Benefits of Digital Twin of Robotized Infrared Flash Thermography for Offline Programming

by S. Maillard*, C. Dubos**, A. Aigueperse**, J.-N. Frouart* and B. Bazin*

* Safran Composites, 33 avenue de la gare, 91760 Itteville, France, *samuel.maillard@safrangroup.com*
** Twin Robotics, 6 rue Jean Giono, 38100 Grenoble, France, camille.dubos@twin-robotics.fr

Abstract

This article presents the benefits to use an Off-Line Programming (OLP) software dedicated to the NDT in order to program a flash thermography robotized machine. Such software uses a digital twin – which are detailed – to permit to a NDT expert with few skills in robotics to autonomously program the robotized inspection machine. A guideline to define the digital twin of the sensor and the process rules is described.

OLP methods adapted to the NDT are presented and compared.

1. Introduction

In order to satisfy industrial requirements, non-destructive testing (NDT) are being more and more automatized. It increases accuracy while reducing inspection time and human factor. Robotization is a complementary step, which additionally allows to reinforce autonomy and versatility of automatized NDT. For complex shape parts, the quality of this robotization is directly related to the simulation and Off-Line Programming (OLP) software. There exists a lot of simulation and OLP software, but few are specialized for NDT. In this paper, the implementation and advantages of using a NDT specialized software will be detailed. Flash thermography inspection will be here considered as an industrial use-case.

Firstly, the mandatory tools to be included in a simulation and OLP software will be listed; e.g. excitation source, camera and tooling in the case of infrared inspection cells. Moreover, the details of the preparatory phase of the software utilization are explained. It aims at identifying the rules of the inspection process. These rules are often naturally known by the manufacturing engineers but are usually not written. Specifying them allows to increase the quality of the simulation and the trajectory generation, given a specific inspection process.

Then, based on these elements, the mathematical concepts used to generate the trajectories for the inspection, avoid collision and optimize robot paths will be briefly described. Furthermore, OLP software – with its digital twin – have a lot of information which may be injected into the digital workflow. Its integration is described.

Finally, a presentation of practical applications will highlight both the benefits of a NDT specialized OLP software and the opportunity to improve the digital thread in an industrial environment. After presenting different way to program a robotic cell (manual and automatic programming), a comparison of those methods on a flash infrared thermography cell will be discussed.

2. System and process modeling

Robot programming can either be done inline or off-line, and using multiple methods that can be seen on Fig. 1. Robot programming methodsFig. 1. The advantages and drawbacks of those methods are explained in [1]. The scope of this paper is limited to off-line programming only, which is done using a simulation and programming software. But before programming any robots, the digital twin of the machine must be defined in order to be simulated.

2.1. Digital twin definition

A digital twin is a virtual representation of an object or a process that allows the simulation of its functioning and characteristics [2]. Three types of digital twins can be defined, based on the purpose of use:

- **System:** represents the modeling of one element. In our case, it is the digital twin of the robot (or its kinematics) or the thermography sensor. The goal is to have the most accurate and reliable representation in order to simulate the system behavior's in the most efficient way. Their configuration requires a significant expertise of the system.
- **Machine:** represents the modeling of a machine composed of multiple systems. It allows to simulate the interactions between the different composing systems, along with to integrate the management of security. Being as close as possible to the machine functioning, it aims at being used by the machine operator who is not necessarily expert in using the different systems. This digital twin is used in trajectory generation.
- **Factory:** represents the machines and their environment. It enables to simulate the interactions between the machines and the production workflows. It can also be used to compute the part manufacturing times or to pinpoint potential bottlenecks in the industrial processes.

