2017 13th IEEE Conference on Automation Science and Engineering (CASE)
Xi'an, China, August 20-23, 2017

Full Automatic Path Planning of Cooperating Robots in Industrial Applications

Lars Larsen[1] and Manuel Kaspar[1] and Alfons Schuster[1] and Michael Vistein[1] and Jonghwa Kim[2] and Michael Kupke[1]

Abstract— **Parts made of carbon fiber reinforced plastics (CFRP) for airplane components can be so huge that a single industrial robot is no longer able to handle them, and cooperating robots are required. Manual programming of cooperating robots is difficult, but with large numbers of different sized and shaped cut-pieces, it is almost impossible. This paper presents an automated production system consisting of a camera for the precise detection of the position of each cut-piece and a collision-free path planner which can dynamically react to different positions for the transfer motions. The path is planned for multiple robots adhering to motion constrains, such as the requirement that the textile cut-piece must form a catenary which can change during transport. Additionally a technique based on machine learning has been implemented which correctly resolves redundancy for a linear axis during planning. Finally, all components are tested on a real robot system in industrial scale.**

Index Terms - Intelligent and Flexible Manufacturing, Industrial Robots, Cooperative Manipulators, Motion and Path Planning, Robust/Adaptive Control of Robotic Systems

I. INTRODUCTION

Nowadays in nearly all manufacturing facilities industrial robots are used to automate operational procedures. Currently, the term industry 4.0 is omnipresent. It describes the change of production facilities to smart factories and the computerization of manufacturing [1]. The goal is to get away from static and very specialized production scenarios, where robots execute the same task several thousand times – like spot welding in the automotive industry. Autonomous operations are the enablers of industry 4.0 systems. The market demands for much more flexible production systems that can easily be reconfigured to produce many different products. Ideally the reconfiguration should be so easy that it would be possible to produce just one exemplar of a product.

Traditionally industrial robots are equipped with a program which clearly defines all necessary steps and can only react to small deviations in the process. The program is either created "on-line" (using the real hardware for development) or "off-line" (using a simulation environment). In both cases, the concrete process is encoded into the program. Furthermore, both methods have the problem that the programming can be very time consuming in complex environments and it takes a large amount of time to adapt to small changes in the process.

To save weight, and therefore also fuel, many modern airplane components are manufactured from CFRP. One possible production process consists of the forming of the final workpiece from dry, textile-like cut-pieces using a positive or negative mould. Once all cut-pieces are at the right place, resin is infused and hardened. For the best stability of the components, large cut-pieces are desirable. At the moment the lay-up of those is a manual, labor intensive task. A single industrial robot often is not able to handle such cut-pieces, therefore cooperating robots are used. Many of the numerous cut-pieces are of different size and shape, creating a very high complexity for the required robot programs. Thus it is highly desirable also in this scenario to have a flexible and easily (automatically) reconfigurable production system. In this work we introduce a fully automatic CFRP preforming system that supports cooperating industrial robots. It automatically detects the position of the CFRP cut-piece in a storage with a camera based system as well as the final positions in the mould based on computer-aided design (CAD) data and calculates a collision-free path for the robots. The system can be used for any static production scenario with one or more robots which can be fixed or on a linear axis. In this publication we use the system for a scenario with two robots on a common linear axis. Typically in a master-slave application the relation between the two robots is fixed. A typical master-slave example is the technology package RoboTeam by KUKA where the Tool Center Point (TCP) can be fixed geometrically during motions. In our approach the relation between the TCPs is described by a catenary function where the coefficient can change during the planning process. Furthermore it is possible to describe the relation by any other mathematical function. The system is evaluated using a real robot facility, consisting of two 6-Degrees of Freedom (DOF) KUKA Quantec KR-210 R3100 industrial robots mounted on a shared linear unit. To solve the 7-DOF a machine learning method based on random decision forest has been used which predicts the appropriate positions. The system is demonstrated on a production scenario of the lower fuselage of an Airbus A321. The overall scenario for the demonstration is depicted in Fig. 1. Both robots are equipped with endeffectors called *stripe grippers* to handle the CFRP material. The cut-pieces are delivered to the robotic cell in different drawers of a storage system.

[1]Lars-Christian Larsen, Alfons Schuster, Michael Vistein and Michael Kupke are at the German Aerospace Center (DLR), Center for Lightweight-Production-Technology (ZLP), Am Technologiezentrum 4, D-86159 Augsburg, Germany {lars.larsen, manuel.kaspar, alfons.schuster, michael.vistein, michael.kupke}@dlr.de

[2]Jonghwa Kim is with the University of Science & Technology (UST), 217 Gajeong-ro, 34113 Daejon, Korea kim@ieee.org

