Automated Path Generation for Robotic UT Inspection of CFRP Components

Loïc SÉGUIN-CHARBONNEAU [1], Julien WALTER [1], Pierre-Olivier DUBOIS [1], David St-JACQUES [2], Michel BRASSARD [2]

[1] Centre technologique en aérospatiale, Saint-Hubert, Québec, Canada
e-mail: l.seguin-charbonneau@cegepmontpetit.ca, julien.walter@cegepmontpetit.ca, po.dubois@cegepmontpetit.ca

[2] TD NDE, Laval, Québec, Canada
e-mail: dstjacques@tdnde.com, mbrassard@tdnde.com

Abstract
This paper describes an automated method for the generation of robotic paths based on the part's geometry and on the specific constraints of 0°LW phased-array UT inspection. The core idea is to generate a raster scan path on a flattened representation of the inspected surface using curvature information. The input is a 3D CAD file of the surface of the component to be inspected. To simplify geometric computations, the input surface is first discretized into a triangle mesh. The curved surface mesh is then mapped to the plane using a transformation called boundary first flattening. This conformal map minimizes area distortions and yields a flattened representation that preserves general shapes and is thus easily recognizable by the operator. It usually has an inverse that maps the flattened mesh back to the 3D-embedded mesh. Using standard techniques from discrete differential geometry, principal curvatures are computed. The surface is segmented based on curvature values. In each segment of the flattened surface, a raster scan path is computed using the principal component with the largest eigenvalue as the scan direction. The distance between successive scan lines is computed based on the curvature perpendicular to the scan direction along the last scan line and to the UT probe characteristics to ensure a good coverage as well as a sufficient overlap between passes. The larger the curvature, the smaller the distance between scan lines. Finally, the path on the 2D mesh is transformed back into 3D coordinates using the inverse of the flattening map and exported to a simulation and offline programming software for final machine code generation. The proposed method requires minimal operator intervention while allowing for flexibility in the selection of the inspection parameters. A robotic system programed with the developed method is used to inspect CFRP components. Inspection results are presented and discussed.

1. Introduction

Carbon Fiber Reinforced Polymers (CFRP) are extensively used in the design of aircraft structures, as they allow for significant weight savings while providing high mechanical strength and resistance to fatigue and corrosion. To ensure optimal performance and prevent catastrophic failures, CFRP structural components must be fully inspected to detect potential manufacturing flaws such as delamination, porosities or inclusions. The preferred Non-Destructive Testing (NDT) method for the inspection of composite parts is Ultrasonic Testing (UT) [1].

While manual UT inspections are still widely used in the aerospace industry, they are time-consuming and strongly dependent on the technicians' skills. These are major drawbacks in a context of ever-growing requirements of reliability and cost-effectiveness. Robotization of NDT is intended as a way to achieve high repeatability and productivity, and is thus increasingly used in the industry [2].

The classical approach to inspect aeronautical CFRP components is the pulse-echo technique using zero-degree longitudinal waves (0°LW) which consists of moving an ultrasonic probe over the inspected part while maintaining a normal incident beam. The scanning path must include an overlap between successive inspection lines to ensure full coverage of the part. While being quite straightforward for simple shapes (plates or slightly curved components), such path programing can become very challenging for more complex geometries involving multiple curvatures, radii, holes, etc.

Moreover, the inspection of large surfaces often motivates the use of phased array (PA) instead of conventional probes, which further complicates the scanning strategies over curved surfaces. Indeed, PA probes generally used for composite inspection are linear (1D) or matrix (2D) arrays composed of multiple individual elements. Instead of producing a single beam as for conventional probes, a subgroup of elements (i.e. the active aperture) is excited to produce a beam which is electronically translated across the PA probe's surface. An array is thus generally larger than a conventional probe and the ultrasonic beam can emerge from different points of its front surface, not only from its center. This needs to be considered when calculating the probe's coverage: over a curved area, even if the PA probe's center aperture is normal to the surface, the beam emitted by a subgroup of elements located far from the center will have a very different incidence angle on the part and may be reflected beyond the active surface, causing amplitude drop or even signal loss. Consequently, the width covered by a PA probe will vary at each position with respect to the curvature of the surface beneath it.

This paper presents an automated offline path planning tool for 0°LW pulse-echo inspections performed with PA probes. It was designed to fulfill the following requirements:

- The scan path should be generated directly from the part's geometry (CAD file) with minimal human's intervention,
- The method should be easily adaptable to various geometries,
- The part's curvature should be considered to calculate the distances between two scan lines,
- The number of required passes to get a good coverage of the surface should be minimized,
- The classical 2D C-scan views should be produced in real-time during the inspection,
- Parts' geometry should be easily "recognizable" by the technicians on the 2D C-scans.

The core idea is to use curvature information calculated from the part's CAD model to automatically create inspection paths that cover the whole surface using as little passes as possible. Another fundamental concept of our approach is to flatten the 3D original geometry before applying a 2D raster scan path and then to rebuild the 3D scan path by applying an inverse transformation. This idea leads to simpler movements more adapted to UT raster scans and allows for the use of low or mid-range ultrasonic instruments equipped with only 2 encoders inputs (scan-index).

2. Background – Previous work

Teach pendant programing is one of the simplest approaches to program a scanning path with a robot or a Cartesian scanner. Nevertheless, considering the complexity of the parts this approach is not adapted to generate trajectories under the specific constraints of UT inspection and can be extremely time-consuming.

