Software Toolchain for Offline-Programming a Jig-Less Fiber Placement Process Using Cooperating Robots

Michael Vistein[1][a], Lars Brandt[1][b], Gabriel Côté[2][c], Julien-Mathieu Audet[2][d] and Bruno Monsarrat[2][e]

[1]*German Aerospace Center, Center for Lightweight Production Technology, Am Technologiezentrum 4, Augsburg, Germany*

[2]*National Research Council Canada (NRC), Advanced Robotics and Mechatronic Systems (ARMS), Aerospace Manufacturing Technology Center (AMTC), 2107 Chem. de Polytechnique, Montréal, Québec, Canada*

{michael.vistein, lars.brandt}@dlr.de, {gabriel.cote, julien-mathieu.audet, bruno.monsarrat}@cnrc-nrc.gc.ca

Keywords: Offline Programming, Dual Robot, Thermoplastic Composites, Automated Fiber Placement.

Abstract: Automated Fiber Placement (AFP) is one technology that can be used to produce lightweight Carbon Fiber Reinforced Plastic (CFRP) aircraft parts which can help in the decarbonization of the aviation industry. Usually this process requires an expensive, rigid mold into which the material is laid using a tape laying head. By using a second industrial robot with a specialized counter-endeffector, the need for a mold can be avoided. However, in order to be able to efficiently program two industrial robot simultaneously, an end-to-end offline-programming (OLP) approach is needed. This paper demonstrates a software toolchain covering the whole process from initial computer aided design (CAD) to the final robot controller programs.

1 INTRODUCTION

Decarbonization of the commercial aviation industry is targeted to be completed in 2050. This requires new propulsion systems, improved aerodynamics and efficient lightweight structures. Significant reduction in the structural weight could be achieved for long-haul aircraft programs using Carbon Fiber Reinforced Plastics (CFRPs) in the past. In order to transfer the technology to high volume single-aisle aircrafts, new joining, consolidation and recycling processes have to be established. A particular focus is given to thermoplastic CFRPs, as they offer great potential for direction consolidation and new welding processes enable additional weight reductions. The first can be exploited using Automated Fiber Placement (AFP) where thermoplastic prepregs are deposited as narrow tapes layer by layer.

Traditional AFP processes use a rigid mold. This has a big disadvantage: Producing the mold is very cost intensive, and therefore changes to the product are difficult since they usually require a new (expensive) mold. In (Kochoski et al., 2022), an approach utilizing a second robot instead of a rigid mold was demonstrated. In this paper, a dual-robot AFP process is shown for CF/LMPAEK (Carbon-Fiber Low-Melt PolyArylEtherKetone) materials. The setup is quite similar to the one described by Kochoski et al., two KUKA 6-DoF industrial robots are used, and real-time synchronization is achieved using KUKA's technology package RoboTeam.

For complex processes such as AFP, offline-programming is a key requirement. Manually "teaching" hundreds or even thousands of tape tracks clearly is no viable option. For single robot applications, several (commercial) offline programming software applications exist. Many robot manufacturers provide specific tools for their hardware, and there are also vendor-independent sofware tools available such as DELMIA[1], Cenit FastSuite Ed2[2], Visual Components[3] or RoboDK[4].

While some of these tools allow for the simultaneous programming of two (or more) independent robots within a single cell, automatic programming of cooperating robots is often not possible. An overview of programming techniques for multi-robot applica-

[a] https://orcid.org/0000-0001-6998-0017
[b] https://orcid.org/0009-0007-8608-0759
[c] https://orcid.org/0009-0005-7842-908X
[d] https://orcid.org/0009-0007-4667-4595
[e] https://orcid.org/0009-0002-4369-2986

[1]https://www.3ds.com/products/delmia
[2]https://www.fastsuite.com
[3]https://www.visualcomponents.com/
[4]https://www.robodk.com

Vistein, M., Brandt, L., Côté, G., Audet, J. and Monsarrat, B.
Software Toolchain for Offline-Programming a Jig-Less Fiber Placement Process Using Cooperating Robots.
DOI: 10.5220/0012889700003822

In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024) - Volume 2*, pages 379-386
ISBN: 978-989-758-717-7; ISSN: 2184-2809

tions is provided by (Gan et al., 2013). There are some approaches that try to facilitate the programming of a master/slave robot system with one robot holding the workpiece, while the other performs the manipulating task (e.g. (Wagner et al., 2014; Xiong et al., 2020). In these approaches, an automated trajectory generation based on the trajectory required on the workpiece is done.