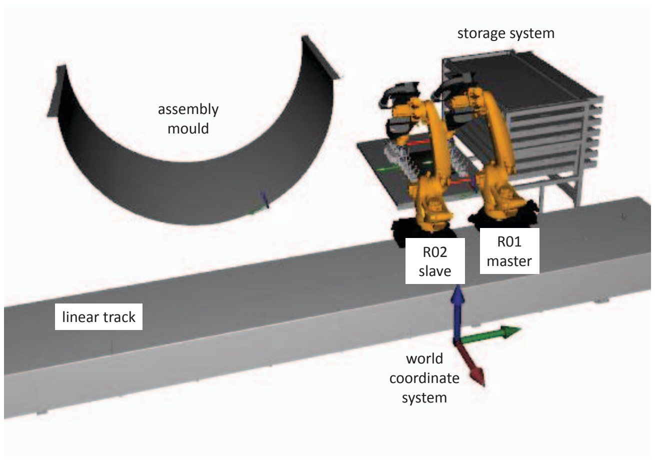


Fig. 1: Fuselage production scenario with a storage system on the right, two KUKA Quantec KR-210 R3100 robots - R1 (master) right, R2 (slave) left - on a linear track and an assembly mould on the left. The world coordinate system is in the lower part of the picture. Directions are: red → x, green → y, blue → z.

II. Related Work

Smith *et al.* [2] define a hierarchical classification of different types of two-arm manipulation. Firstly, there is the *un-coordinated* type where two arms execute different tasks within the same workspace, and secondly the *coordinated* type which itself is further subdivided to the *goal-coordinated* and *bi-manual* types. In a goal-coordinated scenario, both arms are palleting parts into the same box while in a bi-manual both arms are lifting and moving the same box full of parts.

In this publication, the bi-manual scenario for cooperation is dealt with where the robot arms build a closed kinematic chain. Additionally in our scenario the robots are mounted on a linear track and grip an object which is flexible and must be handled in the catenary form during re-orientation. The problem of the closed chain results in the fact that the configuration space does not build a manifold any more but rather a variety. Thus the probability to find a good state which satisfies the so called closed-constraints is impossible [3, p.52], [4]. The reason is that the positions of all 14 axes cannot be determined stand-alone because they depend on each other. Thereby usual sampling based algorithms can not find a solution for such problems [5, p. 167 f., 338 ff.]. There are several publications (e.g. [3], [6]–[8]) which deal with the closed chain problem, but all handle a static object between the robots.

In the industrial context, a typical scenario for cooperating robots are load sharing applications or the transport of huge parts. Usually publications in this area avoid the closed chain problem by planning for the first robot and afterwards for the second. We also choose that principle but to our knowledge state of the art publications do not handle flexible objects whereby the relation (distance and orientation) between the robots is changing during the transport. The scenario which is considered here explores the cooperative transport of a carbon fiber cut-piece whose properties are similar to a piece of textile.

A. Programming of cooperating industrial robots

Usually there are two ways of programming industrial robots: *on-line* [9, p. 337] and *off-line* [9, p. 353] programming. For on-line programming (also called *teach-in*) the real robot must be moved to each point of the path by hand, and the user has to define the acceleration and velocity for each path segment. After first finishing a collision-free trajectory, the programmer can further optimize the position of each point as well as the acceleration and velocity for each segment.

The second possibility is off-line programming (OLP). The robot is programmed in a simulation environment and afterwards the appropriate code for the robot is generated by a post processor. The programming itself is similar to on-line variant: The user has to move the robot in the simulation to the desired path points and define the acceleration and velocity values for these segments. There are many different products on the market from various manufacturers e.g. KUKA.Sim Pro or ABB RobotStudio. Nof [9, p. 353] lists some advantages from off-line programming: reduction of robot down time, verification of robot programs, removal of the programmer from dangerous environments and a single programming system. After finishing coding the instructions have to be translated to the appropriate robot language and the program must be loaded on the robot controller. Afterwards the code must be tested and, if necessary, adjusted to the small differences of CAD models and the real world. Gan *et al.* [10] and [11] provide an overview of existing OLP tools. The big drawback of these tools is, that they do not support the programming of cooperating robots.

B. Path planning for cooperative industrial robots

As already mentioned path planning for cooperative robots itself is a highly complex process due to the various possibilities for positioning each robot. Variable positions of the cut-pieces during the grip and drop phases add another level of complexity which results in manual programming being at least very time consuming, if not infeasible. In order to react to "live" changes during the process (like a different position of the cut-pieces on the table) which can be detected using a camera, a dynamic planning system is necessary.

Path planning of closed chains has already been addressed before 1999 by [4], [12] but at the moment current state-of-the-art approaches touch the borders of full automatic generation of assembly processes. Most of the publications deal either with the path planning for multi robot scenarios in simulation or with the real time synchronization of industrial robots in order to realize e.g. load sharing applications. There are very few publications which cover the combination of these two problems.

The following publications deal with planning in a simulation environment. Lahouar *et al.* [13] present a system for up to three PUMA560 robots using a real-time local method based on constraints in a dynamic environment. Pellegrinelli *et al.* [14] and [15] point out that multi robot path planning has still a lot of open issues which should be solved by multi-disciplinary activities and research fields. The focus of their work is on the optimization of the layout of a production cell for multiple welding manipulators. Zribi *et al.* [16] introduce a force controlled system where two planar robot arms handle a constrained object. Tzafestas *et al.* [12] introduce a master-and-two-slaves system for the planning of three cooperating robots which handle a large triangle shaped workpiece. The TCP-positions are calculated by using homogeneous transformations from the motion of the manipulated object. Gan *et al.* [11] use SolidWorks, a 3D design and modeling software which can also be used as OLP tool. The software does not support programming cooperating robots out of the box, but an Application Programming Interface (API) is offered which can be used to automate and customize the software.