Various off-line path planning software are commercially available and capable of generating complex trajectories. However, these tools are designed for Computer Aided Manufacturing (CAM) processes such as milling or trimming. They often include a lot of features unnecessary for NDT purposes, while lacking the essential ones, which usually makes them expensive and cumbersome to use to program UT inspections. Moreover, even if these tools are powerful, the path programing is still manual and strongly relies on the operator's experience (to define proper part segmentations, for example).

Different approaches were proposed to automatically generate scan paths for NDT purposes directly from the part's geometry (usually using its CAD file). [3] propose a method consisting of successive automated segmentations of the part based on its CAD model and generation of a raster scan pattern over each segment. [4] propose an eddy current robotized technique for evaluating the fiber orientation in preforms and CFRP parts. The path planning in this technique is also directly based on the CAD files of the components. Both applications use small eddy current single-element probes. Hence, they do not address misalignment problems that arise when using rectangular phased-array UT probes to inspect curved surfaces.

[5] proposed a path planning software named *RoboNDT* which automatically generates raster scans paths dedicated to UT inspection of large and complex CFRP components. The path planning module uses a triangular mesh of the part to generate equally spaced parallel scan lines on the part's surface. The operator can specify the raster scan step as well as the probe footprint. However, it seems that both parameters are considered constant during the scan. It is thus unclear if this tool can deal with situations where large phased-array probes are placed over a strongly curved area (a stringer radius for example).

3. Overview of the inspection process

The objective is to fully inspect a part using a robotic system with a PA probe. The inspection process has an offline phase that must be completed once for every inspected part and an online phase that runs in real time during the actual inspection (Figure 1). Central to the process is a software, called *Flatten*, that does most of the required data processing. *Flatten* takes as input a STEP file of the component's surface. It creates a triangle mesh of the surface, maps the surface to the plane and saves the mesh and the map in a so-called *textured OBJ* file. *Flatten* then generates an inspection path in a raster scan fashion using curvature information to adjust the indexation between two passes in order to obtain full coverage with the minimum number of passes. Four path generation methods have been developed and are implemented in *Flatten*. One is detailed in the present paper

and the other three methods are briefly discussed in Section 5.4. The path is exported directly to a robot simulation and programming software (such as *RoboDK*) for final processing and generation of the robot program.

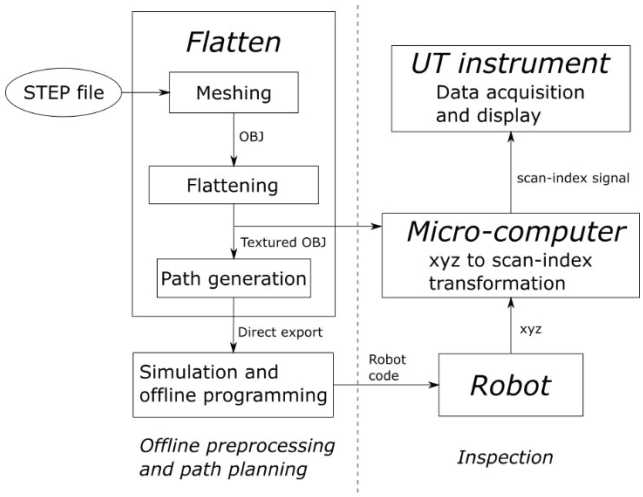


Figure 1 - UT inspection process

During the online phase of the inspection, the robot controller executes the programmed trajectory and sends the UT probe position to a micro-computer. This micro-computer transforms the three-dimensional coordinates of the robot into the plane scan-index coordinates using the textured OBJ file produced by *Flatten*. It then reproduces a classical encoder by generating an electrical signal coding the scan-index coordinates. This signal is then fed into the UT instrument which acquires the data and displays the results in real time.

4. Offline phase preprocessing

The inspection process starts with a CAD drawing of the inspected surface in STEP format. This file first goes through a couple of preprocessing steps to transform the surface into an easily usable triangle mesh and to compute curvature information required for path generation. A flattening of the curved surface mesh is also computed and stored in a file for later use in path generation and intercommunication between the robotic system and the UT instrument.

4.1 Meshing

Geometric computations on arbitrary surfaces can be challenging. However, CFRP components can usually be segmented into a few primitive shapes (spherical caps, cylinders, planes, NURBS). Each of these primitive shapes has a precise mathematical description that can be used in computations using traditional differential geometry (see, for instance, [6]). However, considering multiple shapes and figuring out how to properly join the differential properties at boundaries can be tedious. The method presented in this paper instead creates a triangle mesh of the whole surface and uses discrete differential

geometry to process the whole surface with a common mathematical framework (see, for instance, [7]).

Geometric computations perform much better on triangle meshes where the triangles are close to equilateral [8], [9]. Commonly used CAD software can export surfaces as triangle meshes, but these meshes often include skinny triangles. Hence, the specialized meshing software *Netgen* [10] is used for computing triangular meshes. It reads files in STEP format, performs corrections on the data when necessary (most importantly, stitching adjacent surfaces) and produces high quality triangle meshes. Triangle sizes are automatically adjusted to obtain a good representation of the original surface while maintaining the total number of triangles relatively small. The operator can also adjust the maximum and minimum triangle sizes when needed. *Netgen* exports the mesh as an STL file which is then converted to Wavefront OBJ format for further processing.

Figure 2 shows an example part. This part has two highly curved radii (with radius of curvature about 5 mm) and also has a smoother curvature in the direction parallel to the radii.

