

Title:

Design Methodologies for Automated Placement of 3D Complex Geometries

Authors:

Joshua Halbritter, halbritj@email.sc.edu, McNair Center, University of South Carolina

Ramy Harik, harik@mailbox.sc.edu, McNair Center, University of South Carolina

Alex Zuloaga, zuloaga@email.sc.edu, McNair Center, University of South Carolina

Michel van Tooren, vantooren@sc.edu, McNair Center, University of South Carolina

Keywords:

Automated placement, complex geometries, manufacturing.

DOI: 10.14733/cadconfP.2016.332-336

Introduction:

The term fiber placement refers to composite fabrication processes that involve laying down reinforcing fibers along predefined trajectories onto a component [1]. A number of different automated processes have elements that could be considered fiber placement, such as filament winding, pultrusion, and preforming. However, these are not considered here due to their limited range in adaptability to complex component surfaces. This paper will focus on highly adaptable automated fiber placement processes (AFP) which will have the ability to render varying geometrical surface complexities automatically and then be able to effectively execute the laying of bundles of fibers or tapes, at a range of orientations, using a variety of toolings, upon a variety of complex geometries. In exploring an AFP manufacturing option which will take increasingly complex geometries from rendering to real world execution, several key components to achieve this application must be accounted for in both the planning and designing of the AFP methodology. In regards to planning, computational and simulation needs will need to be assessed before designing the experimental plan, which will be the basis of determining this AFP's application feasibility. This paper explores the first findings and analysis on effects of part geometry. We present multiple tools developed to understand the manufacturing of part. The demonstration cell was built with the perspective of a future robotics platform that would provide extended degrees of freedom to overcome path and geometry complexities. We present a java slicer developed specifically for this end that outputs RSL code, needed for the functionality of the demonstration cells. The paper ends with a discussion on manufacturing criteria indexes that are selected as well as proposed future steps for experimental verification.

Manufacturing failure criterion:

A failure criteria is a relation between a structural state, e.g. a stress state, and material properties to predict the onset of failure. A comprehensive laminate failure criterion would account for both first ply failure and interply delamination. The so-called LaRC04 [5] criterion is such a three-dimensional criteria. It accounts for the following failure modes. Delamination, matrix compression failure, fiber compression failure, matrix tensile failure, fiber tensile failure. Lopes [2] simplifies this approach by assuming only first-ply failure and neglecting out-of-plane components of the stress tensor. Design for manufacturing is an approach that deals with the manufacturing consequences of the design early within the design phase. This bounds the design space. For instance, AFP machines normally impose a limit on the curvature of the fiber orientations of a composite laminate. Thus, a part should be designed accounting for both functional and fabricational requirements. Van Tooren et al. [3] performed such a design constraining it to a specified minimum turn radius. The phenomenon behind

this constraint is the local out-of-plane buckling of the inner-tow fibers under compression. This minimum turning radius depends on the course and tow width. Blom [4] shows a table summary of minimum radii for different tow width. However, there are more manufacturing considerations to account for. Lopes [2] lists as important, besides the curvature constraint, fiber angle deviations where it is explained that the fiber path within a given tow is different for the centerline fiber than for the fibers at the edges. Marouene et al. [6] performed an experimental study of buckling behavior for variable stiffness laminates.

Methodology:

The following is the methodology used to make the carbon fiber placement machine follow a uniquely curved object. In order to do this an STL file of the object must be transformed into an RSL file in order for the machine to understand the code.

- Input: An STL file models the unique part that needs to have carbon fiber placed on it. The STL file will be loaded into java in order to work with it.
- Creating points along surface: Surface vectors and triangular planes come with the STL file. We use equidistant plane slices to intersect the triangular plane edges. This is where the path generating points are.
- Generating Path Vectors: The code will provide a path vector at each of the surface points that the slices create. The average of 2-3 of these vectors creates the path the carbon fiber machine will follow.
- Transition to RSL file: The path vectors transition into points the Kuka mounted fiber placement tool can follow. The RSL file will also allow for the object to move to start a new line of carbon fiber placement. A robot simulation program will be used to test the viability of the code.