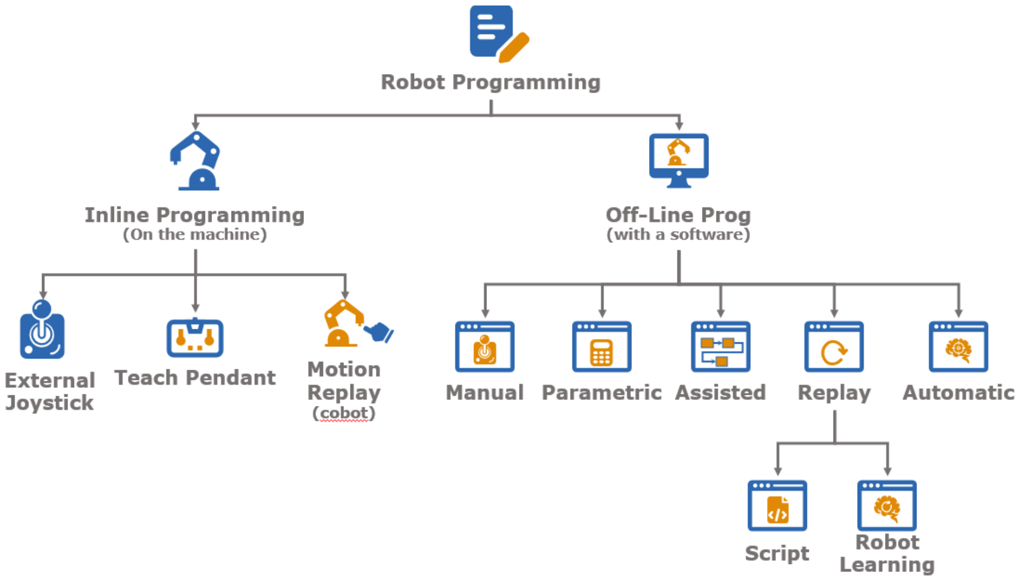


Fig. 1. *Robot programming methods*

Having the digital twin of the machine (robot and sensor) appears like a necessity for a NDT application. The sensor's digital twin allows to simulate the acquisitions and to have a most precise information on the acquired data, while the robot's digital twin enables to validate the acquisitions from a mechanical point of view. Both of these seems like the minimum requirements.

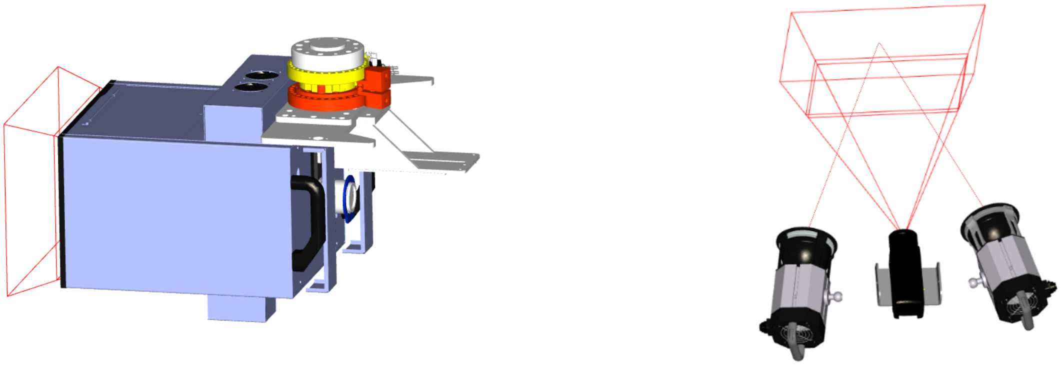
Indeed, the validity of an acquisition scenario comes from its robotic feasibility. The robot's digital twin then allows to validate that all the acquisitions poses are reachable regarding the workspace. The lack of collisions may also be tested for each acquisition pose as well as along the whole inspection trajectory. This trajectory can be generated by taking into account the machine's speed limits and also the acquisition parameters, such as heating time, acquisition time… All those parameters are needed in order to automatically program a machine.

It is also worth noting that a digital twin of a machine allows to limit the robotic knowledge needed for an operator to program the machine since all the technical expertise linked to the robotic is managed by the digital twin.

2.2. Thermographic sensor modeling

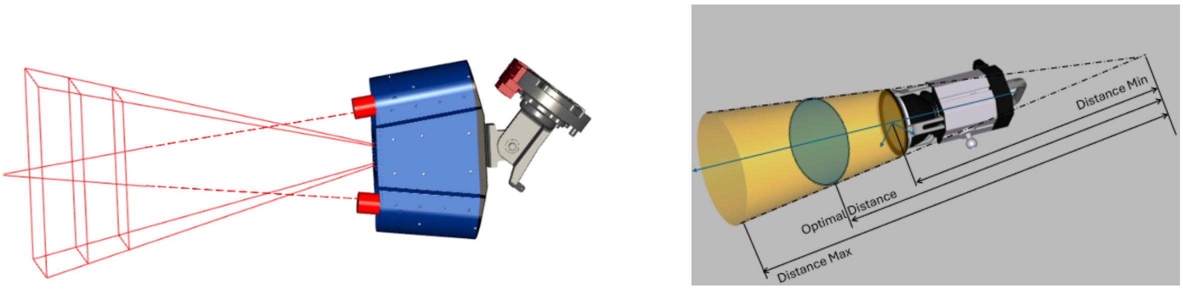
Since the goal of the sensor's digital twin is to validate the sensor's pose in reference to the part, in the first instance it is thus deemed unnecessary to have a digital twin as complex as available in the software CIVA [3]. Even if the inspection method is a volume inspection, the modeling of a thermographic sensor can be made into a surface inspection method, based on optical laws (raytracing, reflection angle…), as it is done in metrology [4], [5].