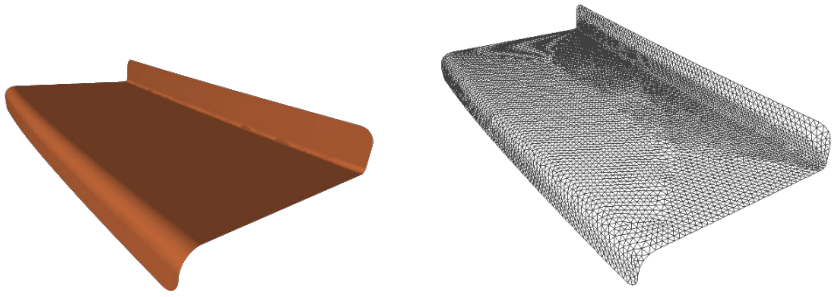


Figure 2 - CAD of a 3D component (left) and the corresponding triangle mesh of its surface (right).

4.2 Surface flattening

A flattening is a mapping from a 3D-embedded curved surface to the plane. This mapping serves two main purposes: first, automated path creation is computationally easier in two dimensions; second, it converts the three space coordinates of the robot into scan-index coordinates used by the most common phased-array UT instruments. Thus, small and mid-range systems generally equipped with 2 encoders inputs can be used.

The inspection process can run with very little operator interaction. However, during path generation and UT data display, the operator is able to look at the results and easily recognize the inspected part. This facilitates validating the inspection path and locating and sizing indications. Hence, flattening the surface needs to preserve the overall shape of the surface. Several flattening algorithms were tested and evaluated by asking experienced UT technicians to assess the similarity between the 3D surface and the flattened surface. Conformal, i.e., angle-preserving, flattening produced the best results. *Flatten* uses a specific kind of conformal flattening called *boundary first flattening* or BFF [11].

This mapping is fast, has a readily available implementation provided by its authors and can compute a conformal mapping that minimizes area distortion. This allows better defect sizing when analyzing the final UT data.

The BFF implementation takes as input the OBJ file produced during meshing. It then produces a second *textured* OBJ file in which each vertex is assigned 2D texture coordinates. The textured OBJ file contains the topological information about the mesh as well as the 3D and 2D geometries. Figure 3 shows a curved mesh, its flattening and the area distortion created by the flattening. Since the flattening should allow to go back and forth between the 2D and 3D representations of the surface, it should be one-to-one. BFF does not guarantee a one-to-one map, but tests on a variety of complex CFRP parts showed that BFF did provide one-to-one mappings for these parts. This means the mapping has an inverse that maps 2D coordinates back to their original 3D coordinates.

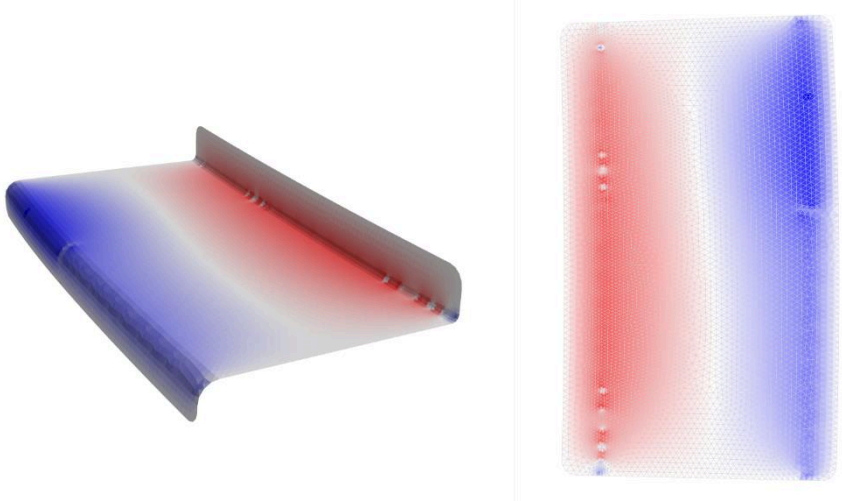


Figure 3 – 3D mesh (left) and the corresponding flattened mesh (right). Colors represent local area distortion with red indicating a triangle was stretched and blue indicating a triangle was compressed. All changes in area are less than 2% in this example.

4.3 Curvature computation

Given a point on a surface and a curve going through that point, the *normal curvature* measures the rate of change of the unit normal vector along that curve. Considering *all* the curves passing through a point, it is possible to identify the curve with the largest normal curvature and the curve with the smallest normal curvature. The directions of these curves are called the *principal directions* and the corresponding curvatures are called *principal curvatures*. Figure 4 shows principal directions at a specific point on an example surface.

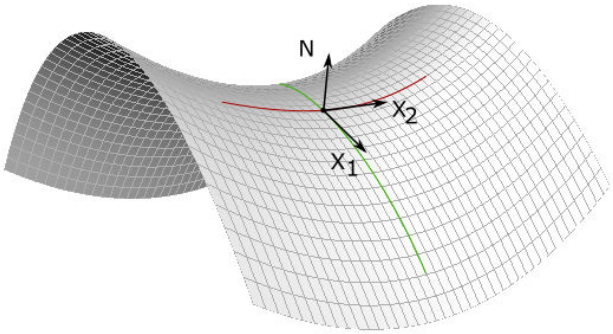


Figure 4 - Principal directions of maximal curvature (X_1) and minimal curvature (X_2) form an orthonormal basis with the surface normal (N).

Curvature, a differential property, can be discretized in many ways leading to different values of principal curvatures and directions on a mesh. The following choices were made based on ease of computation and quality of the generated paths. Face normals are computed as the cross product of edge vectors. Vertex normals are computed as the area-weighted average of the adjacent faces [12].