Another kind of synchronization is required when two robots share the load of a workpiece, e.g. for transportation purposes. Rigid structures can be transported by all robots performing the same trajectory, either by using the built-in features of the robot controllers, or by planning identical motions (e.g. (Vistein et al., 2019)). For flexible pieces such as textiles, the deformation of the piece due to gravity has to be taken into account (e.g. (Larsen et al., 2015; Larsen et al., 2017)).

The dual robot AFP process is special with regard to some aspects. First, there are commercial tools available for motion planning for a single laying machine or robot. Second, while the secondary robot needs to move on a very similar trajectory as the primary robot, the strategies used for load sharing cannot be applied due to small differences in the trajectory for the thickness of the part or the required (small) fixture. Therefore, an approach which builds upon already available tools, extending them at the necessary points, has been chosen. The work done by (Kochoski et al., 2022) seems to follow a very similar path by extending their own MikroPlace software, although little information about the specific implementation is available.

The remainder of this paper is organized as follows: In section 2, the setup of the dual-robot AFP process is described. Section 3 describes the toolchain that is being used for offline-programming, and in section 4 the necessary steps for robot synchronization and robot code generation are explained. Finally in section 5 a conclusion is drawn and an outlook for future extensions is provided.

2 JIG-LESS DUAL-ROBOT AUTOMATED FIBER PLACEMENT

For the AFP processes, a laser-assisted thermoplastic layup machine provided by AFPT GmbH is used. The laser has a rated power output of 6 kW and is supplied by Laserline GmbH. The layup machine is designed to deposit 3 × 1/2” prepreg tapes per track. During layup the tape and laminate are heated over melting temperature equally. In this case a conformable consolidation roller (refer to figure 1) is used that forms a consolidation area under pressure. Due to this compaction force intimate contact between laminate and tape is established and polymer self-diffusion dissolves the interface between the materials. Water-cooling of the compaction roller ensures that the bonding partners are below their melting temperature after the roller even for elevated layup velocities(Brandt et al., 2023).

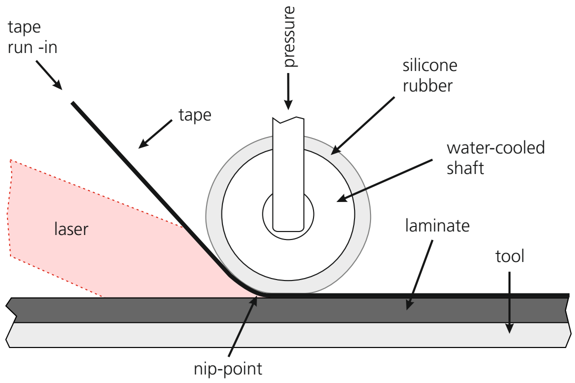

Figure 1: Process principle AFP.

To achieve a jig-less (or – at least – low-jig) solution, two additional components are required:

- A mobile counterpart that replaces the traditional full-size jig.
- A small fixture that can be used to attach the created piece to.

In this case study, a second robot, equipped with a special counter-endeffector is used as counterpart. This endeffector consists of an aluminum roller that resembles the compaction roller, except not being conformable. The roller is driven by a servomotor which is synchronized to the Tool Center Point (TCP) velocity (in rolling direction) of the second robot, i.e. the roller surface pointing to the primary robot does not move with respect to the world, therefore providing a surface much similar to a mold. Synchronization of roller and robot velocity is done using the Robot Sensor Interface (RSI) technology package providing the robot’s current position to the Beckhoff TwinCAT PLC driving the servo motor.

Since the part cannot be laid purely into thin air, a small, lightweight fixture to attach each tape at the beginning and the end is required. The fixture consists of aluminum profiles for the overall frame and numerous (identical) aluminum modules that form the surface where the beginning and end of each tape is placed. The modules are beveled on one side with the straight side pointing to the AFP head. The setup can be seen in figure 2.