These two publications introduce control strategies for multi robots. Gudiño-Lau *et al.* [17] present a control strategy for two 3-DOF industrial robots which hold a rigid body. The robots are kinematically and dynamically coupled while holding the workpiece. Mediavilla *et al.* [18] propose a path planning method where many robots cooperate in a dynamic environment. In the first step many paths for the robots are generated off-line and randomly. In the second, on-line step, the robots evolve a collision free path to their goal.

As already mentioned the combination of path planning for cooperating robots and execution on real hardware is sparsely discussed in literature. We present a full automatic planning and execution system in an industrial scale producing aircraft fuselage parts. In times of often changing and widely ranging products, such a system can be decisive to be at the forefront. With the use of the system in CFRP production, its capabilities can be demonstrated even more. Dry fiber cut-pieces are very similar to a piece of textile. Therefore the requirements are completely different to the ones in metal processing. Because an unforeseeable quality inspection step can happen which must be carried out immediately, it must be possible to dynamically re-plan the next steps of the robots without a downtime of the facility. In this work we present a system which can provide such a function.

III. Automatic Path Planning of Industrial Robots

The automated production system presented in this paper has been evaluated on a test component called *DemoPanel* which has been specified in [19]. It is a full scale demonstrator which is based on the geometry of the lower fuselage of an Airbus A321. The structure is similar to an half cylinder with a radius of $1977\,\mathrm{mm}$ and a length of $1989\,\mathrm{mm}$. The process is a pick&place application where the cut-pieces are picked up with an endeffector (which works similar to a vacuum cleaner) in 2D from a storage system and is positioned into a mould at the proper coordinate in 3D. In sum there are 208 different dry fiber carbon cut-pieces which must be handled and laid up in the tool form. 112

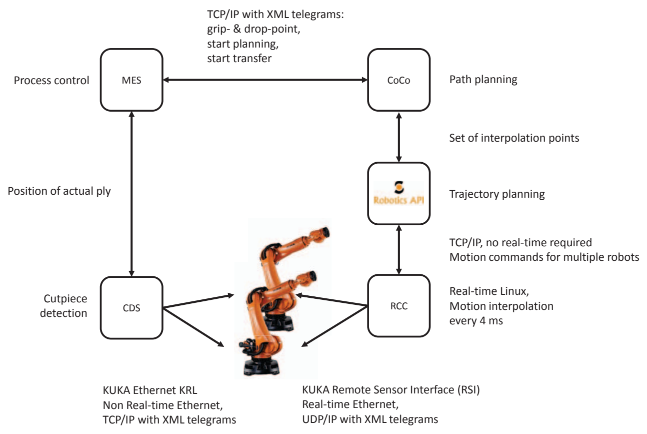


Fig. 2: Architecture of the system with corresponding communication messages and protocols.

cut-pieces are $1989\,\mathrm{mm}$ long and $1031\,\mathrm{mm}$ wide and thus must be transported with two cooperating robots. After the cut-pieces are placed in the mould, heaters are extended from the endeffector to melt a thermoplastic binder on the material which bonds temporarily to the lower layers.

Today there is typically a gap between the computer-aided x (CAx)-environment and the automation solution that, depending on the process, can be overcome by third party software, e. g. in the case of tape-laying. Since there is no out-of-the-box solution for our application we had to implement a solution for closing this gap. We chose a way that requires almost no user interaction, especially no teaching in order to enable automated, autonomous production already beginning at the CAx-side. In the regular CAx-Part the part is designed by defining the layup in 3D. After that the 2D-Geometry of the semi finished material is determined by draping simulation (so called flattening). We extended the CAx-Chain by determining a good gripper placement on the 2D-material by maximizing the number of active vacuum cups with respect to some boundary conditions (e. g. not swapping the grippers or keeping grippers parallel) and then reverting the flattening back to 3D by CATIA's feature "geometry transfer", yielding the meta-information for the layup: Grip-points plus TCP, drop-points plus TCP, active vacuum cups and welders.

A. System architecture

To achieve a completely automated production process, different components need to be integrated into a single system. The overall architecture is illustrated in Fig. 2. Components of the system are the Manufacturing Execution System (MES), the Cutpiece Detection System (CDS), the Collsion free cooperation (CoCo) path planning system, the Robotics API and the Robot Control Core (RCC). All systems communicate using Ethernet, although different networks are used depending on whether there are real-time requirements or not.