Even though the faces of the triangular mesh are flat, they can still correspond to regions on the original surface that were curved. Hence, it makes sense to compute the principal curvatures on the faces. This is done using the method described in [13]. The strategy is to evaluate the change in vertex normal along each edge of a face to set up a system of linear constraints on the second fundamental form (a linear operator describing curvature). Using least squares, the second fundamental form can be estimated. Then, the eigenvectors and corresponding eigenvalues of the second fundamental form are computed, giving the principal directions and principal curvatures, respectively. This method generally yields good results but tends to be unreliable in regions where curvature is small. For raster scan path generation, this is not a problem since near-zero curvature means that the part is flat, and no curvature-based adjustment is needed.

However, some path generation methods require a smooth principal direction field across the part, including flat regions. *Flatten* can compute such a field using an implementation of the globally optimal direction field algorithm described in [14]. This algorithm defines an energy which depends on both the alignment of the direction field with the principal directions as well as the smoothness of the direction field. Minimizing this energy can be done efficiently by solving a sparse linear system. Figure 5 shows the direction fields obtained by both methods.

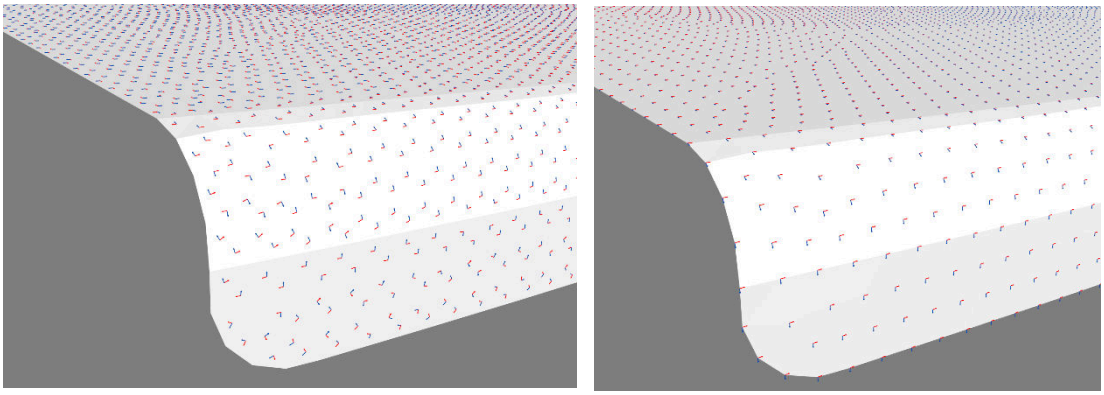


Figure 5 - Principal direction fields obtained by the methods described in [13] (left) and [14] (right). Direction of smallest curvature indicated in red, direction of largest curvature in blue. The direction field on the right is much smoother, even in flat regions.

5. Path generation

We developed four path generation methods that are all inspired by the raster scan approach traditionally used for UT inspections but also incorporate curvature information in order to ensure full coverage. This section details the *2D raster* method and Subsection 5.4 briefly introduces the other methods. All methods require a procedure for generating individual scan lines and a procedure for computing the spacing between successive scan lines.

5.1 Segmentation

The operator has the option of segmenting the inspected surface prior to path generation. Segmentation uses principal curvatures at each face to perform hierarchical clustering [15] of the faces into regions of similar curvature. It also uses connectivity information between the faces to make sure the regions are connected. Once the segmentation is done, paths can be generated in each region separately. This step is entirely optional but can prove very useful on complex surfaces.

5.2 Generation of a 2D raster path

The raster scan path is constructed on the flattened mesh of the inspected surface. First, scan and index directions are determined using principal component analysis of the vertices in 2D. The principal component with the largest eigenvalue corresponds to the longest direction of the flat mesh and is used as the scan direction. Rotating this direction by 90° gives the index direction. This choice is justified since the targeted parts in this study were all longer in a direction along which curvature was much smaller than in the perpendicular direction. When this hypothesis is not satisfied, the scan direction could be determined using the direction along which the curvature is the smallest across the whole surface. The alternative path generation methods presented in Section 5.4 were developed to circumvent this limitation.

Generating a scan line consists in creating a succession of points along the scan direction. All scan lines are parallel (in the flat mesh). For a conventional (i.e. single element) probe,

the separation between the scan line would be the beam width minus a small overlap. In the case of a phased-array probe, each active aperture generates an ultrasound beam. The probe is positioned in such a way that the ultrasound beam generated by the central active aperture has normal incidence meaning the central active aperture will always receive a significant portion of the reflected ultrasound beam. However, for active apertures far from the probe center, the surface curvature might cause a large part of the ultrasound beam to be reflected away from the active aperture. Hence the separation between scan lines must be computed based on the beam properties, the surface curvature and the probe/part distance. Using a simple model for the ultrasound beam shape and the curvature in the direction perpendicular to the scan line, it is possible to compute the covered width for each position of the probe. To simplify computations, the probe is considered to be moving along the surface in the flattest direction, which implies the perpendicular curvature is the maximum of the absolute value of the principal curvatures. While this might not always be exactly the case, this assumption can only lead to small underestimation of the covered surface, i.e., the method errs on the side of caution and might generate a few extra scan lines to ensure good coverage. For each scan line, the minimum coverage across the whole line is used to compute the distance to the next line. The resulting path has variable distances between scan lines.

5.3 Mapping a 2D path to 3D

Once the flat raster scan path is created, it needs to be transformed back into the original 3D coordinate system. The information to perform this transformation is contained in the textured OBJ file produced by the BFF program. However, this file only contains the mapping between 3D and 2D coordinates of the mesh's vertices. The mapping for points other than the vertices needs to be inferred in order to transform the 2D raster path whose points usually do not lie directly on the mesh vertices. Given an arbitrary point in scan-index coordinates, its nearest triangle is determined and the barycentric coordinates of the point are computed [16]. The 3D coordinates of the point are obtained by linear combination of the triangle's vertices with barycentric weights. Figure 6 shows an example path.

