Planning and Execution of Collision-free Multi-robot Trajectories in Industrial Applications

Andreas Angerer[1], Alwin Hoffmann[1], Lars Larsen[2], Michael Vistein[1,2], Jonghwa Kim[1], Michael Kupke[2] and Wolfgang Reif[1],
[1] Department of Computer Science, University of Augsburg, Germany
[2] German Aerospace Center (DLR), Center for Lightweight Production Technology, Augsburg
Email: `angerer@isse.de`, Phone: +49 821 598-2171, Fax: +49 821 598-2175

Abstract

Production of carbon fiber reinforced plastics is nowadays mainly done by hand, as process automation is very difficult. This paper presents first steps towards robot-based automation of such complex processes: a simulation environment performing collision free motion planning for cooperating robots is combined with an object-oriented, real-time capable control framework for industrial robots. First qualitative results from an industrial setting are presented.

1 Introduction

In order to reduce fuel consumption and CO_2 emissions, there is a trend in aircraft industries to reduce the weight of airplanes. Considering the life cycle of an airplane which is about 20 years with around 50 000 flying hours, it becomes clear that every kilogram less results in significantly less fuel consumption and CO_2 emissions. A possibility to reduce the weight of an airplane is to use carbon fiber reinforced plastics (CFRP) which have a high potential for lightweight and stress-optimized construction. The structure of modern airplanes like the Airbus A380 (ca. 28 %), the Airbus A350 XWB (ca. 53 %) and the Boeing 787 (ca. 50 %) already consists of CFRP parts. Hence, CFRP have already become an integral part of aircraft industries.

However, the production process of CFRP parts for airplanes still consists mainly of manually performed steps. **Figure 1** shows the widespread *vacuum assisted resin infusion* (VARI) technique which uses non-resinous dry carbon fiber textiles [1]. Here, a CFRP part is manufactured from a multitude of fragile textile cutouts with different shapes and sizes. Each cutpiece has to be laid into an assembly mold which is a laser-guided, but still manual positioning and draping step. To fix a cutpiece, a thermoplastic binder material is melted by bringing in heat. Before the resin is injected, the layers of cutpieces are covered with a vacuum build-up. Finally, the part is transported to an oven for curing.

This manufacturing process in combination with the small-batch sizes in aircraft industries raises a number of challenges for its automation. The fragile textiles and the different sizes – from a few centimeters up to several meters – lead to complex material handling. Accordingly, several robot-based solutions with sophisticated end-effectors have been suggested recently (e. g. [2, 3, 4]). Usually, every solution is developed for a particular component shape and size. As the transportation and draping of huge carbon fiber textiles is comparable to putting a large tablecloth on a table with two persons, a solution with cooperating robots and vacuum-based end-effectors was developed at the German Aerospace Center (DLR) [5].

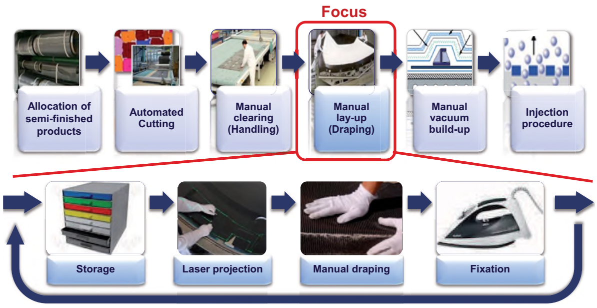


Figure 1 Process chain for manually manufacturing CFRP parts from dry carbon fiber textiles [2].

However, the high number of different cutpieces in combination with the low number of produced CFRP parts in aircraft industries demands for multi-functional robot cells and very flexible automation software [6]. To manufacture parts with a length up to 30 m a *Multifunctional Robotic Cell* with five KUKA industrial manipulators on linear tracks was developed at the DLR Center for Lightweight Production Technology in Augsburg [7]. For evaluation purposes, a smaller *Technology Evaluation Cell* with only two KUKA manipulators on a shared linear track was implemented, too.

Furthermore, the common teach-in process in industrial robotics is not feasible anymore. Due to the variety of different composite parts and cutpieces, it becomes necessary to completely plan and simulate the manufacturing process offline using a virtual model of the appropriate multi-functional cell. We have already shown in a proof of concept that this approach is feasible for single robot solutions with different material handling end-effectors [8]. In this work we demonstrate that a CFRP manufacturing process involving cooperating robots (1) can be planned and simulated offline and (2) can be deployed without changes to the real cell for execution. These two steps are essential requirements for multi-functional cells in small-batch size industries such as aircraft manufacturing [6].