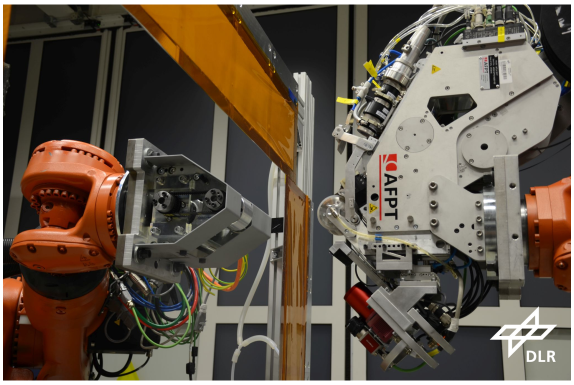

Figure 2: Setup of dual-robot AFP process.

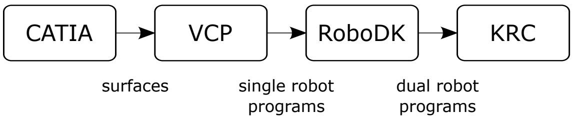

Figure 3: Schematic flow of software toolchain for dual-robot AFP applcation with the main type of data that is transmitted between the steps.

3 OFFLINE PROGRAMMING TOOLCHAIN

Since AFP requires a large number of individual placement trajectories, manual robot programming ("teaching") is no viable option. Starting with the CAD design of the desired component, an automated path for generating the final robot programs is desired. The overall flow is depicted in figure 3.

3.1 CATIA

For high performance CFRP components, a load-compliant design is generated by adjusting the laminate thickness and fiber orientation accordingly using computer aided design (CAD). In this work Dassault CATIA V5 is utilized. The design process starts with the definition of the part surface. All ply boundaries are defined with reference to this surface. Most CAD systems do not implement process driven design changes, thus an export to a computer aided manufacturing (CAM) software is sensible.

3.2 VCP

Using the surface and ply boundaries, the laminate is constructed sequentially. Each ply is first segmented into tracks comprising the width of all tapes placed simultaneously. An optimization is carried out to maximized coverage of the ply and minimize gaps and overlaps between the tapes. Afterwards the layup sequence and retraction movements of the robot are defined.

Vericut Composites Programming (VCP) generates layup tracks and usually creates Numerical control (NC) programs for automated layup machines. By using custom-made post-processors, it is possible to create programs for other types of machines, too. In the dual-robot use case, the NC programs are transformed into KUKA Robot Language (KRL) source files which can be executed by the KUKA KRC4 controller. These source files contain the trajectory, as well as all necessary I/O operations needed to control the AFP head (e.g. cutting the tapes at the appropriate locations, turning the laser source on and off).

Since the work-pieces can consist of free-form surfaces, an approximation of the robot programs to the ideal trajectories is necessary. A common practice of offline-programming software is to segment the trajectories into small pieces that can be connected with either linear or circular motion blocks that are blended into each other, or by using spline motions. The VCP post-processor uses spline motions for approximation. The layup trajectory is split into small auxiliary positions (with configurable maximum distances and orientation changes) which are inserted into the source program using the `SPL` motion command and embedded into a single `SPLINE` motion block. The KUKA controller uses these commands to plan a trajectory which exactly passes every auxiliary position and maintains a constant velocity as far as possible.

3.3 RoboDK

The KUKA programs that are generated by VCP can be executed directly on a single robot for a traditional fiber placement system using a rigid mold. For a dual-robot application however, the corresponding programs for the secondary robot still need to be generated. Several important requirements must be taken into account:

- VCP takes the thickness of the already placed layers into account while the counter tool must be at the position of the first layer during each run.
- The trajectory of the secondary robot must be adjusted for the surrounding frame that holds the tapes which has a defined thickness.
- During the layup motion, both robots need to be synchronized, i.e. each robot must be at the designated position at the right time.
- The approach and retract motions must be adjusted for the secondary robot.

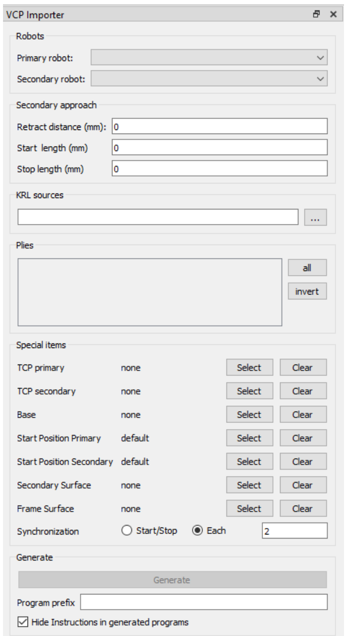

Figure 4: Graphical user interface of VCPImporter plugin in RoboDK.