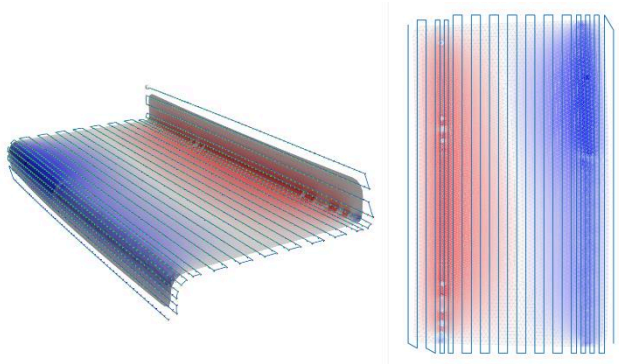


Figure 6 - Raster-like path generation starts by generating a raster on the flat mesh (right) and then transforming this path back in 3D (left).

Once the whole path is transformed into 3D coordinates, it can be exported directly into a robot simulation and programming software such as *RoboDK*. This software allows the user to specify the part location according to the robot as well as to optimize the robot configuration for singularities and collision avoidance. Finally, the program is exported as an SRC file and transferred to the robot.

5.4 Other path generation methods

Flatten offers three other path generation methods. All these methods use the same approach to calculate coverage but differ in the way the scan lines are constructed. They all use the principal directions to determine how to construct the scan lines. For these methods, it is crucial that the principal direction field be smooth, meaning the method of [14] presented in Section 4.3 is used.

Some parts do not have an obvious longest flat direction that can be used as the scan direction. For these parts, it is better to have a scan line that curves along the surface, following the locally flattest direction. The *2D minimum curvature path method* generates such scan lines. It computes principal directions on the 3D mesh and transforms them to the 2D coordinate system using the flattening produced by BFF. A scan line is generated in the flat mesh by incrementally moving in the flattest direction at the current point. Since the parts considered in this study had few or no saddle points, the flattest direction at each point was the principal direction with the smallest principal curvature in absolute value. Once the end of the scan line is reached, indexation is computed in the direction perpendicular to the current scan direction. The 2D path is then transformed into a 3D path according to the same procedure as in Section 5.3.

The *3D minimum curvature path method* is similar to the *2D minimum curvature path method*, but it generates a path directly in 3D space. The advantage is to avoid small errors that accumulate as geometric information is converted back and forth between the 3D and 2D spaces. Again, scan lines are built by repeatedly stepping in the locally flattest direction. Since computations are done in 3D, it is necessary to project the current position onto the mesh since following a tangent direction always leads outside the part (unless curvature is zero). This method avoids using the flattening altogether.

Like the *2D minimum curvature path method*, the *minimum curvature line sweep method* uses the flattening. The first scan line is generated in the same way but starting at the vertex with the largest curvature. Then, this scan line is swept across the part. This has the benefit of producing a path where all scan lines are parallel (in 2D). The 2D path is then transformed into a 3D path like in Section 5.3.

6. Inspection

6.1 Hardware configuration

To ensure a proper synchronisation between the robot controller and the UT instrument, the system shown in Figure 7 was developed. This system involves three main components: the robot controller (KRC4), a micro-computer and a custom coupling board.

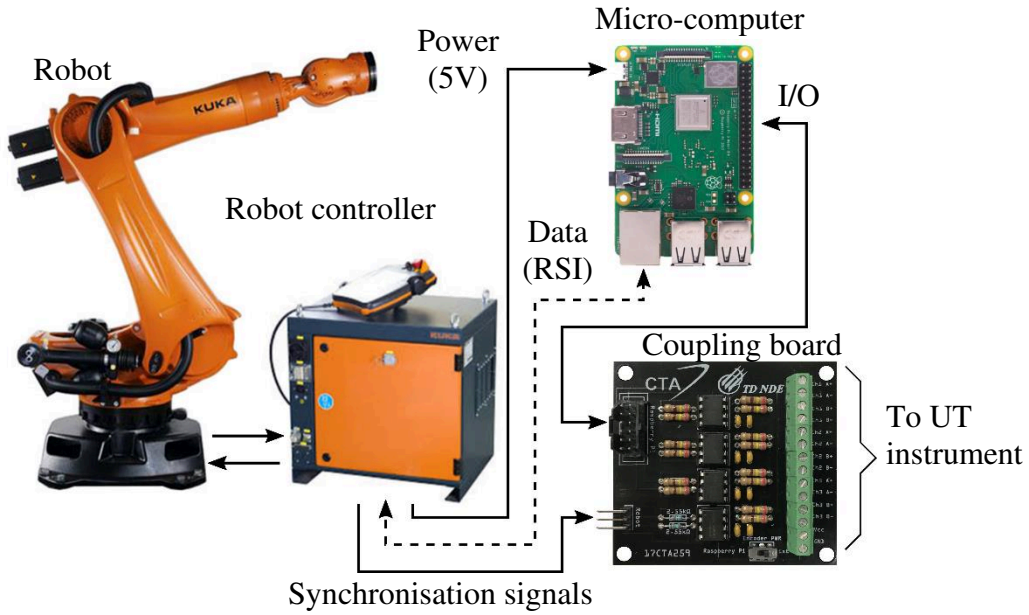


Figure 7 – Overall view of the robotic UT inspection system

The KRC4 communicates with the micro-computer via an option of KUKA controllers, namely the *Robot Sensor Interface* (RSI) [17]. This option sends the position, velocity and acceleration of the UT probe installed at the end of the robot relative to the part every 4 ms. In addition, it permutes a digital output of the KRC4 that is connected to an input of the micro-computer through the coupling board (the synchronisation signal). The purpose of the synchronisation signal is to use a built-in *interrupt on state change* function on the micro-computer to make the system synchronous. The change of state triggers the generation of signals replicating the behavior of optical encoders. Again, these signals go through the coupling board. This card is mainly made of optocouplers with very low latency (40 ns). It serves as a galvanic isolator, allowing every device to have its own electrical sources and voltage levels.

