospace applications, these plybooks usually contain hundreds of individual cutpieces.

[1]A ply is a set of cutpieces forming a layer in a layup.

Figure 1: The experimental setup for the layup in the multi-functional cell.

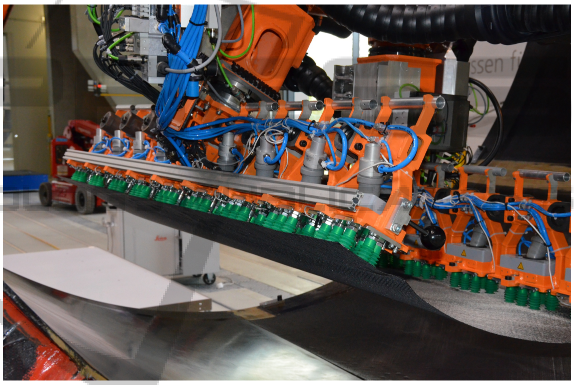

Figure 2: Both robots gripping a cutpiece.

2.2 Experimental Setup

In 2014, the DLR in Augsburg implemented an automated demonstration for this process on site, using a section of an aircraft fuselage. This component is basically a half-cylinder with a radius of ~1.8 meters and roughly a length of 2.5 meters. To illustrate the automation process, a simplified plybook was created consisting of 208 mostly uniform cutpieces. Table 1 summarizes the different ply classes. After plybook design, a rough layout of the process in the robot cell was drafted. The DLR's multi-functional cell (Krebs et al., 2014) was chosen as a platform.

This cell is intended for the evaluation of automated production processes for CFRP parts and consists of two industrial 6-DOF arms on a common linear track in overhead configuration, and three XYZ-portals spanning a workspace of roughly 30 x 15 x 7 meters. For this application, only the two central industrial arms were used (cf. Figure 1). The validation of cooperative layup of single cutpieces was one of the main objectives of this setup. Therefore the larger cutpieces were handled using the KUKA.RoboTeam technology which allows multiple robots to act cooperatively. Three modes of operation are possible:

- Synchronized start: all robots start a process step at the same time
- Synchronized motion: both the start and the finish times of a step occur simultaneously for all robots
- Motion cooperation: one robot acts as a master while the other robots follow in a geometrically linked fashion

On each end-effector, a gripper was mounted that was able to hold an edge of the fabric (cf. Figure 2). One gripper was additionally equipped with a system to measure the layup quality using a laser scanner.

2.3 Process Description

In a nutshell, the process is a pick&place application that requires the cutpieces to be picked up from a table and positioned into the mold at the proper coordinates for each cutpiece. At the beginning, both robots pick up each side of the cutpiece with their vacuum grippers from a table together using a synchronized motion. The following motions transfer the robots into safe a position over the mold and include a motion of the linear track. Although this motion needs to be synchronized as well, this cannot be carried out by KUKA.RoboTeam since linear tracks are not fully supported. A workaround was found by starting both robot motions at the same time and letting them travel the for same distance. After reaching the safe position, the robots are geometrically linked and the master robot leads the slave to a position just before layup. Then, breaking the link again, each robot moves in a synchronized motion to its individual target position, making final adjustments along the way to compensate for small deformations in the mold.

When both robots are in place, heaters are extended to melt the thermoplastic binder and temporarily bond the cutpiece. After a ply has been laid up completely, the robots move back into the safe position. The robot without the laser scanner moves out of the mold. After that, the measurement is started by the second robot to verify that all cutpieces have been positioned correctly according to the plybook. Assuming the measurement showed no anomalies, both robots move back into the starting configuration and repeat the cycle until all plies have been laid.

2.4 Programming the Process

Currently available offline programming toolchains like DELMIA (Dassault) and Process Simulate (Siemens) lack proper support for cooperating robots and could thus not be used directly for programming the process. Instead, a hybrid approach was employed: The transfer from the table to the mold was done using a classical teach-in process to define fixed motions for both of the robots. Picking up the cutpieces as well as layup onto the mold was done by using parameterized robot programs. The parameters – mainly the gripping and target points – were generated in CPD[2] and exported manually by the programmers. Controlling the cooperation of the robots had to be done by manually inserting appropriate commands into the robot programs.

In sum, a lot of steps had to be done manually due to the lack of appropriate support in CAD and offline programming tools. Additional difficulties had to be solved in dealing with KUKA.RoboTeam and online programming as mentioned by Larsen et al. (2014). While this experiment – dealing only with a minor number of variations – already formed a challenge, it is obvious that this approach will not scale to a real world example with a multiple of individual plies.