Model Analysis

Code has been developed to read the model into memory, then perform slicing on the model according to user input. The slice contains information about the model's outer surface, particularly the vector position and the normal vector to the surface at each point along the slice. The user input includes the number of slices across the model, as well as the direction these slices should follow. A plane is iterated through the model according to the direction specified. Each facet that makes up the mesh contains three vectors which define the 3 points of the triangle, as well as a vector that defines the outer direction of the facet. In order to generate the slices, the code searches for the facets which intersect the plane, and a line is found across this intersection. The total number of set lines found at this intersection defines the cross section of the model. The portion of slice on the upper surface is then used for the generation of the path. After the initial generation of the slice, which contains the direction vector (Z-axis) orthogonal to the surface at the vector location of the point (figure 1a). Now the X and Y axes must be generated for the points. The X-axis is defined by averaging the directions from the surrounding points to the point of interest and normalizing the length of the vector. The X-axis represents the gradient on the surface at this vector, and points towards the next vector along the path. The Y-axis, simply generates from the cross product of X,Z axis, and it is orthogonal to both (figure 1b).

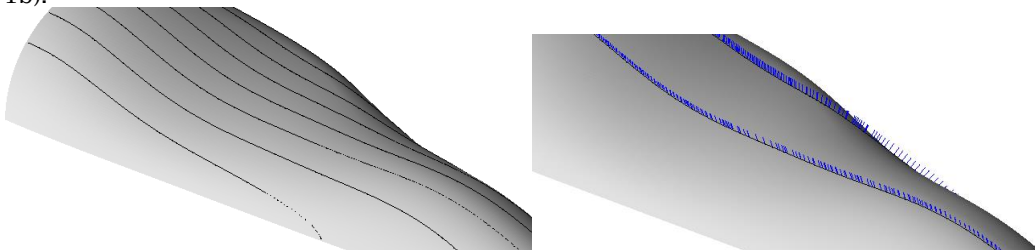


Fig. 1: From left to right: (a) The black lines indicate the path along the upper surface of the model. Portion of the part is shown for proprietary reasons. (b) The blue lines indicate the Z-axis that is normal to the surface of the model.

Generating the Robot Scripting Language (RSL)

The information required to generate the robotic motion instructions along the surface of the object have been provided to this RSL generation module from the path generation portion of the code. To define a motion point for the robot, the vector location of the point and the XYZ axes must be defined. The XYZ axes define the orientation for the robot's tool at the point along the path. The RSL generation module contains methods which simplify the generation of RSL. The methods use the necessary inputs, primarily vector location and directional axes, and reformats it with the RSL markup syntax, in order to generate the final functional code. Additional instructions are automatically included within the RSL file, which extends to Boolean inputs and outputs. The additional instructions serve to create a more realistic simulation, adding functionality beyond blindly following a provided path. They account for motion routines between individual sections of the paths, as well as changing the configuration of the robot and manipulating the available binary I/O ports. The binary ports can be used to communicate with other machines within the workcell, as well as enable tracing of tool's motions for visual inspections of its movements. The types of motions available to control the robot are linear and point to point. The linear motion directly translates the tool data, which is the point defined at the end of the tool, as well as the tools orientation, along the path between the current position and the specified coordinate. The point to point method involves internal optimization by the robot to find a path to the newly specified point, such that the path used may be unpredictable and result in undesirable movements. A limitation of the RSL is that the position along the rail, also known as the external axis, is not internally computed by the robot's motion controller. Therefore, the code developed for converting the path locations into motion points for the robot needs to compute the optimal travel distance along the external axis in order to complete the instructed motion. The base of the robot may be aligned with the end effector, or the base may be staggered from the head up to the length of the arm itself. The variation in this staggering along the external axis allows the robot to reach extreme locations on the surface, which may be very close to the rail or distant points. In general, the variation along the external axis significantly expands the working envelope available to the robot. More distant points may also be brought into the working envelope of the robot by manipulation of the surface's orientation with regards to the rail. The available manipulation was the simulated effect of a mandrel for the testing. A connection between the robot and the part was created, and the robots code could send a signal which would initiate a partial rotation of the surface. Rotation of the particular part was performed about its long axis, and the rotation could be stepwise or continuous depending on the pathing requirements.

Work Cell Organization

Several work cell organizations (figures 2a, 2b and 2c) were prepared to account for the limitations between each object's physical boundaries and the optimal degrees of freedom between the robot and the given model. Limitations for the robot include both its movement path and its reach about the model's surface. Movement path limitations include the robot's general restriction to a finite linear track and arm movements which do not collide with the track, the model, itself, or go beyond the designed tolerable axes of motion for the robot. For reach limitations, it was vital to account for the fiber placement processes needed to be applied at an angle which is normal to the surface of the object. To address this specific reach requirement, several simulated setups were created in KUKA sim pro, in which inner and outer boundary tests were systematically performed. First, it was determined that the initial orientation of the object relative to the robot would need to run parallel to the robot's track to utilize the robot's linear range. Second, the model was assessed statically and dynamically (rotating about the axis parallel to the track), at varying angles, degrees of rotations, and general proximity from the robot to determine the most cost effective and feasible work cell organization for our client. Each workcell was populated by a KR-120P Kuka Robotic arm, which rode upon a KL-10000, a 10 meter rail. A simple placeholder tool was constructed and mounted onto the KR's end effector to simulate the use of a fiber placement type device. Motion instructions for pathing along the surface were generated using an approximately 10 meter long, conical shape, with a roughly 3 meter radius. The generation of the path is restricted to the outer surface of the model and also incorporates predetermined aspects of the simulation, including work cell configuration and the parameters specific to the robot's orientations and limits.