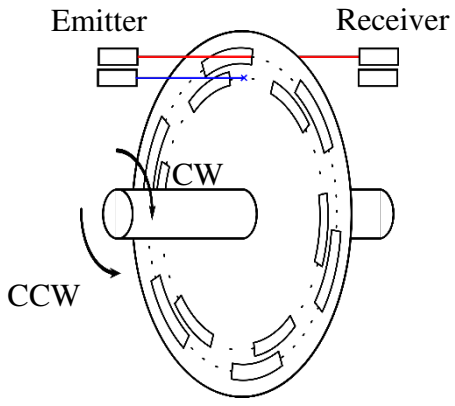
6.2 Signal processing

The purpose of the signal processing is to convert the actual position given by RSI to an optical encoder signal. The first step of the processing is to compensate for the communication latency and discrete measurement of the position. Indeed, since we want to run the robot up to 500 mm/s and that RSI gives the position every 4 ms, this could lead

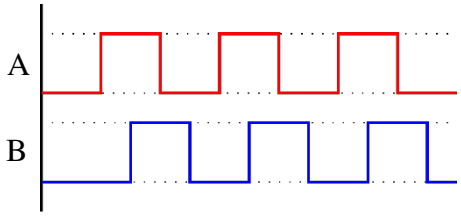
to a synchronisation error of up to 2 mm, which is not acceptable for the application. Hence, a position estimation algorithm was developed. Knowing that the robot interpolation cycle period (12 ms) is greater that the RSI period and that the robot's motors have a constant acceleration during each cycle, one can use the uniform acceleration motion equation to estimate the robot's position in this period. Furthermore, since the system is synchronous, it is possible to measure experimentally the exact time delay required for the position estimation. This estimation procedure leads to a maximal error of 0.1 mm at 500 mm/s.

The second step is to convert the *xyz* position into a scan-index (*uv*) coordinate system. To do so, the micro-computer uses the textured OBJ file produced by the BFF algorithm. The 3D coordinates are first projected onto the surface. Each projected point is associated with the nearest triangle and its barycentric coordinates are computed. Transforming into the scan-index then uses the same technique that allows transforming the 2D path to 3D, i.e., by linear interpolation (see Section 5.3 for details).

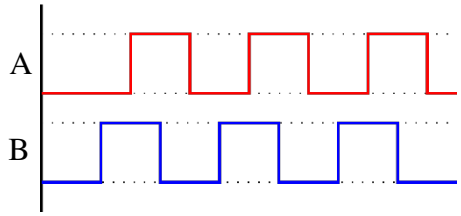
Finally, since the UT instrument needs to receive signals from an optical encoder as shown in Figure 8, four signals are generated by the micro-computer (two signals for the scan input and two for the index input). Every 4 ms, the micro-computer compares the last *uv* position sent to the UT instrument and the predicted *uv* position and generates some pulses on the A and B channels of the scan and index inputs. To add or subtract a value, the generated signal will either have the clockwise rotation (CW) order (channel A leading channel B) or the counterclockwise (CCW) order (channel B leading channel A). The number of pulses will then simply be equal to the *uv* difference divided by the encoder resolution configured in the UT instrument, in our case 20 steps per mm.



a) Mechanical schematic of an optical encoder



b) Signals for clockwise rotation (CW)



c) Signals for counterclockwise rotation (CCW)

Figure 8 – Schematic and signals of an optical encoder

7. Results

The scan path generation method using the 4 developed algorithms was tested on various aeronautical components: different L-shape stringers, a Z-shape part with double-curvature and a complex CFRP landing gear component. For each part, it was verified that at least one of the methods was able to automatically generate a raster scan path that an experimented operator would expect. For each of these components, the scan paths were generated in less than one minute.

A Z-shape component was selected to perform an experimental validation of the whole inspection process (Figure 9). This part includes 2 small radii as well as curvatures in two directions. Four steps were machined in the center portion of the part to obtain areas with different thicknesses: 6.72 mm, 5.35 mm, 4.35 mm and 2.35 mm. Two rows of flat bottom holes (FBH) of diameters Ø3 mm and Ø6 mm were drilled at different depths in these steps. FBH of diameters Ø3 mm and Ø6 mm were also included in the radii: 6 were drilled on the back face of the "external" radius (noted R_{ext} on Figure 9) and 12 on the back face of the "internal" radius (noted R_{int}).

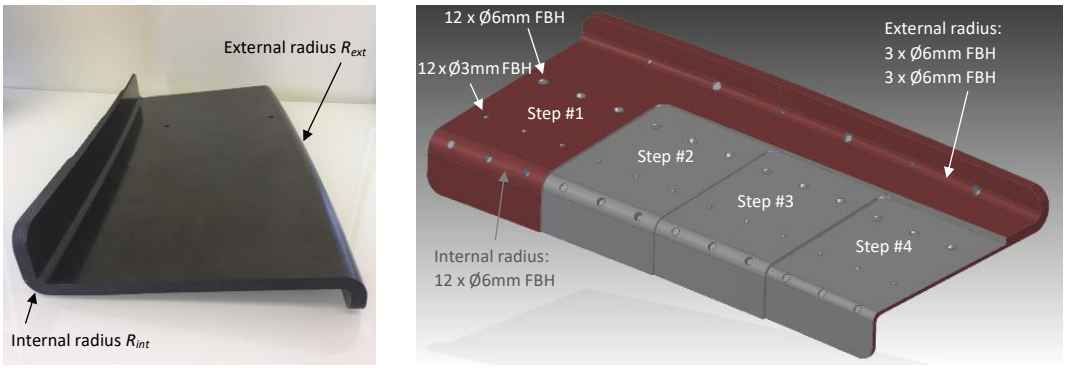


Figure 9 - Z-shape part used for experimental tests.