The cut-piece detection and the collision-free transfer systems have different requirements for robot control. While the cut-piece detection system only employs a single robot and can rely on several pre-programmed positions and paths for taking relevant camera pictures, the transfer system must synchronize multiple robots with very high timing precision. Therefore, two different approaches to robot control have been used. The CDS uses the non-real-time EthernetKRL interface of the KUKA robot which allows to trigger pre-programmed motions using a Transmission Control Protocol/Internet Protocol (TCP/IP) connection. In contrast, CoCo uses the Robotics API and the KUKA real-time Remote Sensor Interface (RSI) to control both robots with a high, deterministic frequency.

The MES is responsible for the overall coordination of the tasks and delegates control to either the cut-piece detection or the transfer system. In addition, the MES is also responsible for the activation of the proper control mode (real-time/non real-time) on the robot controller and the transmission of the grip- and drop-points calculated by the CDS to CoCo.

For the execution of the planned motions on real robotics hardware, the Robotics API development platform [20] is used. This platform provides an object oriented application programming interface (for the Java and C# languages) and allows the execution of real-time critical tasks for multi-robot systems. Precise and deterministic timing is essential for cooperating robots in order not to damage the cut-pieces, e.g. by shearing during transport.

Internally, the Robotics API development platform is split into two separate parts. The robotics application can access a rich class library from the Robotics API which contains algorithms for trajectory planning or (inverse) kinematics. Motion tasks generated using this library are automatically converted into data-flow graphs (so called Real-time Primitive Interface (RPI) nets, cf. [21]). The second part of the platform, the RCC, runs on a real-time operating system (usually Linux/Xenomai) and can interpret RPI nets deterministically in a real-time environment. All RPI nets are self contained, i.e. no real-time communication between the application and the RCC is required. Therefore, a standard TCP/IP connection is used. The RCC directly communicates with all hardware systems. For the example of KUKA industrial robots, the RSI protocol based on User Datagram Protocol/Internet Protocol (UDP/IP) on a discrete Ethernet network is used.

B. Cut-piece detection

In CFRP production the part design aims at an optimal shaping of the 2D cuts in order to achieve a good approximation of the later desired 3D shape. Due to the resulting complexity of the layup comprising hundreds of cut-pieces, extensive care has to be taken to correctly grip and place the cut-pieces. Robotic gripping needs a well known position and a correct alignment of every single cut-piece, leading to a situation where productivity potentials during layup are hard to tap due to previous alignment of the cut-pieces by manual work. This can be overcome by means of computer vision and comparing the 2D shapes from CAD to a camera image [22]. In the case of carbon fibers which can not be marked and show a strong spatial anisotropy in reflectivity, best results were achieved by template-matching rotated bitmap-representations of the 2D CAD shapes to the camera image, yielding coordinates and rotation for gripping the more or less randomly placed cut-pieces. Raster images of the same resolution as the camera are created in memory by using the 2D contours of the actual cut-piece, e.g. using zeros for cut-piece and ones for background. To avoid false negative detections caused by nearby cut-pieces only a ≈2cm band of ones is applied around the shape, the rest of the image is marked as transparent, e.g. by twos. By rotating this bitmaps and calculating the normalized cross correlation between every equally sized camera sub-image A_i at position x, y and every bitmap B_i at angle ϕ while skipping the transparent pixels, both x-y-coordinates and rotation angle ϕ are obtained. For normalization, the averages μ_A and μ_B and standard deviations σ_A and σ_B of sub-image and rotated image are computed. The covariance $v(A, B)$ is obtained by multiplying the pixel-values corrected by their averages. Finally, the normalized cross correlation c is calculated by dividing $v(A, B)$ by both standard deviations σ_A and σ_B (Eqs. (1) to (4)).

$$c = \frac{v(A,B)}{\sigma_A \sigma_B} \tag{1}$$

$$v(A,B) = \frac{1}{n}\sum_{i=1}^{n}(A_i - \mu_A)(B_i - \mu_B) \tag{2}$$

$$\sigma_A = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(A_i - \mu_A)^2} \quad \sigma_B = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(B_i - \mu_B)^2} \tag{3}$$

$$\mu_A = \frac{1}{n}\sum_{i=1}^{n}A_i \quad \mu_B = \frac{1}{n}\sum_{i=1}^{n}B_i \tag{4}$$

This enables to distinguish good and poor correlation values by simple thresholding, since a good correlation will be near the value of 1.0 (in practice greater than ≈0.9 depending on the materials used), even when comparing a b/w bitmap created in memory with a camera image that will always vary with exposure and illumination. Since the process is very time consuming, Gaussian pyramids of the images are used for rough detection and later expanded to increase resolution, and finally subpixel interpolation is used to maximize accuracy. A commonly used alternative to speed up the cross-correlation, which is actually a convolution, is to perform template matching in Fourier space by exploiting the Fourier transformations properties stated in the convolution theorem. This was considered, but proved inappropriate due to the incompatibility of handling transparency together with normalization.

The image coordinates are transformed to robot coordinates by using a simple perspective transform obtained by reference points in the image corners. For maximum detection accuracy it proved necessary to reposition the camera and perform a second detection at the camera's sweet spot.