We present a vertical prototype of a system which allows the fully automated production of an aircraft fuselage. The system contains a collision-free path planner using a simulation environment and the execution of cooperating robot motions using a flexible robot programming framework. The challenges for multi-functional cells are introduced briefly in **Section 2**. In **Section 3**, the planning of collision-free multi-robot trajectories is described, whereas their execution is part of **Section 4**. An overview of experimental results and an accuracy evaluation is delivered in **Section 5**. The paper is concluded with **Section 6**.

2 Challenges

The automated production of CFRP components raises a lot of challenges on various levels. To cope with the large variety of composite parts and cutpieces as well as additional quality assurance steps, the automation systems have to offer various functionality. The challenges in such multi-functional systems have been analyzed in [6]. This work presents distinct contributions to solve some of those challenges. In the following, we will state those contributions and elaborate on how they simplify the automation of CFRP manufacturing.

As mentioned above, a CFRP part is manufactured using the VARI technique from a multitude of textile cutpieces with a large variety in shape and size. Some cutouts may be laid in the final mould in parallel, while for others a fixed order has to be respected. The resulting manufacturing process therefore consists of a large number of interdependent steps, some of which may require multiple robots and end-effectors (e.g. due to the physical size of the cutouts). Modeling, planning and simulating such complex processes is a challenging task that is not adequately supported by today's tools [6]. While there exists a variety of so-called offline programming tools like DELMIA (Dassault Systèmes), Process Simulate (Siemens), KUKA.Sim Pro (KUKA) or RobotStudio (ABB), the planning of manipulation tasks for teams of robots in particular is seldom supported (cf. [9, 10]).

The long-term goal of this work is to develop a programming environment that supports modeling such complex manufacturing processes that involve multiple robots and end-effectors. An important contribution of this paper is to evaluate that – in combination with CFRP manufacturing data – a previously presented multi-robot path planner [11] can be used to completely model and plan a particular fuselage manufacturing scenario. This is an important step to show the general feasibility of such an approach. Hence, such path planners will be an integral part of the envisioned new programming environment.

A second challenge is the deployment of pre-planned and simulated process models to real-world systems. The classical approach in this case is code generation, which is supported by various tools for different target platforms (e.g. KRL generated by KUKA.Sim Pro, or RAPID generated by ABB's RobotStudio). However, this approach is hardly feasible in the context of automated production of CFRP components. As mentioned before, certain process steps require teams of robots for execution. To the authors' knowledge, code generation for synchronized robot teams is not supported by offline programming tools today. Furthermore, code has to be generated not only for robot motions, but also for end-effector control. For the processes involved in CFRP production, complex end-effectors often require stand-alone controllers that require additional code to be generated, which increases the effort for developing code generators.

Finally, the overall CFRP manufacturing process tends to get quite complex, and some tasks should be parallelized for efficient production. Thus, the code controlling process execution is usually run on separate systems (e.g. PLCs) that require further code generators. In sum, generating code for such heterogeneous platforms is a very tedious and error-prone task. In practice, generated code often has to be modified to compensate for variations in the real-world setup. Such modifications are hard to re-integrate into the process models, so that they need to be done over and over again in the generated code. In this work, we employ an alternative approach that does not rely on code generation. Instead, an external control software is used that is able to interface with different robot and end-effector controllers. This software can be used for supervising the whole manufacturing process as well as hard real-time critical motion control. From our point of view, this approach greatly simplifies the software development for complex manufacturing processes in multi-functional robot cells.

3 Planning

For this paper, the planning and execution has been performed for a test component called *Demo Panel* which has been specified earlier in [12]. The structure is based on the lower fuselage of an Airbus A320, which is similar to a half cylinder with a radius of 1977 mm and a length of 1989 mm. In total, this component is manufactured from 208 dry carbon fiber cutpieces whereof 112 cutpieces must be handled with cooperating robots because of their size. The outer skin consists of 56 pieces with a length of 1989 mm and a width of up to 1031 mm. Another 56 cutpieces are for reinforcement with a length of up to 784 mm and a width of up to 1031 mm.