A thermographic sensor consists of a camera and one or more components for thermal excitation. In Fig. 2. Examples of thermographic sensors modeling, some thermographic sensors and their modeling are depicted. The camera can be easily represented by a pinhole camera whereas the other components may have different representations. First, the thermal excitation component can be modeled as a omnidirectional light source. In this case, a simple raytracing test is enough to determine if the surface is heated or not. However, another model based on a conical frustum is actually studied, as depicted in Fig. 2. Also, in the case of a sensor composed of multiple excitation components, the visibility conditions must be clarified in the digital twin: should the surface be seen by one or all the sources?



a. *EchoTherm sensor (ThermalWave Imaging)*

b. *Edevis sensor with 2 flash lamps*



c. *Voidsy sensor* ***d.*** *Studied evolution of flash lamp modeling*

Fig. 2. *Examples of thermographic sensors modeling*

2.3. Process rules

The inspection rules describe the machine/sensor behaviors. These rules are often naturally known by the NDT expert but not written. The main feedback, that is interesting to mentioned, is those rules are complex to formalize. At the beginning of the project, thermographic experts have formulated rules but after implementing them in the OLP software, with a systematic application, it was necessary to reformulate or change the rules priorities. Finally, the expert's skills were improved by this specification tasks.

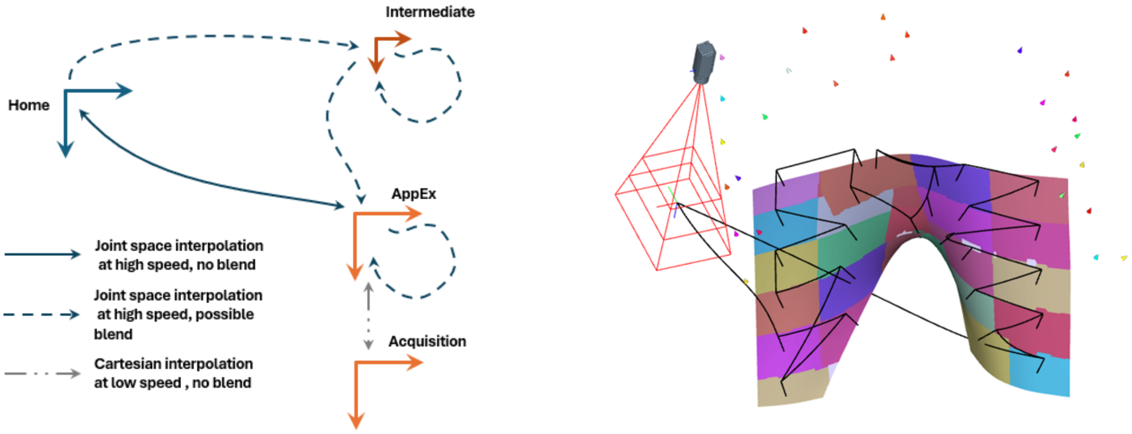
The following sections detail process rules for the robotics and behavior of the sensor with reference to the part to inspect.

2.3.1. Robotics rules

The robotics rules concern the motion interpolation between points, the velocities of the robot, etc. In preparation of this task, 4 types of points were identified to populate the inspection scenario:

- **Home Position:** the inspection scenario starts and finishes at this position. For robotics reason, a scenario cannot be sent to the robot if it doesn't includes those 2 positions. The home position is the waiting position of the robot.
- **Acquisition position:** the robot will reach this position and call the acquisition procedure.
- **Approach & Exit position (AppEx):** to avoid collision during a motion between two acquisition positions, two safety motions are required with reduced speed and cartesian interpolation. To simplify the motion graph, the approach and exit points are confounded.
- **Intermediate position:** to avoid collision between AppEx positions, one or several positions can be included in the inspection scenario. To reduce the computation complexity, the robot makes stop on those positions. In further works and in the perspective to reduce the cycle time, a blend effect on robot's deceleration should be added.

The motion graph presented in Fig. 3 describes the trajectory parameters (interpolation method, speed, blend, etc.) between all those types of points. Notes that some motions are constrained: acquisitions can be reached only by passing by AppEx because the sensor is working very closely with the part to inspect.