C. Collision-free path planning

The CoCo simulation and planning environment for collision-free path planning in cooperating industrial applications was introduced in [23], and in [24] a static planning approach for a fuselage production system with cooperating robots has been developed and tested. The system was further improved and tested on a real robotic facility in [25]. The CoCo system is created using the C# programming language and the Helix 3D Toolkit[1] for visualization. As improvement to [24], where the collisions were detected with oriented bounding boxes (OBB), the detection is now realized with the BEPU physics[2] engine. Internally convex hulls of the objects are used to speed up the calculation. The carbon fiber cut-piece and the mould are internally represented by a mesh.

For path planning the Open Motion Planning Library (OMPL) [26] was connected to CoCo. OMPL is a library which contains many motion planning algorithms and techniques like path smoothing, experience based planning, parallel execution etc. The big advantage of using OMPL is that it is very flexible and includes most state of the art algorithms such as Rapidly-exploring random tree (RRT), Probabilistic Roadmap (PRM) or Search Tree with Resolution Independent Density Estimation (STRIDE). Because both robots in the transportation process form a closed kinematic chain which makes it impossible for normal, sampling based algorithms to work, our solution is to stepwise plan for master-robot first and afterwards determine the position for the slave by catenary function and inverse kinematic with master robot as collision object. The proposed planner can plan either in joint- or SE(3)-space. As inverse kinematic an analytical solution presented in [27] was used.

The three main parts of our planning implementation - catenary calculation, external axis position determination and path planning are described in the following sections.

1) Catenary: The catenary describes the hanging of a chain or cable under its own weight when only fixed at their ends and is e.g. used for the design of suspension bridges. Formerly even Galileo thought the catenary can be described as a parable but strictly speaking it must be described by Eq. (5). As already mentioned carbon fiber cut-pieces behave similar to a catenary when fixed on both ends. To avoid damages to the fibers it is necessary to transport the material in that form. Fig. 3 and Fig. 4 show different catenary configurations. Given that we grip the material on a table in 2D and drape it during transport to a 3D form in which we finally lay up we need to calculate the shape, grippoints and angles. The blue line describes the form of the cut-piece and the red dots the grippoints.

[1]https://github.com/helix-toolkit
[2]https://bepuphysics.codeplex.com/

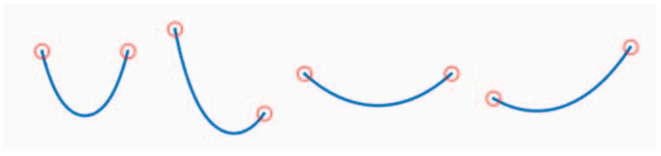


Fig. 3: Different catenary configurations describing the relationship between two robots holding a carbon fiber cut-piece.

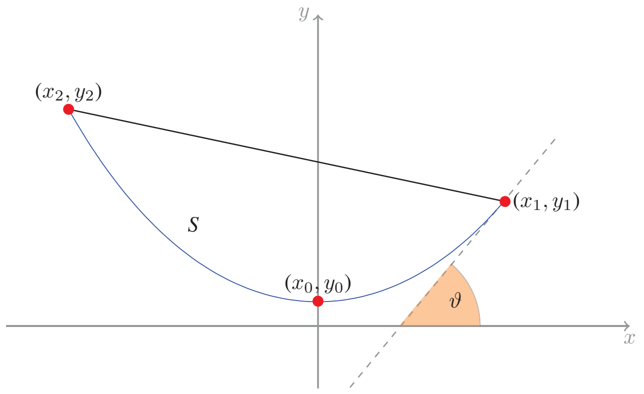


Fig. 4: catenary parameters: S cut-piece length, (x_2, y_2) grippoint R2, (x_1, y_1) grippoint R1, (x_0, y_0) lowermost point, ϑ angle gripper R1

$$y = \frac{a}{2} \cdot \left(e^{\frac{x-x_0}{a}} + e^{\frac{-(x-x_0)}{a}} \right) + y_0 = a \cdot \cosh\left(\frac{x-x_0}{a}\right) + y_0 \tag{5}$$

The variable a marks the radius of curvature and the origin of the x-coordinate is alway located on the lower most point of the catenary. x_0 describes an x-offset and y_0 an y-offset. During the planning the catenary is used for two different configurations with and without constraints. For both cases a non-linear equation system has been established which is numerically solved using alglib[3].
Planning without constraints: In this case the length of the cut-piece (S), the angle of gripper of the master and the desired distance between the grippers ($gripDistance$) is given. With that information the angle ϑ for the gripper of the slave can be calculated.
Planning with constraints: The two suspension points are on the same z-height during transport. For the planning the angle ϑ of the grippers can be determined by the function.

2) Setting of external axis position (redundancy solving): The KUKA Quantec 6-DOF robots used in the experiments are mounted on a linear track and therefore have in total 7-DOF. To automatically predict an external axis position a machine learning method has been implemented and tested. The big advantage of this method is that it is flexible enough to be used in different scenarios. The drawback is that during the planing the machine learner has to predict many robot positions which reduces the planning speed. Before using the classifier, training samples have to be created. Fig. 5

[3]http://www.alglib.net/

TABLE I: Classification results for external axis prediction.