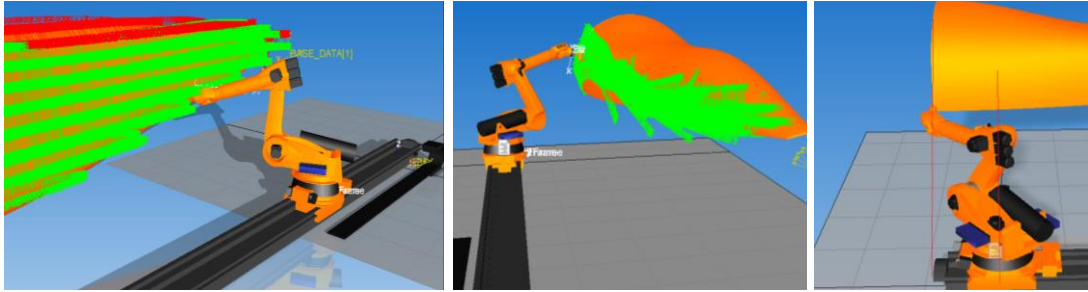


Fig. 2: From left to right: (a) Demonstration of rotating mandrel for horizontal pathing. (b) Example of actual motion pathing performed by robot for horizontal pathing while the part rotates on a mandrel. (c) Example of the staggering between the head and the base.

The top down configuration (figure 3a) relied heavily on the variation of the external axis in order to avoid crashing against the rail, as well as manipulating the robot's working envelope to reach more points which lie further away from the rail, necessitating longer reach from the robot (figure 3b). The primary use of the mandrel was during zero degree pathing along the long axis of the part. Stepwise rotation was used to bring each path into the robot's reach. The rotations would be signaled after the current path had been finished, and the robot removes itself from the surface. After the completion of the parts rotation, the robot would touch down to the surface and continue following the paths. Rotational control of the mandrel was also attempted for non-zero angles, however, such a configuration is not suited for stepwise rotations. Instead, the mandrel would be set to a constant rotation rate, and the head would move accordingly for the path, all the while maintaining a constant height. Such a configuration would ensure full coverage of the surface, however, a great deal of coordination between the speed of the robot and the mandrel rotation speed would be necessary to ensure continuous contact throughout the entire path, as well as ensuring a constant speed between the tool data and the surface (similar to actual AFP processes). Brief ventures into such a configuration encountered problems with timing between speed of path coverage and the mandrel's rotation. As a result, the rotation of the mandrel would outpace the speed of the tool, such that all coordination would be lost.

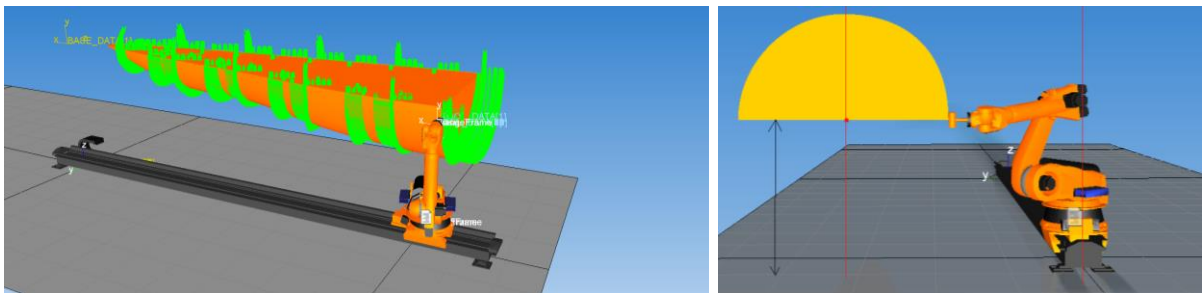


Fig. 3: From left to right: (a) the top down configuration (b) Typical dimensions for the offset configuration, the part location relative to robot base: the long axis of model is 3200mm from the rail. The axis of rotation for the model is 2000mm above the floor.