a. *Motion graph* ***b.*** *Example of inspection scenario*

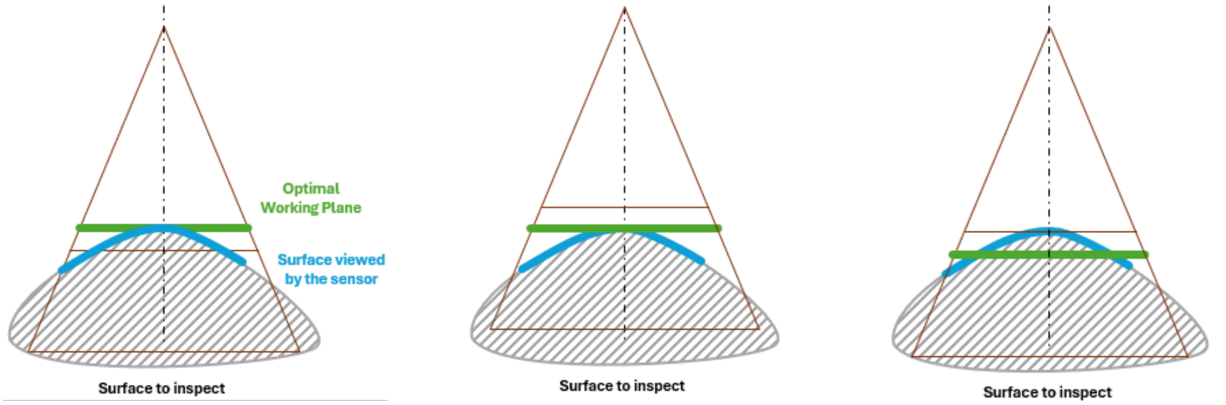
Fig. 3. *Robotics rules*

2.3.2. Sensor rules

The first question to answer is which parameters should be controlled by the OLP software. For a thermography sensor it can be excitation parameters (wave forms / intensity, duration, etc.), or camera parameters (delay for the first shoot, integration time, number of shoot), or other (cooling time between 2 acquisitions).

To determine which parameters to manage, the following question must be answered: what is the optimal scenario for an acquisition? The first answer is to be the closest as possible to the area to transmit the maximum energy from the sensor to the part. The camera digital twin integrate the concept of *Optimal Working Plane* useful to positioning the sensor on a given point. The camera lens – respecting the pinhole model – has three planes: Z near, Z Working Distance and Z Far. Usually, the plane Z Working Distance used is the same as *Optimal Working Plane*.

Second question: knowing the area to acquire, what is the reference point? The Fig. 4 illustrates several configurations to optimize the sensor pose regarding to a surface to inspect. In the illustration, the collision aspect between the sensor and the part is not taken into account. For the EchoTerm device (Fig. 2.a), a strategy to avoid collisions must be implemented. This topic is still discussed, and all the possibilities are being tested.



a. *Optimal plane on the Z Near plane and sensor on the highest point to acquired*

1. *Sensor on the highest point to acquired*

2. *Sensor on the average altitude of points to acquired*

b. *Optimal plane on the Z Working Distance plane*

Fig. 4. *Rules on the optimal sensor localization to acquire a given area (blue line)*

3. Off-Line programming software

The digital twin is the mockup of the real inspection machine. Technically, it is a data component of the Off-Line Programming (OLP) software with algorithms or functionalities that will use the digital twin to perform simulation. To fully understand the interest of having a complete digital twin, it is important to study the next step following the OLP, which is the program adjustment. Before validating the program, it is obligatory to test it on the real machine to ensure the visibility of all the desired zones. During that phase, the operator must go back and forth between the simulation and the real machine to readjust the program if needed. This tedious task can be accelerated by using a software specialized for NDT applications, allowing to have a preview of the acquisitions. Moreover, time spent on the adjustment is directly related to the precision of the digital model. The more it is precise, the faster this step will be.

3.1. Digital twin settings

It may be an obvious requirement, but all the parameters of the digital twin must be at least settable or even better determinable in the OLP software. Of course, all the settings must be changed by a non-developer user which require to have a friendly user interface. It is the guarantee that the defined digital twin will be agile. For example, the description of the camera will change depending on the mounted lens. Those parameters should be set by a user interface or text file. This last option, is very powerful when this setting file is generated directly by the camera calibration step (cf. section 3.4).

For some parameters – like the robot velocity – the device defines one range of use with default values. User may want to adapt them project by project but always within the range constraint.

The last item to have an agile software is to have the capacity to create object from CAD data. As example, the software user wants to integrate a new thermographic sensor. By importing the CAD data, it should have the possibility to defined how it will handled by the robot (mounting point), its tool center point and all the parameters linked to camera field of view, light position and model, etc.