Classifier	R01	R02
SVM	78%	50.7%
KNN	99%	97.0%
Complex Tree	98%	86.9%
Boosted Tree	63.1%	42.4%

shows the process exemplary for one configuration. Initially a random TCP-position is generated which is located in the middle of the interval at index six in the picture. The axis direction correlates to the y-direction in Fig. 1). In the next step the robot is set to different external axis position (from 0 to 12) while the TCP remains on the actual TCP-position. If a collision is detected for a configuration it is labeled as not valid otherwise as valid. If several valid (collision-free) indices are present for one TCP-position the mean of the biggest interval is taken. In Fig. 5 the biggest interval is between 6 and 10. Finally the center of the biggest interval is chosen as an valid external axis position for the current TCP. A sample for the master consists of its cartesian TCP- and external axis-position. The slave sample contains its cartesian TCP-, external axis- and the external axis-position of the master.

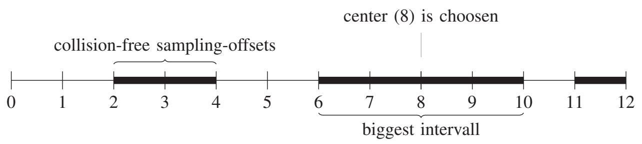


Fig. 5: Sampling of external axis position right and left in y-direction from Fig. 1 - TCP is located in the middle of the complete interval at index six

For the classification first of all different classifiers like Support Vector Machine (SVM), k-Nearest Neighbors (KNN) were tested in the classification learner toolbox of Matlab. The classification was performed with 7877 samples both for master and slave using ten fold cross validation. For the master the x-, y- and z-coordinate of the TCP were used as features and the predicted external axis offset was used as label. For the slave robot additionally the position of the master was used as feature. Table I shows an overview of the classification results. For SVM a linear kernel and one-vs-one was used. KNN was performed with ten neighbors and euclidean distance weight. For the complex tree the maximum number of splits was set to 100 with ginis diversity index split criterion and no surrogate decision splits. The ensemble tree was done with AdaBoost, 200 learners and 0.1 learnings rate. After a successful validation of KNN and Complex Trees in Matlab these two methods where integrated in the CoCo environment.

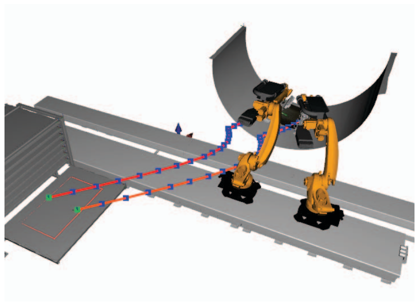
3) Planning: The planning is done stepwise first for the master and afterwards for the slave according to the following pattern. First a collision free path is planned with OMPL together with the external axis machine learning. For each collision free sample of the master first of all the catenary is calculated to get the TCP-position for the slave robot. At this point a configuration for the slave is determined using inverse kinematics and external axis prediction technique.

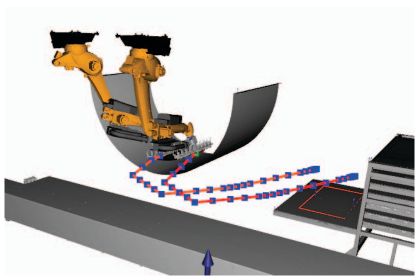
Before using the planners of OMPL in the DemoPanel scenario a pre-study comparing different planers has been done to find the most suitable planner in an industrial context [28].

The study indicates that planning with support points is much faster, because a direct connection in SE(3) is possible from point to point without hitting an obstacle. Support points can be defined by the user on a meaningful position to facilitate the planning. In contrast, planning without support points leads to shorter paths with less joint movements but longer planning time. In average, these paths are also very short, in particular compared to other non-optimizing planners. For our DemoPanel scene, PRM is the best algorithm for planning without support points. If support points are available, RRTConnect is faster. For this scenario the planning in SE(3)-space seems to be faster with support points and the quality of the paths is much better. Without support points the joint-space provides better output.

IV. Experimental results

After validating the different planners in the simulation the whole system was tested on the robotic cell. For planning RRTconnect was used. During our experiments we tested the system with 5 different cut-pieces which together form the first layer of the component. For all pieces a collision free path could be automatically found and a successful lay-up has been executed. Although the distance between the corpus of the drawer system and the assembly jig is just $2750\,\mathrm{mm}$ it is no problem for the planning system to find a path for the $1031\,\mathrm{mm}$ wide cut-piece with enough safety distance to obstacles. To speed up the computation during the experiments support points were defined. For the calculation the system needed about one minute per cut-piece which is compared to the manual teach-in where in contrast the CoCo-system needed about one minute per cut-piece. The execution on the robotic hardware was done in operation mode T2 (for safety reasons, since due to space restrictions no safety fence was available) with 50% of maximal robot speed and acceleration. Execution in automatic mode is possible once all required safety equipment is installed. The overall speed of the robot motions however is limited by the characteristics of the material. The ply acts like a sail during transport, therefore the speed is limited and far below the theoretical maximum speed of the actuators. Another very important point which was analyzed during the experiments was the stability of the system which is very important for an industrial use. It turned out that the communication between all subsystems is very stable. Because of the engagement of the MES the system could react on errors and the process had not to be canceled which is a big advantage compared to off-line programmed processes. Fig. 7 shows sequential images with planning results and related execution.