3 CHALLENGES

Multi-functional cells like the DLR MFZ are able to handle complex, fast-changing processes with rather small lot sizes. The key ingredient are different types of manipulators with the ability to cooperate, varying end-effectors as well as additional tools and fittings. But still planning, programming and controlling processes in such robot cells remains a difficult issue. The problem can be divided into two major challenges, as outlined below.

3.1 Modeling, Planning and Simulation

Processes with a very high complexity and variability cannot be fully planned and programmed manually. Developers of the system software need tools for assisted, semi- or fully automated planning of all tasks involved in a process. Moreover, mapping tasks to a multi-functional robot cell and determining an appropriate combination of robot arms, end-effectors and sensors to reliably execute a certain task is a challenge that nowadays requires skilled and experienced human experts.
In sum, we see challenges in the following areas:

- modeling complex processes with a large number of interdependent tasks,
- modeling devices in a multi-functional cell and their abilities (e.g. manipulators, end-effectors,

[2] Composites Part Design, an extension for Dassault's CAD software CATIA.

actuators, sensors, . . .),

- modeling interaction of the devices (e.g. team and un-team multiple manipulators on the fly),
- representing capabilities of single manipulator-endeffector combinations,
- representing capabilities of teams of manipulators and end-effectors
- (semi-)automated planning of process execution in multi-functional robot cells,
- simulation and qualitative analysis of complex processes in multi-functional cells.

For defining robot paths based on CAD data, there exist various toolchains that are used in industry as well as academia today. A prominent example is the combination of Dassault Systeme's CATIA and DELMIA, sometimes combined with cenit's FASTSURF. DELMIA can be used to perform virtual teach-in of robot motions based on CAD input from CATIA. FASTSURF adds support for motions along more complex surfaces and furthermore integrates simulation and analysis functionality geared towards concrete processes like painting. However, those tools are rather focused on modeling single tasks and reach their limits when it comes to modeling complex processes with hundreds of sub-tasks like in the VARI process. Furthermore, the support for robot teams including different cooperation patterns is rather limited. In research, various single aspects that are relevant for multi-robot systems have been treated, like collision detection and path planning for multi-robot systems (cf. Mediavilla et al., 1998; Larsen et al., 2014). Only recently, research started to address the idea as well as basic challenges regarding off-line programming environments for multi-robot systems (cf. Basile et al., 2012; Gan et al., 2013).

On the other hand, a lot of research has been done in the area of process description as such as well as (automatic) decomposition of processes into tasks (cf. Thomas and Wahl, 2001; Ou and Xu, 2013; Huckaby et al., 2013). This has also been applied to the robotics domain and has been combined with modeling of devices, end-effectors and their skills (cf. Pfrommer et al., 2013; Stenmark and Malec, 2013; Michniewicz and Reinhart, 2014). However, research in these areas has been focused largely on single robot systems, and the additional challenges induced by cooperating robot teams have not been treated in-depth.

3.2 Deployment to Real-world Systems

Similar to the physical actuators and end-effectors in multi-functional cells, also the controller structure is very heterogeneous. Robot controllers mostly are programmed using proprietary, manufacturer-dependent programming languages and add-ons (e.g. for robot cooperation). Devices utilize various technologies like field bus systems (e.g. Profinet, DeviceNet, Ethercat) and ethernet-based protocols to communicate. Often PLCs are used to build topologies and also to implement superordinated logic, oftentimes real-time deterministic and safety-critical logic. But the PLCs typically are programmed using proprietary tools and languages in turn.

In classical offline-programming scenarios, code generation is often used to derive executable artifacts from models for the target platforms (cf. Feldmann et al., 2013; Stenmark et al., 2014). Even in a simple robot cell, this has lots of disadvantages:

- many manual steps, e.g. exported robot programs typically have to be moved to the robot controller
- shortcomings in flexibility, e.g. communication to other devices is often done through generated I/O commands in the robot program, this mostly lacks good support for changes e.g. in I/O mapping
- ensuring overall consistency is imposed on the user

In a multifunctional, reconfigurable robot cell, this all gets even more tedious and error-prone to support all of the different target platforms present.

4 TOWARDS EFFICIENT SOFTWARE DEVELOPMENT

In order to efficiently develop software for multi-functional robot cells, progress is needed on various levels. This section presents areas that need be pushed forward from our point of view (cf. Figure 3).

First of all, it is necessary to **model the manufacturing process** and break it down into single *tasks*. It might be necessary to *decompose tasks* into further sub-tasks. *Dependencies* among tasks need to be modeled in order to determine a feasible order for execution and parallelization. For example, in the VARI process, the cutpieces may be produced all in parallel, while the layup process obviously requires a certain order of tasks. Each of the tasks has certain *requirements* that have to be fulfilled for successful execution. For example, cutpieces may be bent only to a certain degree in order not to damage the fibers, which has to be ensured during handling. Furthermore, a maximum position tolerance has to be respected when the cutpieces are laid into a mold. While some of those requirements are mandatory for successful production, others can be met with a certain tolerance, which influences the result quality. To judge the resulting quality, the process model has to introduce *metrics for quality*, which allows to predict quality of the final result. For more accurate predictions, empirical values (e.g. known inaccuracies) should be usable for extrapolation.