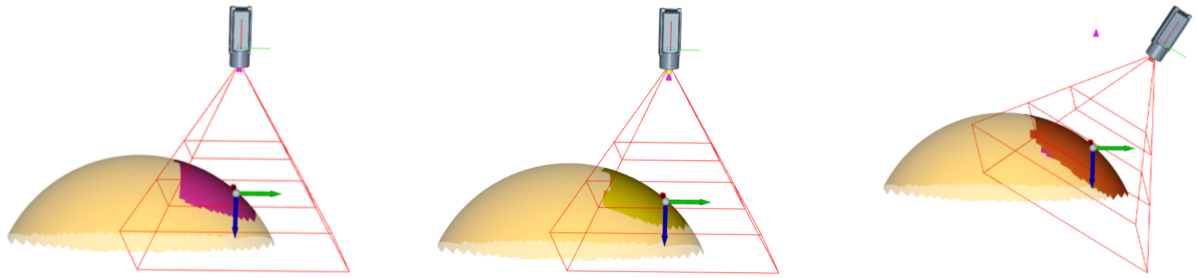
3.2. Programming: defining acquisition points

The only purpose of the digital twin is to simplify the programming step. For thermography use cases, robot programming means essentially the definition of the acquisition points. OLP methods are summarized in Fig. 1. Considering the constraint of the thermography, Manual or Automatic OLP methods are recommended in [6] and authors open the discussion on the opportunity of the assisted methods.

3.2.1. *Manual teaching method*

The manual teaching process is very similar to inline programming but made in simulation. The user creates inspection points by moving the sensor the same way objects are moved in CAD software. The digital twin of the robot allows to check the reachability of the point, the lack of collision and smooth running of the trajectory between two acquisition points. Some programming-support tools have been developed to assist the user during this process, such as a covering computation algorithm [7] to compute the overlapping between acquisitions, and an optimization sensor pose algorithm.

The digital twin of the sensor is useful to find the optimal position with reference to the surface to acquire. The optimization can be done by a trial-and-error approach or by algorithms. The Fig. 5 shows possible optimization of the sensor pose done with an algorithm that can optimize either the altitude of the sensor (b) or the altitude and orientation (c) from a reference pose (a).



a. *Initial Sensor pose (magenta cone)* ***b.*** *Optimized pose only in altitude* ***c.*** *Optimized pose in altitude and orientation*

Fig. 5. *Optimization of the sensor*

3.2.2. *Assisted method*

In [8], a method to evaluate the complexity of a OLP task for a given part and a given sensor is proposed. Two classification factors are taken into account:

- **Shape Complexity Management** [9][8]**:** describes the geometrical complexity as a number between [0; 1], based on the curvature, number of symmetric planes and number of elementary components. This factor has already been used in robotized metrology [9].
- **Acquisition ratio:** defines as the ratio of the total surface to inspect on the field of view of the sensor (see Fig. 6).

A first segmentation of OLP method based on those factors is presented in Fig. 7. This segmentation is still discussed and needs to be adjusted.

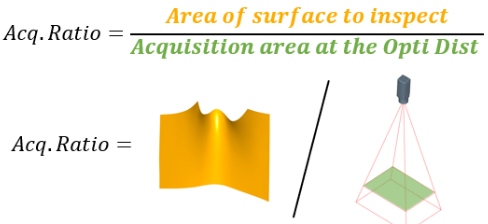


Fig. 6. *Graphical definition of acquisition ratio*

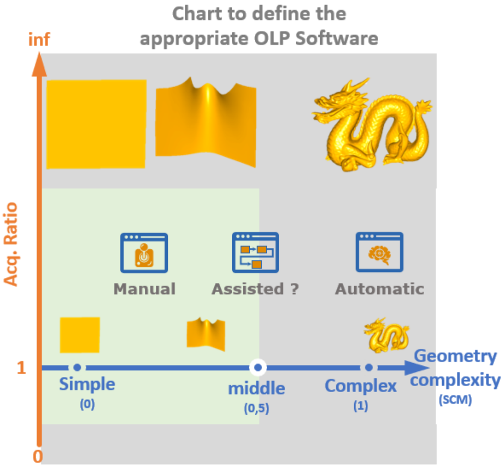
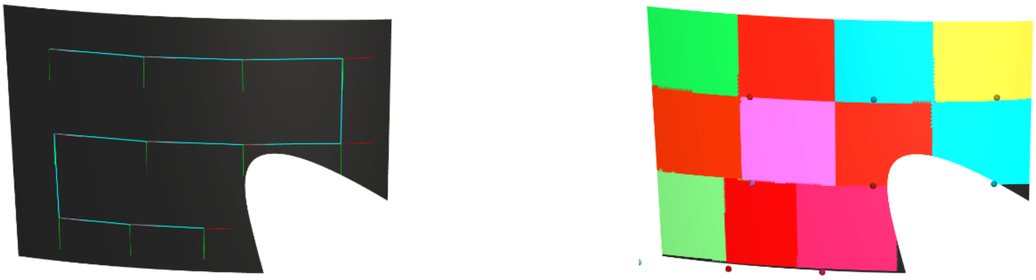


Fig. 7. *OLP methods classification*

The assisted method is already used for programming robotized ultrasound inspection machines [10]. In those cases, the trajectory is a succession of straight line, whose dimensions depend on the field of view of the sensor, as seen in Fig. 8. It is a global approach which works well on part with no sudden curvature change or stringers. Therefore, it is efficient on multiple parts, such as fuselage external parts or blades.



a. *Assisted trajectory* ***b.*** *Acquisitions results with some missing area (in grey)*

Fig. 8. *Assisted programming method applied to thermography*

3.2.3. Automatic method

Automatic programming is a process that automatically generates a trajectory using digital twins of the machine and the inspected part. Based on the part profile, the acquisition conditions and the robotic parameters, the software can generate acquisition points and a program to inspect the given part.