(a) Layup tooling on an alternate position.


(b) Ceiling mounted robots.

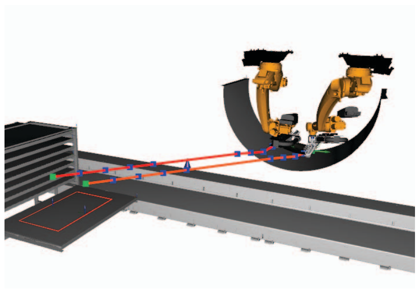

(c) Ceiling mounted and alternate position.

Fig. 6: Different scenarios tested in the CoCo framework. The planning has been performed without support points and machine learning linear axis position prediction.

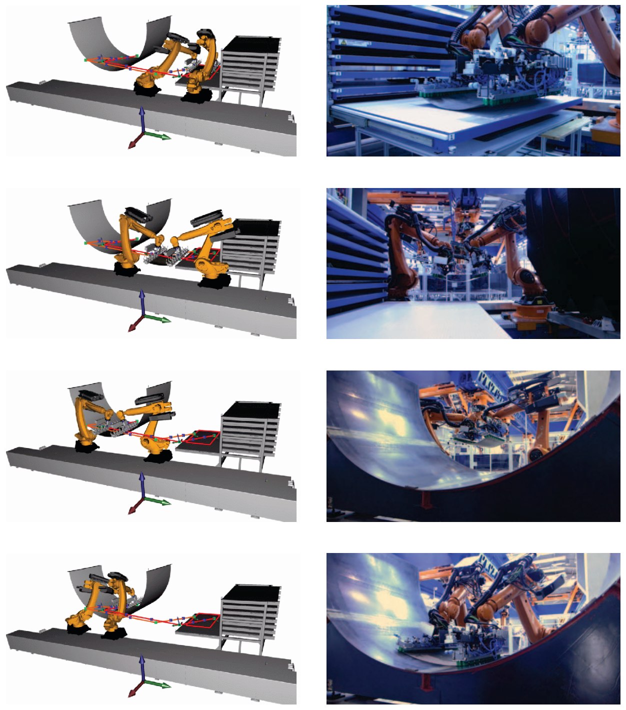


Fig. 7: Execution of a full automatic planned process for dry fiber placement of the lower half of an airplane fuselage. The left column shows the planning result and the right column shows the execution in the real industrial environment.

V. Conclusions

We showed a full automatic planning system for cooperating robots in an industrial scale. During pretests where we manually programmed the proposed scenario with KUKA RoboTeam and teach-in, we needed about one day per cut-piece. For the proposed A321 part we need to teach-in 200 cut-pieces which would need a long time. With our system it is possible to realize the process in a profitable way. At the moment we just tested the system with some test cut-pieces to validate the proper operation. In the next step we want to produce the whole fuselage component which was described in Sect. III. Additionally we are interested in the positioning accuracy of the complete system. In [25] we already determined the accuracy of the positioning of the cooperating robots to each other while moving around using a laser tracker. In the next step we want to measure the position of the cut-pieces after the placement in the mould. For the determination of the position of the cut-pieces we will use a 3D laser light section sensor to measure the contour, which was developed before.

The path planning approach can easily be transfered to different robots from various manufacturers. Beforehand the CAD data for collision detection and the denavit hartenberg parameters for inverse kinematics of the robot are needed. The Robotics API provides a robot programming interface which is agnostic of the concrete robot. The underlying RCC already supports industrial robots from different vendors (e.g. KUKA, Stäubli, Schunk) and can be extended to support further robots, without the need to adapt the robot programs.

We already successfully tested the planning system in different scenarios in the CoCo environment (see Fig. 6). The framework is also prepared to handle more than one slave robot. In the future we want to integrate the planner in a bigger robotic cell at our site called Multifunctional Cell (MFZ) at our site which has been introduced in [29]. This cell is $30\,\mathrm{m}$ long, $15\,\mathrm{m}$ wide and $7\,\mathrm{m}$ high. The facility offers maximum flexibility in production scenarios based on six ceiling mounted robots using the same workspace and an undisturbed shop floor.

At the moment the major bottleneck of the implementation is the collision detection mechanism which is currently running on a single blocking thread.

References

[1] H. Lasi, P. Fettke, H. G. Kemper, *et al.*, "Industry 4.0", *Business and Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[2] C. Smith, Y. Karayiannidis, L. Nalpantidis, *et al.*, "Dual arm manipulation - A survey", *Robotics and Autonomous Systems*, vol. 60, pp. 1340–1353, 2012.

[3] J. Cortés, "Motion Planning Algorithms for General Closed-Chain Mechanisms", PhD thesis, 2003.

[4] S. M. LaValle, J. H. Yakey, and L. E. Kavraki, “A Probabilistic Roadmap Approach for System with Closed Kinematic Chains”, 1999, pp. 6–11.