For the collision-free path planning of cooperating robots for aircraft CFRP production, a simulation environment called *CoCo* has been introduced in [9] and in [11] a planning approach for a fuselage production scenario with cooperating robots has been developed and tested in the *CoCo* environment. Since then, the planner has been developed further and been evaluated in a real scenario. **Figure 2** sequentially shows several stages of a simulated dry fiber placement process in the aforementioned *Technology Evaluation Cell* at the DLR in Augsburg. On the right side, a storage table is arranged where a planar cutpiece is positioned and can be picked up by the robot team. On the left side, the assembly mold or tooling is positioned where the robot team lays down the textile. The robots can move from right to left along a (not visualized) linear track. Each robot – both in the simulation and in the real world – is equipped with an end-effector which was developed earlier at the DLR [5]. The grippers are 1580 mm long and 370 mm wide and work with several vacuum modules whereby they can handle the fragile material very gentle.

The planning is divided into three major phases which can also be seen in **Figure 2**. First the planar cutpiece is picked up (cf. **Figures 2a** to **2c**), subsequently it is transported to the tooling (cf. **Figures 2d** and **2e**) where it is finally put to obtain its three-dimensional shape (cf. **Figure 2f**). The pick-up phase is calculated depending on grip points which define the position where the robots have to grasp the cutpiece. Both the pick-up and the lay-down points are specified by using special software for CFRP products like Composites Design (CPD) which is an extension for the CAD software CATIA from Dassault Systèmes. Hence, the accurate usage of CAD data in aircraft industries facilitates the modeling and planning of the manufacturing process remarkably. In a next step, the pick-up movement is calculated using the catenary function which describes how a robe or chain is bend which is hanging at two fixed points [11]. This function provides the angle and distance between the robots to lift the material.

During the transport phase, the robots turn the material over the linear unit where it can be transported collision-free to the middle of the assembly mold. After the center of the cylinder, i. e. the assembly mold, is reached with the center of the cutpiece, the robots move into the tooling along the cylinder length. In the next step, the robots are rotated around the center of the cylinder until the grippers are located over the lay-down position were the material is finally positioned. The last step is to determine the backward path which is simply the inverted movement from the transport phase until the pick-up preposition. All those steps need to be repeated for every cutpiece of one composite part.

4 Execution

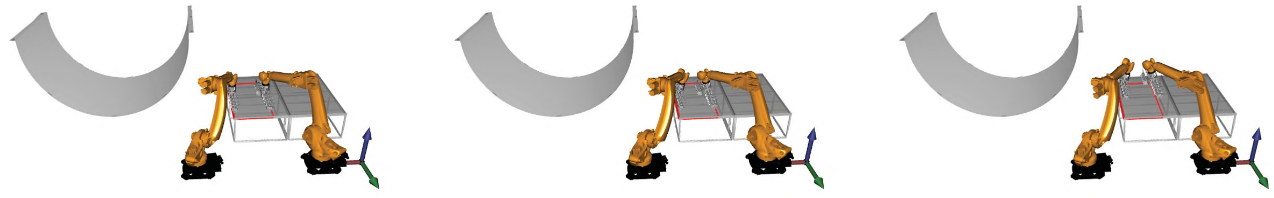
To connect the presented collision-free motion planner to the robot system, it was decided to use the Robotics API [13], which provides an object-oriented programming interface for industrial robots. The Robotics API can be used directly from a Java or C# application and allows to control multiple robots from within a single program. Applications developed with the Robotics API do not need to be executed using a real-time operating system, all operations which require hard real-time (such as motion interpolation or triggering tool actions) are automatically translated into an intermediate language and executed by an external motion controller called Robot Control Core (RCC) [14]. The RCC itself is developed using the C++ programming language and requires a real-time operating system; usually a standard Linux with Xenomai extensions is used. The communication between the application and the RCC is based on a plain TCP/IP connection.

The result of the *CoCo* path planning process consists of an ordered set of intermediate positions for both robots, enriched with meta-data for tool control (i. e. when to activate or deactivate the grippers). Each element in the result-set describes the exact pose of each robot (including the orientation of the tool) and the linear unit. Both robots must reach the positions described within a single element simultaneously in order to guarantee the collision-freeness of the trajectory, and to ensure the appropriate relation of one gripper to the other as required by the task (e. g. applying the proper angles and distance for the catenary function). The collision-free path planner generates an ordered set of intermediate positions for both robots.