The generation algorithm is often an expert algorithm. Many studies and state-of-the-art of using such algorithm for metrology have been published [11]. The main difference between that field and thermography is the working distance between the sensor and the part. For instance, in metrology, the working distances are around 800 mm, while in thermography it is often between 300 mm to 100 mm. That difference implies a much important complexity.

The used method for the automatic method is iterative, so it is a greedy algorithm. First, the piece is segmented in multiple zones. Then, for each zone, a set of acquisition points are generated based on the zone's dimensions. Next, these zones are evaluated and the one maximizing a criteria based on coverage is selected. Lastly, the acquired data are subtracted from the model, and the Next Best View generation process start all over again with the non-acquired data. The algorithm finally stops when a global coverage criteria meets its threshold, or when it is not possible to generate new acquisition points (not enough unacquired data for instance). The quality of an algorithm must take into account the computation time, so it must be optimized, and there are many ways to do this.

3.3. Motion Planning

During the previous step, all the acquisition points have been defined. While doing a manual teaching, the user could have autonomously managed the collisions by adding intermediate points. But this step can also be automatically done. Indeed, a collision avoidance algorithm is used to automatically add intermediate points to avoid collisions between two acquisition points. The order of the acquisitions remains unchanged. Multiple solutions to the collision avoidance problem exists and are still studied [12].

Then, a motion planning algorithm is used to optimize the cycle time and solve the collision avoidance problem at the same time. It is based on the optimization of the robot's motion in its joint-space. This problem is much alike to the Travelling Salesman Problem [13]. One difference is that the distance matrix, containing all the distances between every points, must be smartly filled to avoid long computation time. Indeed, computing a collision-free robotic path can take up to several seconds, and the number of paths computation to fill the matrix is $\frac{n(n-1)}{2}$, with n being the number of acquisition points.

3.4. Integration into the digital workflow

The first advantage of using a NDT specialized OLP software is to have tools and functionalities adapted to the specific needs of a user, who is an inspection process expert but not necessarily a robotic one. The second advantage is the possibility to have an advanced integration with the machine, to work towards a seamless digital workflow. Indeed, the first goal of every OLP software is to generate a machine program for a given part and that program must be entirely compatible with the machine, with as little to no manual modification necessary. This continuous digital workflow can then be reinforced, based on the following points.

Before even using the software, the digital twin must be defined. It is known that during the operational phase of the inspection machine, the location of the robotic elements (such as robots, turning tables, tools) and static environmental elements (such as calibration chart, control bay) barely changes. One thing that may vary is the parameters of the sensor (camera/lens). Since calibration processes are often executed, why not used those results and re-inject them in the machine's digital twin? Indeed, a trajectory can be initially generated for a given part using theorical parameters for the sensor. Then, during the inspection phase, a calibration process is executed. Finally, the digital twin is updated with these measured parameters, so that the trajectory previously generated can be either validated or optimized with the real parameters.

The OLP software also knows the position of the sensor with reference to the part. This information is particularly useful to match with the acquisitions on the part. The goal is to go beyond a simple cartography of acquisitions and to merge the acquired data and the positions to recreate a sort of textured part. The reprojection of the data can be simplified if the OLP software exports, alongside the robot program, a file with all the positions that can be used in the post-treatment software.

Other functionalities are potential options when exploiting digital twin inside the OLP software. For each sensor pose, an acquired area is defined. From there, metadata can therefore be extracted in that zone such as minimal / average / maximal thickness, material… and used to effectively defined the excitation parameters of the sensor or the acquisition time. The OLP software then becomes a much global mean to program the robot and also the sensor.

4. OLP methods comparison

A series of tests have been made to define an inspection program manually, with the assisted method and automatically, given the same part and sensor. For each method, the results shown in the Fig. 9 are the immediate output of the programming. A motion planning algorithm (without collision avoidance) has been applied on the trajectory of the automatic method in order to smooth it. It will not impact the comparison, since the trajectory optimization is not the discussed subject, it is just an esthetic step. Unfortunately, this test could not have been realized on a real machine.