The CAD model was loaded in *Flatten* to be automatically meshed and flattened to a 2D surface. A scan path was then calculated with the method detailed in this paper. The generated path is a raster scan adapted to this geometry (Figure 10). In particular: the algorithm was able to automatically identify the longest dimension as the scan direction and to correctly position the scan lines; the scan lines density was automatically increased in the radii, as a consequence of the variable coverage calculation coded in *Flatten* which considers the part's local curvature.

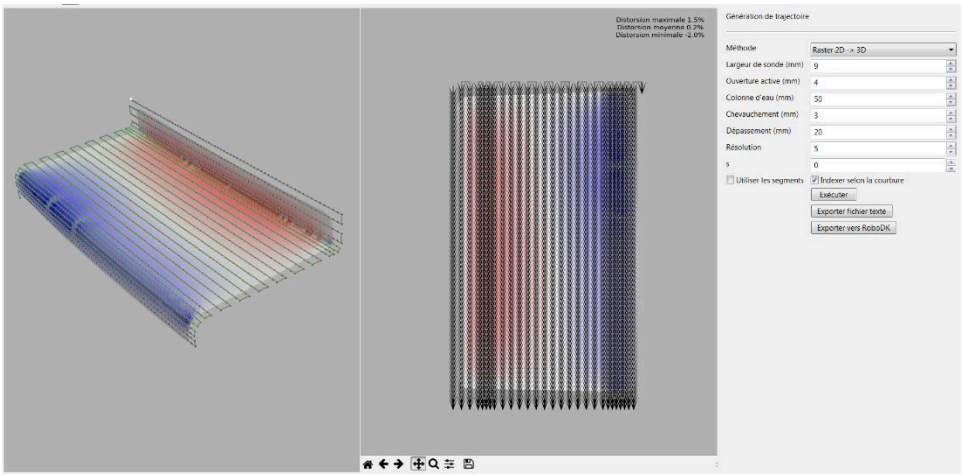


Figure 10 - Screen capture of Flatten showing the generated raster path for the Z-shape component.

An inspection of this part using the generated scan path was performed in an immersion tank equipped with a KUKA KR10 R1100 robot (Figure 11), using a 5 MHz, 64 (4 x 16) elements matrix array. The elements array had a pitch of 0.6 mm along the probe length (16 elements) and a pitch of 1.1 mm in the other directions (4 elements). A quasi-square aperture was used to perform a linear scan. The ultrasonic instrument was a M2M MultiX++.

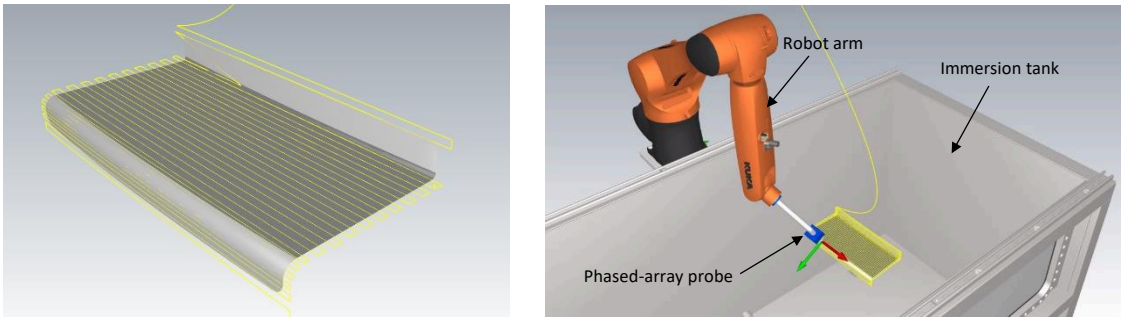


Figure 11 - Inspection path as seen in the robot simulation software *RoboDK*. The path is exported directly from *Flatten* to *RoboDK*.

It is worth noticing the bounding box of the probe and probe-holder available for this inspection was not optimized for this kind of geometry. Consequently, we chose not to inspect the "internal" radius (noted R_{int} on Figure 9) in the same path to avoid collision. The C-scans presented below are thus representing a path only including the quasi-flat portion of the part and the "external" radius R_{ext}.

Amplitude and time of flight (ToF) C-scans were produced during the inspection (Figure 12). All the FBH included in the central quasi-flat portion of the part are detected. On step #3, the black spots observed on the ToF C-scan, corresponding to the darkest zones on the amplitude C-scan, are caused by surface delamination induced during the machining of the steps.