- The (initial) park-position of the secondary robot needs to be defined (in joint coordinates or alternatively in Cartesian coordinates with additional redundancy information such as status/turn).

In order to create the trajectories for the secondary robot while fulfilling all requirements, a software solution based on the offline-programming tool RoboDK has been created. RoboDK allows to model a robotic work-cell and to create programs graphically. The created programs are independent of a concrete manufacturer or robot type that is used, but can later be converted into manufacturer-specific program code by adjustable post-processors. A 3D simulation of the robot-cell and of the programs is possible. The simulation takes specifics of the robots (such as singularities) into account and can optionally perform a collision check based on CAD geometries. This allows to preview and validate all trajectories prior to running on the real robots.

RoboDK offers an Application Programming Interface (API) for custom extension. Besides API implementations for Python, C# and C++ there is also a plugin-interface (based on C++ and Qt) which allows to run custom code directly within the RoboDK process. For the dual-robot AFP use-case, the latter interface has been selected. The number of auxiliary points can become very large, therefore a fast connection between the custom software and RoboDK is necessary. The plugin-interface is faster by a magnitude compared to the (network-based) API interface.

The created plugin (called VCPImporter) provides a graphical user interface that is embedded in RoboDK (see figure 4). Because the programs generated by VCP are intended for a single machine, some additional information has to be provided by the user to allow the generation of a secondary trajectory. These information contain:

- Retract distance: The distance the primary robot retracts after finishing a track in order to move on a connection link to the next track.
- Start- and stop-length: The distances the tape laying head is moving on the frame at the start and stop of each track.
- TCPs, Base: The TCP and Base (defined within RoboDK) that should be used for primary and secondary robot.
- Start positions: The park-positions defined in RoboDK at which the primary and secondary robot start each ply.
- Surfaces: The CAD objects in RoboDK which represent the frame and the final workpiece.

The VCPImporter plugin parses the KRL source files generated by VCP. A program consisting of all motions specified by VCP is created within RoboDK, which immediately allows for a simulation of the trajectory of the primary robot.

The VCP post processor uses a defined structure for each generated program. Every track is placed in its own source file, and every source file consists of the following blocks:

- Program header
- Optional: point-to-point (PTP) motion to start position (first track of each ply)
- Optional: Motion commands for transfer from last track to current track (connection link)
- Motion commands for the track, consisting of a number of `SPL` motion commands embedded in to a `SPLINE` block.
- Program footer

Each segment is clearly embedded in a `FOLD` structure (a source code hint that usually allows to show or hide certain blocks within the program on the limited screen of the programming pendant). This predictable structure allows the VCPImporter plugin to split each trajectory into the relevant segments, and to calculate appropriate trajectories for the secondary robot.

Another important feature of RoboDK is the possibility for geometric calibration of the used robots. While industrial robots usually offer a very good repeatability, the absolute accuracy is often worse.

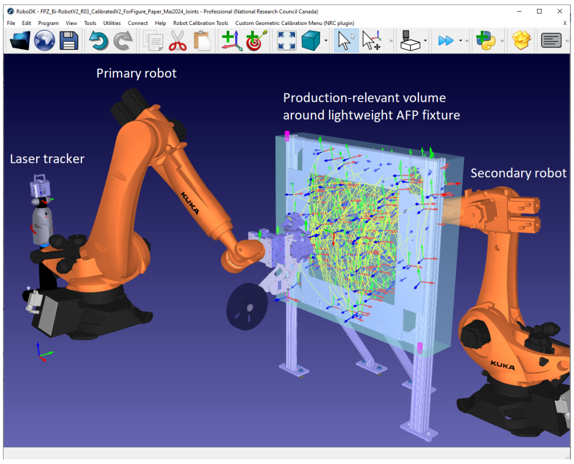

Figure 5: RoboDk interface showing the 100 test poses that were randomly generated in the production-relevant parallepipedic volume located around the lightweight fixture to verify the accuracy of the primary robot following geometric calibration. After each test pose is reached, the laser tracker measures a metrology marker located at the vicinity of the robot tool-center-point. A similar approach was used to verify the accuracy of the secondary robot.