The part has a shape complexity management (SCM) coefficient of about 0.35 and its dimensions are around 1380 * 1430 * 750 (mm). It has a shape similar to a horse saddle, with an elevation in the middle. The sensor used for the tests is the one depicted in the Fig. 2.a. Its visual parameters are displayed in the Table 1 and the acquisition ratio is 23.98. The manual test was performed by an expert user of the AppRob NDT software [14].

Table 1. *Sensor parameters (in mm)*

Z near	408
Z working distance	418
Z far	498
Field of view along x axis	317
Field of view along y axis	256

The Table 2 contains all the relevant output data that will be used for the comparison. The programming time does not take into account the definition of the digital twins, but only the acquisitions generation part. The number of acquisition is important because it is linked to the cycle time, but it needs to be compared to the coverage rate, as fewer acquisitions will lead to a fastest cycle time, but less surface acquired. The surface covered by only one acquisition, meaning no overlapping, can also be relevant depending on the post treatment applied to the data. The average distance between the camera and the center of the acquisitions and the standard deviation of that distance are interesting to see how well the sensor rules are applied (see figure 4).

Table 2. *Characteristic parameters for comparison*

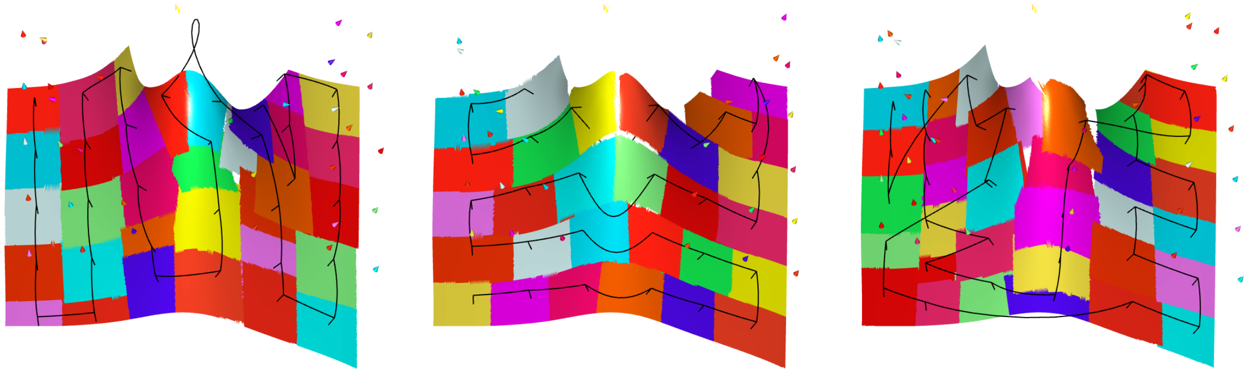
	Manual OLP	**Assisted OLP**	**Automatic OLP**
OLP Time	35 min (around 1min/acq)	15 min	5 min
Number of acquisitions	33	29	36
Coverage rate (%)	98.92	94.59	98.56
Surface covered by only one acquisition (%)	74.29	76.10	67.04
Average distance (mm)	441.498	433.446	431.729
Standard deviation of the distance (mm)	9.779	16.530	10.252
Min distance (mm)	424.844	417.998	415.887

The biggest difference between all those methods is, unsurprisingly, the time it takes to generate a program. For the manual method, it takes around 1 minute per acquisition. That can be explained by the fact that the user must not only properly place the sensor, but also check for collisions. Indeed, in the assisted and automatic processes, collisions between the part and the other elements are automatically avoided, by moving back the sensor (if it is allowed by the user), or tilting it. Whereas in the manual process, the user has to find a non-colliding pose by hand. Moreover, the user can take some time on determining the best way to inspect the part, and how to optimize the overlapping between the acquisitions. Nonetheless, it allows a better control on the acquisition positions. And of course, this notion of time nearly only depends on the desired precision for each acquisition position.

Regarding the coverage rate, the results are quite similar despite the acquisition number being not. It is worth noting that the assisted method was not developed for this type of application, so it is not yet optimized. The curvature management has yet to be improved. It can be seen in the Fig. 9.b, as the curvature gets more important, the acquisitions are moving away from one another, leaving gaps in the acquired area, but it is giving encouraging results.

Overlapping ratio can also be a matter of importance depending on the data post treatment. Indeed, for a better data reprojection, it can be useful to have overlapping between each acquisition. While the manual process offers almost no control over that parameter, the assisted method allows to specify a desired overlapping between each line and between each acquisitions in a line, and, in the automatic process, the minimal desired overlapping value for each acquisition is specified by the user. But since it is an iterative process, some acquisitions may be added to cover areas smaller than the field of view, leading to an increased global overlapping value. Consequently, the percentage of the surface covered by only one acquisition is smaller for the automatic process. Although this value is similar in the manual and assisted methods, the latter method allows a more equal distribution of the overlapping among the acquisitions. For the results shown in Fig. 9, an overlapping of 5% was set between each line in the assisted method and for the automatic process, the minimal overlapping was also set at 5%.