Path Evaluation

The two pre-requisites for path evaluation, are the generation of the robotic instructions as well as the modeling and configuration of the workcell that the robot interacts with. The tool used for path evaluation was Kuka Sim Pro 2.2. The robotic instructions generated by the initial code are transferred to Kuka Sim Pro 2.2. The Kuka software provides many useful tools for evaluating and improving programs. In terms of evaluation, Kuka's software simulates the robot's motor controllers, such that it can internally determine whether or not specific motion points are within the robot's reach. The software also allows for visual inspection of the motion pathing, as the robot's motions are simulated

within the software, such that anomalies may be noted and used to improve the code for generating the robotic instructions. The particular anomalies encountered throughout testing varied from flipping of the robot's end effectors as joints hit their rotational limits, in addition to the "crashing" of the arm into different components within the workcell. The general accessibility of the path by the robot and the presence of the physical anomalies are used to evaluate the overall viability of the path generated.

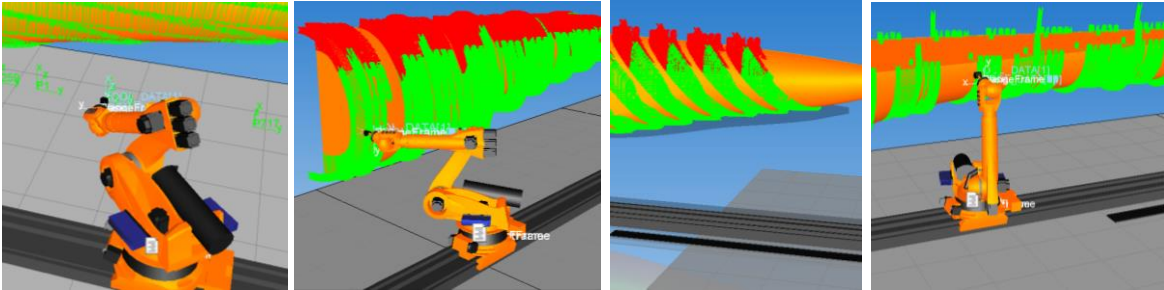


Fig. 4: From left to right: (a) Demonstration of the safety points used to prevent collisions between the robot and the work cell. (b) Demonstration of 0° and -45° pathing along the surface. Green areas indicate the possible motion points, while red areas are nonviable. These extreme points require physical manipulation of the work cell configuration to bring them into reach of the tool. The most likely solution would be the use of a mandrel to rotate the part for better accessibility. (c) Positioning the part above the rail enables complete coverage of the surface by the tool. The configuration's success results from the reduction of extreme points, either too near or distant.

Concluding remarks:

This paper allowed initial investigation and understanding of complex part geometry that will eventually be needed in special manufacturing. Whilst we attempted a robotics solution, the main target was to understand the underlying part complex geometry, and how we can, at one point, manufacture it as a single part with fiber placement. Moreover, the study allowed us to rationalize which parameters to optimize for the scenario of variable stiffness composites. The next step is to develop a design methodology that would quantify limitations of surface-to-w coverage combinations to support an optimization framework, in the context of the selected manufacturing parameters.

Acknowledgement:

The authors would like to thank Fokker for their financial support.

References:

- [1] Bryuneel, M.; A Modified Fast Marching Method for Defining Fiber Placement Trajectories Over Meshes, *Computers and Structures*, 125(0), 2013, 45-52.
- [2] Lopes, C. S.; *Damage and Failure of Non-Conventional Composite Laminates*, PhD dissertation, TU Delft, Faculteit Luchtvaart- en Ruimtevaarttechniek, 2009.
- [3] Tooren, M.; Elham, A.; Harik, R.: Optimal variable stiffness distribution for a composite plate with a row of holes subjected to tension/shear load cases, 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2015, 1-14. <http://dx.doi.org/10.2514/6.2015-2488>
- [4] Blom AW.: *Structural performance of fiber-placed, variable-stiffness composite conical and cylindrical shells* PhD Thesis, Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands, 2010.
- [5] Pinho, S. T.; Davila, C. G.; Camanho, P. P.; Iannucci, L.; Robinson, P.: *Failure Models and Criteria for FRP Under In-Plane or Three-Dimensional Stress States Including Shear Non-Linearity*, NASA Langley Research Center, Hampton, VA, United States, 2005.
- [6] Marouene, A.; Boukhili, R.; Chen, J.; Yousefpour, A.: Buckling behavior of variable-stiffness composite laminates manufactured by the two-drop method, *Composite Structures*, 139, 2016, 243-253. <http://dx.doi.org/10.1016/j.compstruct.2015.12.025>