RoboDK allows to calibrate the robots using a laser tracker as an absolute reference. A model of the deviations of the robot within its workspace is generated, and programs can be adjusted automatically with small offsets to the desired Cartesian positions such that the robot reaches the target more precisely. As an illustration of the importance of geometric path correction in the off-line programming flow, the absolute accuracy of each robot has been evaluated before and after geometric calibration using a set of 100 test poses that were randomly generated in a production-relevant volume of rectangular parallelepiped shape located around the lightweight AFP fixture, as shown in figure 5. The accuracy improvements obtained for both the primary and secondary robots are described in table 1.

By embedding the VCPImporter into RoboDK directly and by creating the programs within RoboDK, the geometric calibration feature can be used without additional effort.

3.4 Secondary Trajectory Generation

For each of the program segments described in section 3.3, a different strategy for the generation of the secondary trajectory is applied.

Program header and footer can be ignored, since they will be regenerated appropriately by the post-processor that is exporting the programs from RoboDK (cf. section 4). The PTP motion to the start position is simply replaced by a motion to the start

Table 1: Summary of the accuracy improvements obtained thanks to the geometric calibration of the primary and secondary robots. For each robot, the accuracy was verified by measuring a set of 100 test poses using a laser tracker, as illustrated in Fig. 5.

	Mean	Max	St. Dev.
Model-based geometric calibration results - Primary robot			
Norm of absolute positional error before calibration (mm)	0.865	1.201	0.182
After calibration (mm)	0.086	0.181	0.035
Error reduction	90.06 %	84.93 %	80.77 %
Model-based geometric calibration results - Secondary robot			
Norm of absolute positional error before calibration (mm)	1.590	1.938	0.175
After calibration (mm)	0.094	0.222	0.046
Error reduction	94,09 %	88,54 %	73,71 %

position specified by the user for each robot. Because this is the first motion of the final program, this will always be a PTP motion.

The connection links that return the tool from the end of a track to the beginning of a new track are mirrored. The approach and retract motions generated by VCP consist of a single linear motion directly in the Z-direction of the tool. All other intermediate points of the connection link also have the same distance (in Z-direction) from the workpiece. Therefore, the secondary trajectory is calculated by moving each frame of the primary trajectory for twice the retract distance along the Z-direction. The result is a trajectory mirrored along the work-piece. A diagram of the resulting trajectories can be seen in figure 6.

The layup motion itself needs to be handled differently. By using the overall track length and the length that has been laid up to a certain point, together with the user-specified start- and stop-lengths, it can be determined whether the current track point is laid onto the frame or in the void in the middle. If the track point is on the frame, it is projected along the Z-axis to the backside of the frame. If it is in the air, a projection to the closest point on the workpiece surface is performed. In order to achieve good results, it is necessary that a precise CAD model of the workpiece's lower surface – but without thickness – is provided.

Using either the projection to the frame or the workpiece surface, it is ensured that the secondary trajectory both takes the thickness of the frame as well as the thickness of previously laid tracks into account.

From a 2D-perspective (like in figure 6) it seems that the secondary trajectory (on the right) is identical for every layer since the part only grows to the left. In order to achieve optimal part quality however, the tracks of each layer are usually not identical to

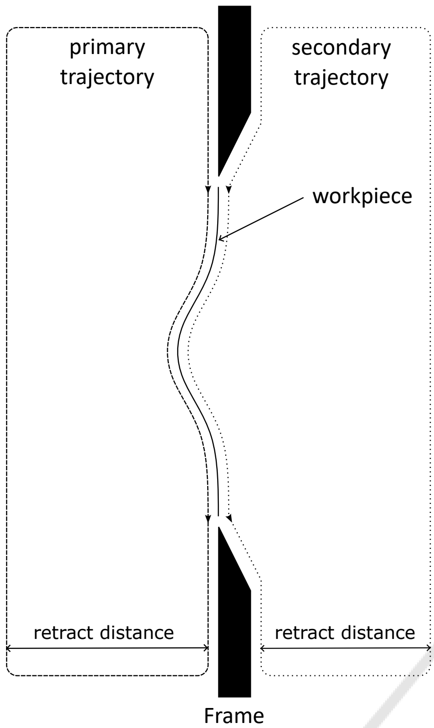

Figure 6: Schematic side display of the trajectories of the primary robot (dashed) and the secondary robot (dotted) and the frame that is used for support.

the previous one. Depending on the part design, they might be shifted sideways in a way the border of two tapes of the previous layer is covered with a tape in the next layer, or the fiber angle might even be completely changed e.g. for 0°, ±45° and 90° layers. Therefore, the secondary trajectory has to be calculated independently for each track.