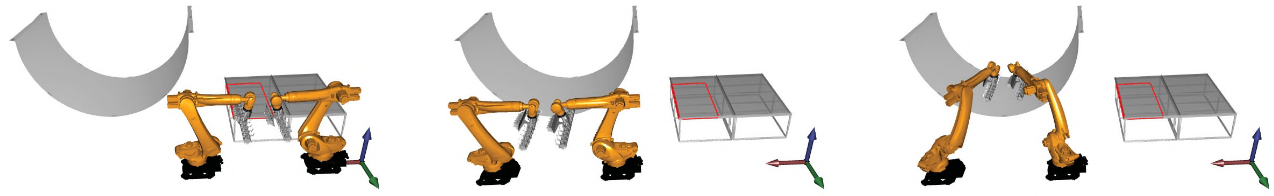
The intermediate positions generated by the path planner are still to coarse for direct hardware control (Cartesian distances are in the range from 5 cm to 100 cm). For a smooth execution of the path plan, fine interpolation is achieved by performing linear motions between adjacent intermediate positions which are precisely timed to reach the next position simultaneously with each robot. The Robotics API allows the specification of multiple synchronous motions for different robots which will be executed by the RCC with hard real-time guarantees.

The Robot Control Core executes the generated commands cyclically with a frequency of up to 1 kHz. In each execution cycle, joint coordinates are calculated for every robot, and, if required, tool actions are triggered. The RCC controls a KUKA robot using the *Remote Sensor Interface* (RSI) technology which allows for remotely manipulate a robot's motion in real-time. Communication between the RCC and the KUKA KRC-4 motion controllers is performed using a dedicated network connection and the exchange of XML telegrams embedded into UDP/IP packets. The overall architecture can be seen in **Figure 3**.

The KUKA controller transmits an UDP telegram every 4 ms containing the current position of the robot arm and all external axes (e. g. the linear unit) in both Cartesian and joint coordinates. A reply is expected from the external motion controller within at most 4 ms, thus hard real-time capabilities are required. If replies are lost, an emergency stop is automatically applied. The reply must contain the new set-points for all joints and additional data to control digital outputs connected directly to the KUKA controller.



(a) Start of pick-up **(b)** Intermediate catenary **(c)** Final catenary



(d) Start of transport at table **(e)** End of transport at tooling **(f)** Ongoing draping

Figure 2 Several planning stages during the dry fiber placement process using the CoCo environment

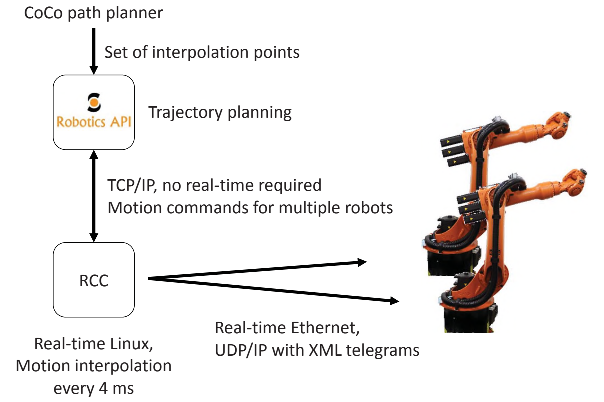


Figure 3 The architecture for cooperative real-time motion control using the Robotics API

The external motion controller must ensure that all provided set-points are reachable within one cycle time, which includes that an appropriate motion profile with limited acceleration and jerk values must be planned. Motion synchronization is achieved by the simultaneous execution of precisely timed linear motions between two adjacent intermediate positions.

5 Experimental Results

Using the Robotics API together with the real-time RCC offers a comfortable development and evaluation environment. Since this approach is not based on the traditional tool-stack provided by the robot manufacturer, tests have been performed (see **Section 5.1**) to show that the approach fulfills the accuracy requirements for the CFRP production scenario. The practical experiments, i. e. draping cutpieces for the *DemoPanel*, are briefly described in **Section 5.2**. All experiments have been conducted at the *Technology Evaluation Cell* consisting of two KUKA Quantec KR210 R3100 industrial robots on a shared 10 m linear track.