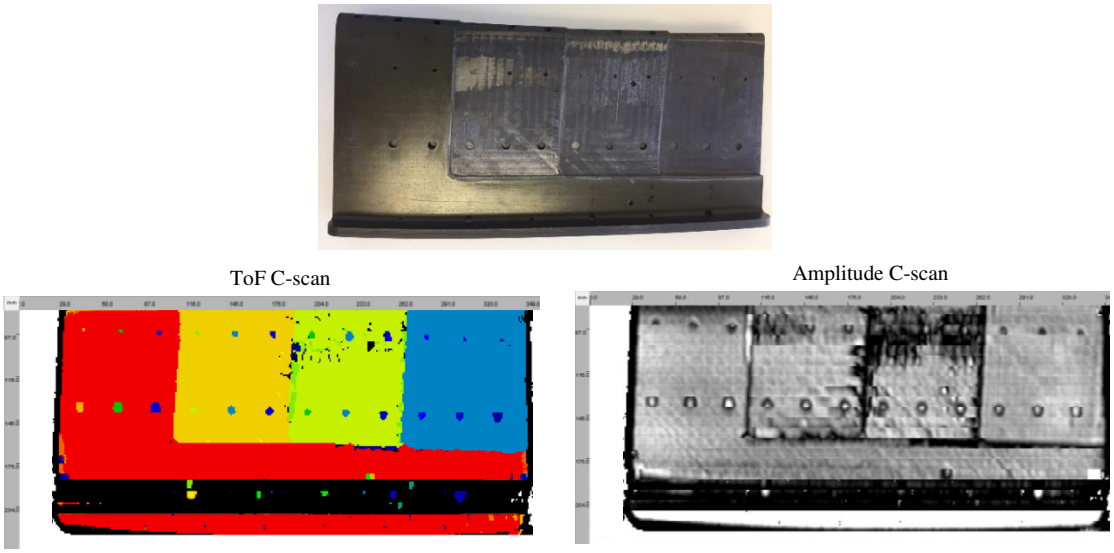


Figure 12 - ToF and amplitude C-scans obtained for the Z-shape component.

The backwall in the radius region appears black on the ToF C-scan because the received amplitude drops in the radius and we choose to code in black the pixels where no echo was crossing the gate's threshold. However, this effect does not hinder the detection of 5 out of the 6 FBH drilled in the radius. In production, an additional gain could easily be applied automatically during the scan only for this region.

8. Conclusion

An offline path planning tool dedicated to 0°LW pulse-echo UT inspections using phased-array probes was developed. This tool only requires the part's CAD file as input data. It computes a triangular mesh of the component's surface, proceeds to a flattening that avoids excessive distortions and then automatically generates a raster scan path adapted to the part's geometry. The optimal indexation distance between each scan lines is calculated by considering the part's local curvature. The path generation process was tested on various CFRP components and showed that the planning tool was able to generate adequate scanning paths in a few seconds even for complex geometries. An experimental validation of the whole process was conducted on a Z-shape CFRP component with double-curvature using a KUKA robot coupled to an immersion tank. The C-scans produced in a single inspection program allow for the detection of all the defects in the nearly flat regions of the part and 5 defects out of 6 in the radius.

Acknowledgements

This research was supported by National Sciences and Engineering Council of Canada (NSERC).

References

1. G. L. Workman, D. Kishoni, and P. O. Moore, *NDT Handbook, Volume 7, Ultrasonic testing, Editor P.O. Moore*, 3rd ed. USA: American Society for Nondestructive Testing, 2007.
2. R. H. Bossi, *ASNT Industry Handbook: Aerospace NDT*. USA: American Society for Nondestructive Testing, 2014.
3. P. Olivieri, L. Birglen, X. Maldague, and I. Mantegh, "Coverage path planning for eddy current inspection on complex aeronautical parts," *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 3, pp. 305–314, 2014.
4. J. Mersch, G. Bardl, A. Nocke, C. Cherif, M. H. Schulze, and H. Heuer, "Development of a method for the non-destructive evaluation of fiber orientation in multilayer 3D carbon fiber preforms and CFRP with robot-guided high-frequency eddy current testing technology," presented at the 10th International Symposium on NDT in Aerospace, Dresden, Germany, 2018.
5. C. Mineo, S. G. Pierce, P. I. Nicholson, and I. Cooper, "Robotic path planning for non-destructive testing – A custom MATLAB toolbox approach," *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 1–12, Feb. 2016, doi: 10.1016/j.rcim.2015.05.003.
6. M. P. Do Carmo, *Differential geometry of curves and surfaces*, 2nd ed. Courier Dover Publications, 2016.
7. M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds," in *Visualization and mathematics III*, Springer, 2003, pp. 35–57.
8. M. Bern and D. Eppstein, "Mesh generation and optimal triangulation," *Computing in Euclidean geometry*, vol. 1, pp. 23–90, 1992.
9. J. R. Shewchuk, "What is a good linear element? Interpolation, Conditioning, and Quality Measures," in *Proceedings of the 11th International Meshing Roundtable*, Ithaca, New York, 2002, pp. 115–126.
10. J. Schöberl, "NETGEN - An advancing front 2D/3D-mesh generator based on abstract rules," *Computing and Visualization in Science*, vol. 1, no. 1, pp. 41–52, Jul. 1997, doi: 10.1007/s007910050004.
11. R. Sawhney and K. Crane, "Boundary First Flattening," *ACM Transactions on Graphics*, vol. 37, no. 1, pp. 1–14, 2017.
12. K. Crane, Discrete Differential Geometry: An Applied Introduction. 2019.
13. S. Rusinkiewicz, "Estimating curvatures and their derivatives on triangle meshes," in *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission*, Thessaloniki, Greece, Sep. 2004, pp. 486–493.
14. F. Knöppel, K. Crane, U. Pinkall, and P. Schröder, "Globally optimal direction fields," *ACM Transactions on Graphics*, vol. 32, no. 4, 2013.
15. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Springer, 2009.
16. C. Ericson, *Real-Time Collision Detection*. Elsevier, 2005.
17. KUKA Roboter GmbH, "KUKA.RobotSensorInterface 3.2 - For KUKA System Software 8.3." Dec. 12, 2013.