4 ROBOT PROGRAMS

With both the primary and secondary trajectories modeled within RoboDK, the final programs that are run on the KUKA KRC4 controllers can be generated. For both tools, the TCPs are defined identically, i.e. for the first layer both TCPs usually are at the same coordinates during layup. For the tape laying head, the Z-axis points normal into the workpiece, while for the counter endeffector the Z-axis points out of the work-piece.

4.1 Synchronization

To achieve good part quality, synchronization between the tape laying head and the counter effector is required. Synchronization between the robots is achieved using the KUKA.RoboTeam technology package. This package establishes a real-time communication channel between the robots and offers several possibilities for synchronization:

Program Synchronization. A certain point in each robots program is reached simultaneously, i.e. one robot waits until the other has reached the specified point.

Motion Synchronization. One or more motions are run simultaneously, i.e. both robots start and finish their respective motion at the same time. The velocities and accelerations of the motions are adjusted as needed.

Geometric Linking. One robot is geometrically linked to the other, i.e. it movements are relative to the TCP of the first robot. Without any specific motions of the secondary robot, it performs exactly the same motion as the first.

For the Dual-Robot AFP process, program and motion synchronization are used. Only the track itself and parts of the approach and retract motions are synchronized, the connection links are independent of each other.

The first synchronization point is at the approach motion. The primary robot may only start approaching the frame once the secondary robot has completed the approach motion and is in contact with the frame. This is necessary, because otherwise the tape laying head would apply the process force to the unsupported frame and potentially cause deformation. With the counter endeffector already in place, this force is absorbed by the secondary robot (as it is during the following layup).

The layup track itself consists of a large number of `SPL` motions that are embedded in a single `SPLINE` block. The KUKA controller calculates a single motion of this whole block. For synchronization purposes, it is possible to synchronize the whole block, as well as to specify certain points within the block that must be reached simultaneously. In the dual robot AFP use case, the number of intermediary points in the spline block are identical, so it would be possible to synchronize every single point. This however causes unnecessary load on the robot controller, and synchronizing every third point turned out to be sufficient for the tested parts. Nevertheless, this setting can be modified by the user.

All required synchronization points are inserted into the RoboDK program. Using the Python interface of RoboDK, a rudimentary simulation of the synchronized motions can be performed. Unfortunately, this is very limited and only roughly the equivalent of program synchronization as mentioned above, which leads to both robot stopping at each synchronization point.

4.2 Source Code Generation

RoboDK already provides means to generate robot controller files from the programs. Like VCP, a post-processor is used, which is written in Python and can be adjusted to specific needs. RoboDK already provides an exemplary open-source post-processor to generate KRL programs which has been modified according to the special requirements of the dual-robot application.

In the dual robot AFP use case, the fixed structures of the program can be utilized once again. The overall program structure is provided as template separately for the primary and secondary robot. The sections of the template are filled with the motions of the RoboDK programs, and additionally synchronization markers are inserted where indicated. As template engine, Jinja2[5] is used.

The generated source files make use of TCP and base definitions on the robot controllers, which must be created appropriately. Both robots must share a common world coordinate system and must both define a base coordinate system on the frame. Once these preconditions are met, the generated programs can be uploaded to the robot controller and executed.

5 CONCLUSION

The process for dual-robot AFP is currently still under heavy development. At the current time, it is not yet possible to manufacture a part which is ready to go into production. An example of a double-curved test specimen that has been produced is depicted in figure 7. The previously described toolchain is a key factor for successful process development, because it allows for rapid testing of process parameters and part designs. After changes have been made in the part design for example, only the steps necessary for traditional AFP CAM (i.e. trajectory planning) are required, while all specifics for the dual-robot application are generated automatically.

While the current software toolchain aimed at both robots being synchronized as precisely as possible, a next step in the development will be the possibility for a defined offset between the primary and secondary robot. The aim is to induce prestressing into the material (dependent on the curvature of the part) which helps to reach the desired form better.