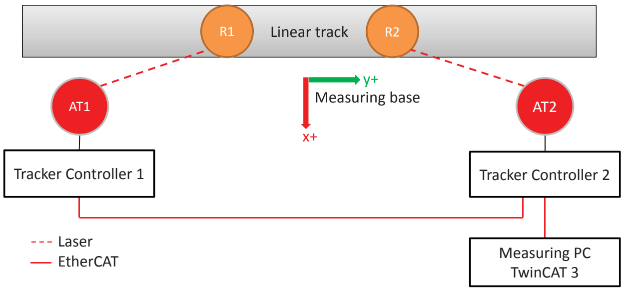


Figure 4 Structure of the *Technology Evaluation Cell* consisting of two robots (R1, R2) on a common linear track and two Leica Absolute Trackers (AT1, AT2) for accuracy evaluation

5.1 Accuracy measurements of external motion control

For the test set, two Leica Absolute Trackers AT901 LR were placed inside the robot cell, each measuring the position of one robot. Each of the two robots was equipped with a Leica T-MAC reflector, mounted nearby the flange. The trackers can determine the 6-DOF position of the T-MACs in their measuring area with a maximal permissible error of $10\,\mu\text{m}$ for distance and $15\,\mu\text{m} \pm 6\,\mu\text{m}$ for angular performance [15]. **Figure 4** displays the structure of the test setup: R1 and R2 describe the position of the robots on the linear track whereas AT1 and AT2 describe the position of their laser trackers. The controllers of both laser trackers provide the current measurement data in real-time using a EtherCAT fieldbus with a frequency of 1 kHz.

A measuring unit, which is connected to both controllers via EtherCAT, receives and processes the measurements. Using EtherCAT's *distributed clock*, it is possible to synchronize all clocks to $\leq 1\,\mu s$ [16] and thus to join the r sults of both tracking systems into a single data recor As the position of the T-MACs is measured by each lase tracker using a spherical coordinate system, the measurin unit translates these coordinates into a Cartesian coordina system based within each tracker. Then the position of o tracking system with respect to the other was determine which allows the measured data to being transformed into common base coordinate system: the measuring base. This base coordinate system has been defined approximately i the middle of the linear track (cf. **Figure 4**).

To evaluate the accuracy that can be achieved with coo erating robots using the Robotics API as external motio control, a set of motions has been defined for robot R while the second robot R2 was set to follow the other wit a defined distance. For the test, robot R1 had to perform t following motions, all specified with respect to the measu ing base:

1. Linear motion of 500 mm in Z-direction
2. Linear motion of 1000 mm in positive Y-direction an then of 1000 mm in negative Y-direction
3. Linear motion of 500 mm in positive X-direction an then of 500 mm in negative X-direction

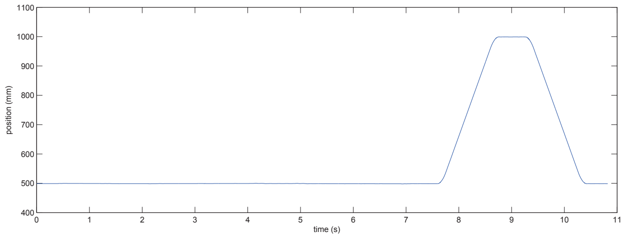
Figures 5a to **5c** show the measured position of robot R during the test in each direction. All motions were performed with a velocity of $0.5\,\mathrm{m\,s^{-1}}$ and an acceleration of $3\,\mathrm{m/s^2}$. The Robotics API uses a "Double-S" velocity pr file according to [17, Sect. 3.4] for linear motions whic limits the jerk.

Using the absolute position of both robots measured by t laser trackers, the deviation of the distance of both robo from the defined distance can be calculated. **Figure 5** shows this deviation, itemized for each coordinate axi The deviation is calculated by using the difference of th absolute positions of both robots and subtracting the difference of the first measured position (before starting the motions). It can be seen that the deviation is mostly in the range of ± 1 mm, however there are some peaks of up to 2 mm deviation. These peaks are mainly at positions where the velocity and acceleration of the robots were adjusted, possibly creating vibrations in the tool. Furthermore, perfect motion synchronization is hardly possible using external motion control, since both robot controllers use their own interpolation clocks and therefore request set-points for slightly different points in time. For the production of CFRP materials however, the achieved accuracy during the transport motions is sufficient. For the accuracy tests, linear units have not been moved.