[5] S. M. LaValle, “Planning Algorithms”, 2006.

[6] V. Kumar and J. F. Gardner, “Kinematics of Redundantly Actuated Closed Chains”, *IEEE Transactions on Robotics and Automation*, vol. 6, no. 2, pp. 269–274, 1990.

[7] D. Uri and D. P. Davis, “Collision Free Path Planning of an Object Using Multiple Robot Manipulators”, 2014.

[8] S. S. Mirrazavi Salehian, N. Figueroa, and A. Billard, *Dynamical system-based motion planning for multi-arm systems: Reaching for moving objects*, In Proceedings of International Joint Conference on Artificial Intelligence 2017, Melbourne, Australia, 2017.

[9] S. Y. Nof, *Handbook Of Industrial Robotics*. John Wiley & Sons, Inc., 1999, p. 1378.

[10] Y. Gan, X. Dai, and J. Li, “Cooperative Path Planning and Constraints Analysis for Master-Slave Constraints Analysis for Master-Slave Industrial Robots Industrial Robots”, *International Journal of Advanced Robotic Systems*, vol. 9, p. 1, 2012.

[11] 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.

[12] C. S. Tzafestas, P. A. Prokopiou, and S. G. Tzafestas, “Path Planning and Control of a Cooperative Three-Robot System Manipulating Large Objects”, *Journal of Intelligent and Robotic Systems*, pp. 99–116, 1998.

[13] S. Lahouar, S. Zeghloul, and L. Romdhane, “Real-Time Path Planning for Multi-DoF Manipulators in Dynamic Environment”, *International Journal of Advanced Robotic Systems*, vol. 3, no. 2, p. 1, 2006.

[14] S. Pellegrinelli, N. Pedrocchi, L. M. Tosatti, *et al.*, “Validation of an extended approach to multi-robot cell design and motion planning”, *Procedia CIRP*, vol. 36, pp. 6–11, 2015.

[15] ——, “Multi-robot spot-welding cells for car-body assembly: Design and motion planning”, *Robotics and Computer-Integrated Manufacturing*, vol. 44, pp. 97–116, 2017.

[16] M. Zribi, M. Karkoub, and L. Huang, “Modelling and control of two robotic manipulators handling a constrained object”, *Applied Mathematical Modelling*, vol. 24, pp. 881–898, 2000.

[17] J. Gudiño-Lau and M. A. Arteaga, “Dynamic Model, Control and Simulation of Cooperative Robots: A Case Study”, in *Mobile Robotics, Moving Intelligence*, J. Buchli, Ed., 2006, ch. 13.

[18] M. Mediavilla, J. L. González, J. C. Fraile, *et al.*, “Reactive path planning for robotic arms with many degrees of freedom in dynamic environments”, *IFAC Proceedings Volumes*, vol. 15, no. 1, pp. 443–448, 2002.

[19] M. Eckardt, A. Buchheim, and T. Gerngross, “Investigation of an automated dry fiber preforming process for an aircraft fuselage demonstrator using collaborating robots”, *CEAS Aeronautical Journal*, vol. 7, no. 3, pp. 429–440, 2016.

[20] A. Angerer, A. Hoffmann, A. Schierl, *et al.*, “Robotics API: Object-Oriented Software Development for Industrial Robots”, *J. of Softw. Eng. for Robotics*, vol. 4, no. 1, pp. 1–22, 2013.

[21] M. Vistein, A. Angerer, A. Hoffmann, *et al.*, “Flexible and continuous execution of real-time critical robotic tasks”, *Intl. J. Mechatronics & Autom.*, vol. 4, no. 1, 2014.

[22] A. Schuster and M. Kühnel, “Automated preforming of curved thermoplastic organic sheets”, in *SAMPE*, Amiens, 2015.

[23] 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.

[24] L. Larsen, V.-L. Pham, J. Kim, *et al.*, “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.

[25] A. Angerer, A. Hoffmann, L. Larsen, *et al.*, “Planning and execution of collision-free multi-robot trajectories in industrial applications”, in *47th Symposium on Robotics - International Symposium on Robotics*, ISR, 2016.

[26] I. A. Sucan, M. Moll, and L. E. Kavraki, “The Open Motion Planning Library”, *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.

[27] M. Brandst, A. Angerer, and M. Hofbaur, “An Analytical Solution of the Inverse Kinematics Problem of Industrial Serial Manipulators with an Ortho-parallel Basis and a Spherical Wrist”, 2014.

[28] L. Larsen, J. Kim, M. Kupke, *et al.*, “Automatic Path Planning of Industrial Robots Comparing Sampling-Based and Computational Intelligence Methods”, in *27th International Conference on Flexible Automation and Intelligent Manufacturing*, Modena: Elsevier, 2017.

[29] F. Krebs, L. Larsen, G. Braun, *et al.*, “Design of a multifunctional cell for aerospace CFRP production”, in *Advances in Sustainable and Competitive Manufacturing Systems*, ser. LNME, Springer, 2013, pp. 515–524.