[5]https://palletsprojects.com/p/jinja/

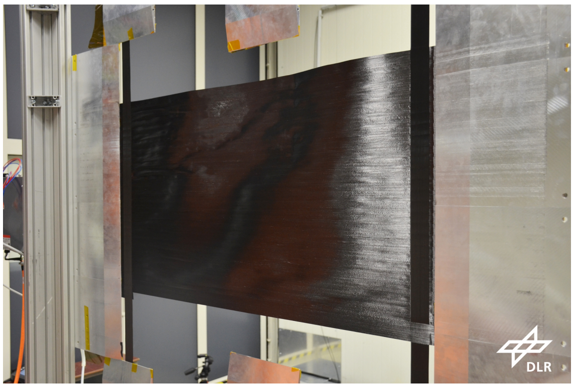

Figure 7: Test specimen produced by dual-robot AFP process.

In addition, while the RoboDK model-based geometrical calibration already improves the overall absolute accuracy of the robots significantly (ref. table 1), it has be shown in a recent complementary research that the performance of the geometric calibration can be further improved by combining the model-based geometric calibration of the primary and secondary robots with a cascaded calibration approach using deep neural networks (Maghami et al., 2023). In the later research, the mean value of the tracking error between the primary and secondary robots, as well as the absolute positional accuracy of the secondary robot could be improved by an additional 57%. Moreover, external forces that are applied to the robot still can cause significant deviations. With dual-robot AFP, both robots are applying the process force of around 600 N against each other and therefore suffer elastic deformation. In a joint research project, the in-situ elastic calibrations and compensation as introduced in (Monsarrat et al., 2024) has been applied to the dual-robot AFP use-case, and in improvement in absolute accuracy for both the primary as well as the secondary robot could be shown.

ACKNOWLEDGMENTS

The authors wish to acknowledge the financial support of the National Program Office (NPO) of the National Research Council Canada (NRC).

REFERENCES

Brandt, L., Vistein, M., Deden, D., Schühler, F., and Gänswürger, P. (2023). Towards toolless manufacturing of aerospace CFRP components via thermoplastic AFP. In *Aerospace Europe Conference 2023*.

Gan, Y., Dai, X., and Li, D. (2013). Off-line programming techniques for multirobot cooperation system. *International Journal of Advanced Robotic Systems*, 10(7):282.

Kochoski, F., Dukovski, V., Samak, S., Cvetkoska, D., and Petkoska, B. (2022). Manufacturing Thermoplastic Composites by Laser Automatic Tape Placement Toolless Technology with Dual Robot System. *International Journal of Engineering and Technical Research*, 11:238.

Larsen, L., Kaspar, M., Schuster, A., Vistein, M., Kim, J., and Kupke, M. (2017). Full automatic path planning of cooperating robots in industrial applications. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pages 523–530.

Larsen, L., Pham, V.-L., Kim, J., and Kupke, M. (2015). Collision-free path planning of industrial cooperating robots for aircraft fuselage production. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2042–2047.

Maghami, A., Imbert, A., Côté, G., Monsarrat, B., Birglen, L., and Khoshdarregi, M. (2023). Calibration of multi-robot cooperative systems using deep neural networks. *Journal of Intelligent and Robotic Systems*, 107(4):55.

Monsarrat, B., Audet, J.-M., Fortin, Y., Côté, G., Vistein, M., Brandt, L., Sadek, A., and Krebs, F. (2024). Insitu elastic calibration of robots: Minimally-invasive technology, cover-based pose search and aerospace case studies. *Robotics and Computer-Integrated Manufacturing*, 89:102743.

Vistein, M., Deden, D., Glück, R., and Schneyer, S. (2019). Automated production of large fibre metal laminate aircraft structure parts. *Procedia Manufacturing*, 38:1300–1307. 29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM 2019), June 24-28, 2019, Limerick, Ireland, Beyond Industry 4.0: Industrial Advances, Engineering Education and Intelligent Manufacturing.

Wagner, M., Hess, P., and Reitelshoefer, S. (2014). Automated programming of cooperating industrial robots. In *ISR/Robotik 2014; 41st International Symposium on Robotics*, pages 1–8.

Xiong, J., Fu, Z., Chen, H., Pan, J., Gao, X., and Chen, X. (2020). Simulation and trajectory generation of dual-robot collaborative welding for intersecting pipes. *The International Journal of Advanced Manufacturing Technology*, 111:2231–2241.