In order to map a manufacturing process to the capabilities of a multi-functional robot cell, a thorough **model of robot teams and tools** is required. This model must be able to describe not only the *characteristics and skills* of a single robot with a specific end-effector, but also that of different combinations of devices, i.e. different combinations of robot arms and end-effectors and in particular the *combination of multiple robot arms to teams* as required by the task. When multiple robot arms form a team, there are different *cooperation patterns* that fit various kinds of tasks (e.g., uniform motion or motion relative to other team members' motions). The type of cooperation pattern has an influence on how the team needs to be represented by the model. In robot teams, different *constraints* apply to the allowed operations, for example due to the workspace of the single members or due to physical capabilities like maximum velocities and accelerations. On the other hand, those constraints can depend on the initial configurations of the members in the team. Thus, the *formation of teams* plays an important role: Certain preconditions have to be met in order to create a team from single robots.

Based on the manufacturing process model, a **CAD model** of the robot cell and the model of robot teams and tools, **a unified programming environment for multi-functional robot cells** becomes feasible as a basis for mapping domain-specific pro-

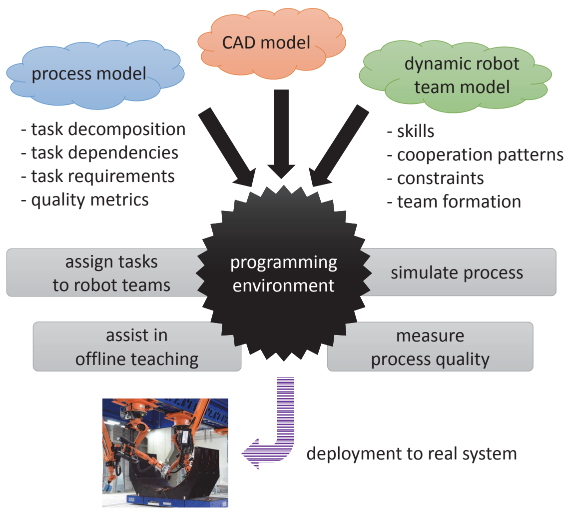

Figure 3: Process, CAD and robot team models should be unified in a programming environment to assist software development for multi-functional robot cells.

duction processes to concrete manufacturing cells. By analyzing task composition and dependencies and matching task requirements with robot (team) skills, the programming environment can assist users in *assigning tasks to robot teams*. To design concrete operations of a robot (team) required for a certain manufacturing task, the programming environment must *assist users in offline teaching*. Conventional on-line teaching techniques are time-consuming and expensive already for systems with single robots. Due to constraints and physical dimensions of robot team setups, on-line teaching becomes infeasible.

The assignment of tasks to robot teams and the concrete task realization by robot team operations has a strong influence on the quality of the process. A programming environment should allow for *measuring the resulting process quality* based on the quality metrics inherent to the process model and the characteristics of the robots and end-effectors. In this context, the ability to *simulate a concrete process in a multi-functional cell* is of great importance. Simulation should be accurate enough to roughly conclude about the process quality and to compare it among different variants of a process. Simulation should furthermore support user-defined parameters for interrupting and replaying, similar to the use of breakpoints in software debugging.

A crucial factor for the effectiveness of the proposed offline programming system is the transition from simulation to the real-world system. As mentioned before, code generation from models is the predominant approach nowadays. In multi-functional robot cells with complex end-effectors and very heterogeneous controllers, the effort for developing code generators increases dramatically. Not only a variety of target platforms (robot controllers, tool controllers, PLCs, ...) has to be supported, but also the semantic consistency of the resulting code has to be ensured. In order to match the required hard real-time requirements in a cooperating robot team, further aspects like communication latencies have to be considered. Furthermore, modifications to the generated code are often hard to re-integrate into the models, requiring those modifications to be made over and over again after code generation. From our point of view, this "classical" code generation approach is no longer feasible for multi-functional robot cells. Instead, a shift towards open, standardized and configurable interfaces of the different devices is necessary. Those interfaces should reflect more closely the semantic concepts of the process model. Thus, it might become feasible to directly execute the models instead of generating code, which would allow for a seamless transition from simulation to real-world operation.

First steps towards the proposed programming environment have already been made. In a first setup, a single robot and multiple end-effectors have been used. Based on a process model and a CAD model of the robot cell, a semi-automatic generation of a feasible process flow is possible, including offline teaching and simulation. Details on the architecture and setup can be found in Nägele et al. (2015).