In these tests, the sensor rules are set so that the optimal plane is on the working distance (418 mm from the focal point of the camera) and that plane must be on the average altitude of points to acquired, if possible. Given the part profile and the sensor volume, it seems quite natural that the average distances are a little bit bigger than the desired one. The best results are obtained with the automatic method, since the computation takes into account the profile of the part. For the manual method, there is no doubt that the results will be better on a planar part, since moving the sensor along one axis is really simple. But the manual collision avoidance makes having optimized acquisition much more difficult in this case. The minimal distance value corroborates this statement, as it is almost 10 mm bigger for the manual teaching than in the automatic case.



a. *Manual OLP* ***b.*** *Assisted OLP* ***c.*** *Automatic OLP*

Fig. 9. *Inspection scenario generated by each method*

In conclusion, the manual method is useful when the part has a simple shape, as it gives the user more control over where to put the acquisition. It also does not take that much time when the acquisition ratio is low. The major drawback is the collision avoidance, which is close to non-existing on simpler parts. The impossibility to precisely control the overlapping must also be noted, but it may not always be a problem. When the acquisition ratio is large, manually teaching the acquisition can be tedious. In that case, the assisted methods works well. This method still has to be optimized to be really effective. Finally, on complex part, the automatic methods is the most recommended. It is also worth noting that both the assisted and the automatic methods can serve as a basis and be manually modified.

5. Conclusion

Of all the benefits of using an Off-Line Programming software dedicated to NDT application discussed in this paper, the major one is to allow a NDT Expert having low skills in robotics to become autonomous in the OLP task. Giving the user the opportunity to focus on the inspection process is the most important part of the robotization of those processes. Moreover, it gives the possibility to retrieve more data during the inspection, and can enhance the post-treatment of these data. On another level, this study offers the opportunity to discuss and put in writing all the rules linked to the infrared flash thermography and how to prioritize them.

Some matters still need to be addressed, such as continuing the work regarding those rules. One path to be explored is to quantify difference between the sensor rules presented in section 2.3.2 for a flash thermography process, and deduce the most adapted. Another work can be performed on the OLP methods classification. It can be improved and refined by testing multiple parts, even if the shape complexity management ratio is complicated to precisely determine. Also, a notion of bulkiness may be added to that classification since the collision avoidance is an important part of the programming process.

REFERENCES

[1] Dubos C., Boulanger T., Rochet J., & Charrier G., Shearographie: De la maquette numérique à l'inspection robotisée. LES JOURNÉES COFREND/ COFREND DAYS, France, Marseille, 06 - 08 June 2023. e-Journal of Nondestructive Testing Vol. 28(9) WCNDT Paper, 2023

[2] USD R&E : US DoD Digital Engineering Working Group, 2018

[3] CIVA website http://www-civa.cea.fr

[4] Loriot B, Gorria P, Université de Bourgogne. Automatisation de l'acquisition et des post-traitements en numérisation 3D. Phd Thesis, 2009.

[5] Aigueperse, A., & Humblot, D., Couverture optimum des zones à contrôler et auto-apprentissage des positions robot. LES JOURNÉES COFREND 2017, 30 June - 1 July, Strasbourg, 2017

[6] Maillard S., Dubos C., Aigueperse A., Frouart J.N., Bazin B., Towards Off Line Programming for robotized NDT applications, 20th World Conference on Non-Destructive Testing, 2024

[7] Korte, B., & Vygen, J., Combinatorial Optimization: Theory and Algorithms (5th ed.), Springer, 2012

[8] Sukumar S.R., Page D., Koschan A., Abidi. M., Towards understanding what makes 3D objects appear simple or complex. 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops. 1 – 8, 2008.

[9] Khalfaoui S., Automatic production of three-dimensional models by 3D digitization. PhD thesis, 2012

[10] Aigueperse, A., & Baroche, W., Etat de l'art des architectures des cellules robotisées pour le contrôle par Ultrasons. LES JOURNÉES COFREND/ COFREND DAYS, France, Marseille, 06 - 08 June 2023. *e-Journal of Nondestructive Testing* Vol. 28(9). 2023

[11] Karaszewski M., Adamczyk M., Sitnik R., Assessment of next-best-view algorithms performance with various 3D scanners and manipulator. ISPRS Journal of Photogrammetry and Remote Sensing. 119, pp 320-333, 2016

[12] LaValle S. M., *Planning Algorithms*, Cambridge University Press, 2006

[13] Lawler E. L., The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization, 1985

[14] AppRob NDT Software, Ultrasound OLP, https://vimeo.com/acande/approbutblade