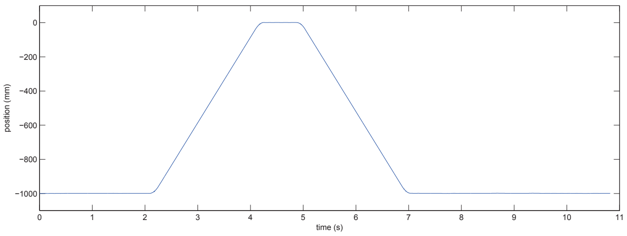
5.2 Path Planning results

For the evaluation of the path planner, a vertical prototype has been implemented which picks up a planar cutpiece and places it into the 3D tooling at the appropriate position. To enable a fully automatic planning of cooperative robot motions, composite engineering data needs to be used. After specifying valid grip points on the planar textile, CATIA's CPD can be used to transform these grip points into the three-dimensional preformed shape of the textile after draping it into the mold. Hence, these transformed points coincide with the final draping positions for the vacuum grippers. Furthermore, base coordinate systems for the tooling and the storage table in the real cell set-up need to be taught with respect to the coordinate system of the robots in order to accurately model the cell in both *CoCo* and the Robotics API.

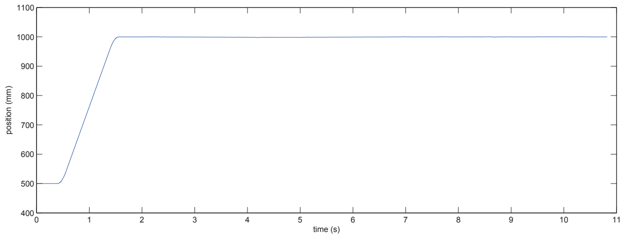
After feeding these measurements and the engineering data into the path planner, the whole cooperative motion for the dry fiber placement could be calculated automatically. As the Robotics API offers a 3-D visualization environment, the motions resulting from the *CoCo* path planner can be



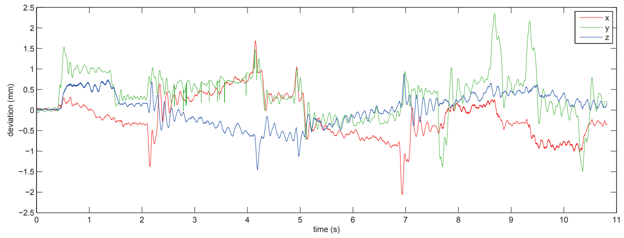
(a) X-coordinate of R1 in measuring base