5 CONCLUSION

This work illustrated the challenges that developers face when complex manufacturing processes should be realized with a multi-functional robot cell. The Multi-Functional Cell located at the DLR ZLP in Augsburg is one of the world's biggest multi-functional robot cells in operation today and can be seen as a prototype of what is yet to come according to current research agendas. Based on a particular CFRP manufacturing process, the main challenges that need to be solved were explained. The vision of a programming environment for multi-functional robot cells that unifies process, CAD and robot team models was presented. By future joint research, the University of Augsburg and the DLR Center for Lightweight Production will strive to realize this vision.

ACKNOWLEDGEMENT

The demonstration depicted in Section 2 was made possible with the help and support of Andreas Buchheim, Somen Dutta, Mona Eckardt, Lars Larsen, Thomas Schmidt, and Alfons Schuster of the DLR.

REFERENCES

Baker, A. A., Dutton, S., and Kelly, D. (2004). *Composite Materials for Aircraft Structures*. American Institute of Aeronautics & Astronautics.

Basile, F., Caccavale, F., Chiacchio, P., Coppola, J., and Curatella, C. (2012). Task-oriented motion planning for multi-arm robotic systems. *Robotics and Computer-Integrated Manufacturing*, 28(5):569 – 582.

euRobotics aisbl (2014). Strategic research agenda for robotics in Europe 2014 – 2020.

Feldmann, S., Loskyll, M., Rosch, S., Schlick, J., Zuhlke, D., and Vogel-Heuser, B. (2013). Increasing agility in engineering and runtime of automated manufacturing systems. In *2013 IEEE Intl. Conf. on Industrial Technology*, pages 1303–1308.

Gan, Y., Dai, X., and Li, D. (2013). Off-Line Programming Techniques for Multirobot Cooperation System. In *Intl. J. of Adv. Robotic Systems*. InTech Europe.

Geisberger, E. and Broy, M., editors (2012). *Integrierte Forschungsagenda Cyber-Physical Systems*. acatech.

Huckaby, J., Vassos, S., and Christensen, H. I. (2013). Planning with a task modeling framework in manufacturing robotics. In *2013 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 5787–5794.

International Federation of Robotics (2014). *World Robotics – Industrial Robots 2014*. VDMA Verlag.

Kagermann, H., Wahlster, W., and Helbig, J. (2013). Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Final report of the Industrie 4.0 Working Group, acatech.

Krebs, F., Larsen, L., Braun, G., and Dudenhausen, W. (2014). Design of a multifunctional cell for aerospace CFRP production. *The Intl. J. of Adv. Manufacturing Technology*, pages 1–8.

Larsen, L., Kim, J., and Kupke, M. (2014). Intelligent path planning towards collision-free cooperating industrial robots. In *11th Intl. Conf. on Informatics in Control, Automation and Robotics*.

Mediavilla, M., Fraile, J., Peran, J., and Dodds, G. (1998). Optimization of collision free trajectories in multi-robot systems. In *Robotics and Automation. Proceedings. 1998 IEEE Intl. Conf. on*, volume 4, pages 2910–2915 vol.4.

Michniewicz, J. and Reinhart, G. (2014). Cyber-physical robotics – automated analysis, programming and configuration of robot cells based on cyber-physical-systems. *Procedia Technology*, 15:567–576.

Nägele, L., Macho, M., Angerer, A., Hoffmann, A., Vistein, M., Schönheits, M., and Reif, W. (2015). A backward-oriented approach for offline programming of complex manufacturing tasks. In *The 6th Intl. Conf. on Automation, Robotics and Applications*.

Ou, L.-M. and Xu, X. (2013). Relationship matrix based automatic assembly sequence generation from a CAD model. *Computer-Aided Design*, 45(7):1053 – 1067.

Pfrommer, J., Schleipen, M., and Beyerer, J. (2013). PPRS: Production skills and their relation to product, process, and resource. In *2013 IEEE Intl. Conf. on Emerging Tech. & Factory Automation*, pages 1–4.

Stenmark, M. and Malec, J. (2013). Knowledge-based industrial robotics. In *12th Scandinavian Conf. on Artificial Intelligence*, pages 265–274.

Stenmark, M., Malec, J., and Stolt, A. (2014). From high-level task descriptions to executable robot code. In *7th IEEE Conf. on Intelligent Systems*, pages 189–202.

Thomas, U. and Wahl, F. (2001). A system for automatic planning, evaluation and execution of assembly sequences for industrial robots. In *Intelligent Robots and Systems. Proceedings. 2001 IEEE/RSJ Intl. Conf. on*, volume 3, pages 1458–1464 vol.3.