(b) Y-coordinate of R1 in measuring base



(c) Z-coordinate of R1 in measuring base



(d) Deviation of robot flanges from their specified distance

Figure 5 Accuracy measurement results of cooperating robot motions

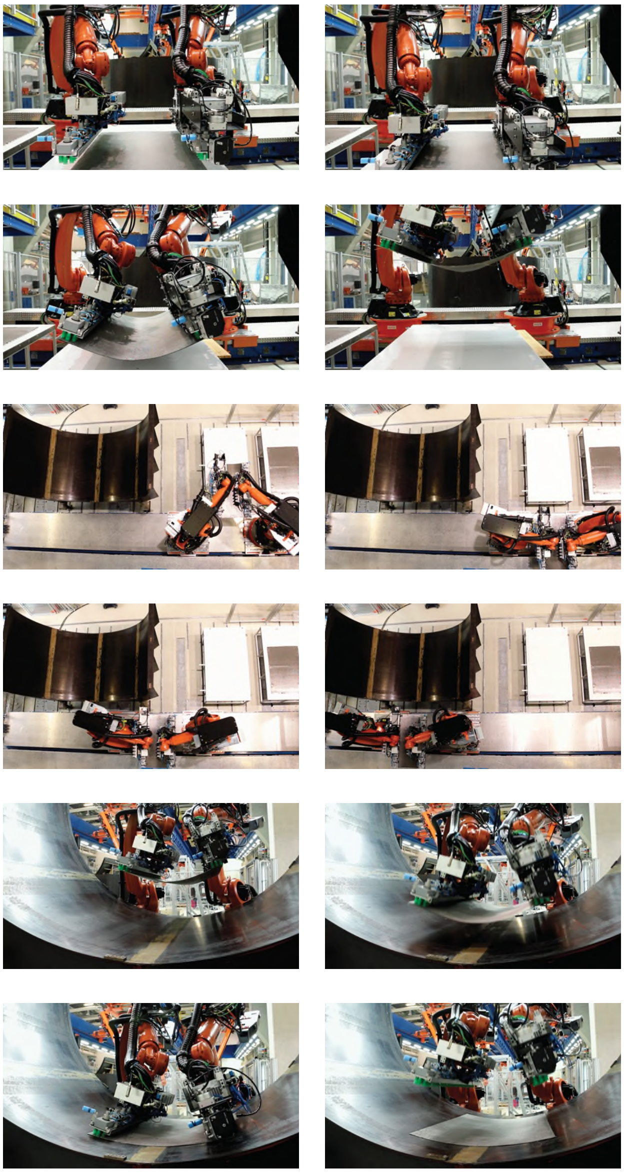


Figure 6 Execution of an automatically planned process for dry fiber placement of the lower half of an airplane fuselage.

tested offline. For offline simulation, the RCC is available for standard linux (without real-time extensions) and Windows. It includes drivers for simulating various robot and linear unit devices. Since the same path planning and interpolation algorithms are used in both hardware control and simulation, the precise path of the robots can already be tested in the simulation.

After successfully simulating the whole process, the program was executed in the *Technology Evaluation Cell* at the DLR and executed with real manipulators and linear units. For this purpose, the simulation drivers in the RCC just had to be replaced by other drivers that interface the hardware, in this case via RSI. The application code containing the robot motion commands and their synchronization, depending on the output of CoCo, was not changed and no code generation process was necessary. **Figure 6** and the attached video[1] show the execution in the Technology Evaluation Cell. The cutpiece was transported by both robots with precisely the required transport position and laid into the tooling at the required position. Moreover, the vaccum and heating modules of both grippers were controlled using digital I/Os. Except the initial measurement of the aforementioned base coordinate systems, no manual interaction has been necessary. Hence, further cutpieces can also easily be laid without any further manual intervention which increases the performance of the automated fiber placement tremendously.

6 Conclusion

In this paper, we introduced a novel approach for the automated manufacturing of CFRP materials using cooperating industrial robots. Production of large-scale lightweight aircraft components requires the handling of numerous large and fragile textile cutpieces which cannot be processed with a single robot. Manually creating robot programs for such a large numbers of different pieces is an extremely tedious and uneconomic task. Thus, the automatic planning of appropriate motions and their execution is required. For cooperating robots, special attention needs to be paid to collision-free motions, since the end effectors can get very close (cf. **Figure 6**).

To facilitate the development of the collision-free path planner *CoCo*, it has been connected with the *Robotics API* which allows for both 3D simulation and real hardware control from a single program for multiple robots. At the DLR, the Robotics API externally controls two KUKA Quantec KR210 robots on a linear track using RSI. With two Leica Absolute Trackers, it could be demonstrated that the precision using external motion control is perfectly suitable for the handling of CFRP materials. Deviations of the defined and real positions of both robots only occur during the transport phases, and are rather small ($\approx 2\,\mathrm{mm}$) compared to the size of the textiles (up to 2000 mm). Once the robot has stopped, the same precision as with traditional robot programming is achieved. Measurements for the absolute precision of the robots have to be performed separately; however the Robotics API in general supports realtime sensor-based feedback (e.g. supplied by laser trackers) for position corrections. An evaluation with real CFRP textiles has shown successfully that the overall production process can be performed using the pre-calculated collision-free path for two robots.

The approach presented in this work for connecting offline path planning with simulation and execution has several advantages. On the one hand, no code generation is necessary, which means the exact same program that has been simulated and tested offline is run on the real robot cell. This is particularly advantageous in multi-robot cooperation scenarios like in CFRP manufacturing, as code gen-

---

[1] Available online: `http://video.isse.de/coco_isr2016/`

eration for multi-robot systems is not commonly available in today's offline programming tools. Using the Robotics API, it is possible to control and synchronize multiple robots, robot tools and linear units from one application. This can save considerable development effort compared to industrial solutions that require developing one program per robot and synchronization among those programs. Finally, the Robotics API and the RCC support real-time motion correction, also for multi-robot systems, to e.g. compensate for inaccuracies in motions of such large robot systems based on inputs of sensors or kinematic and dynamic models.

In [8] an offline programming platform for the robot-based CFRP production has been introduced. However, this platform was designed for single robots only. A next step will be the integration of the *CoCo* path planner which then allows for automated planning for the production of a complete CFRP part with as little manual intervention as possible. Using the simulation technology integrated in the offline programming platform and the Robotics API, it is possible to preview the whole process before actually executing exactly the same motions on the real system.

7 Literature

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

[2] A. Angerer, C. Ehinger, A. Hoffmann, W. Reif, and G. Reinhart, "Design of an automation system for preforming processes in aerospace industries," in *Proc. 7th IEEE Conf. on Autom. Science and Engineering, Trieste, Italy*. IEEE, 2011, pp. 557–562.

[3] C. Löchte, H. Kunz, R. Schnurr, F. Dietrich, A. Raatz, K. Dilger, and K. Dröder, "Form-flexible handling technology for automated preforming," in *Proc. 19th Int. Conf. on Composite Materials, Montreal, Canada*, 2013.

[4] T. Gerngroß and D. Nieberl, "Automated manufacturing of large, three-dimensional CFRP parts from dry textiles," in *SAMPE EUROPE Technical Conf. & Table-Top Exhib.*, Sep. 2014.

[5] T. Gerngroß, F. Krebs, and A. Buchheim, "Automated production of large preforms based on robot-robot cooperation," in *Proc. 4th European Conf. on Materials and Structures in Aerospace, Hamburg, Germany*. VDI, 2012.

[6] A. Angerer, M. Vistein, A. Hoffmann, W. Reif, F. Krebs, and M. Schönheits, "Towards multifunctional robot-based automation systems," in *Proc. 12th Intl. Conf. on Inform. in Control, Autom. & Robot., Colmar, France*. SciTePress, 2015, pp. 438–443.

[7] F. Krebs, L. Larsen, G. Braun, and W. Dudenhausen, "Design of a multifunctional cell for aerospace CFRP production," in *Advances in Sustainable and Competitive Manufacturing Systems*, ser. LNME. Springer, 2013, pp. 515–524.

[8] L. Nägele, M. Macho, A. Angerer, A. Hoffmann, M. Vistein, M. Schönheits, and W. Reif, "A backward-oriented approach for offline programming of complex manufacturing tasks," in *Proc. 6th Intl. Conf. on Autom., Robot. & Appl., Queenstown, New Zealand*. IEEE, 2015, pp. 124–130.

[9] L. Larsen, J. Kim, and M. Kupke, "Intelligent path panning towards collision-free cooperating industrial robots," in *Proc. 11th Intl. Conf. on Inform. in Control, Autom. & Robot., Doctoral Consortium, Vienna, Austria*. SciTePress, 2014, pp. 39–47.

[10] Y. Gan, X. Dai, and D. Li, "Off-line programming techniques for multirobot cooperation system," *Int. J. Adv. Robot. Syst.*, vol. 10, no. 282, 2013.

[11] L. Larsen, V.-L. Pham, J. Kim, and M. Kupke, "Collision-free path planning of industrial cooperating robots for aircraft fuselage production," in *Proc. 2015 IEEE Intl. Conf. on Robot. & Autom., Seattle, WA, USA*. IEEE, 2015, pp. 2042–2047.

[12] M. Eckardt, "Einsatz von kooperierend arbeitenden Robotern zur automatisierten Erstellung eines Preformaufbaus," in *Proc. Deutscher Luft- und Raumfahrtkongress, Augsburg, Germany*, 2014, in German.

[13] A. Angerer, A. Hoffmann, A. Schierl, M. Vistein, and W. Reif, "Robotics API: Object-Oriented Software Development for Industrial Robots," *J. of Softw. Eng. for Robotics*, vol. 4, no. 1, pp. 1–22, 2013.

[14] M. Vistein, A. Angerer, A. Hoffmann, A. Schierl, and W. Reif, "Flexible and continuous execution of realtime critical robotic tasks," *Intl. J. Mechatronics & Autom.*, vol. 4, no. 1, 2014.

[15] *Leica Absolute Tracker ASME B89.4.19-2006 Specifications*, 2008.

[16] *EtherCAT Slave Implementation Guide*, EtherCAT Technology Group.

[17] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Springer, 2008.