IMPROVING ROBOTIZED NON DESTRUCTIVE TESTING FOR LARGE PARTS WITH LOCAL SURFACE APPROXIMATION AND FORCE CONTROL SCHEME

Olivier Patrouix, Sébastien Bottecchia, Joseph Canou

IMPROVING ROBOTIZED NON DESTRUCTIVE TESTING FOR LARGE PARTS WITH LOCAL SURFACE APPROXIMATION AND FORCE CONTROL SCHEME

O. Patrouix[1*], S. Bottecchia[1], J. Canou[2]
[1] ESTIA Recherche, ESTIA, Technopole Izarbel, Bidart, France,
[2] ESTIA Recherche, CompositAdour, Parc Technocité, Bayonne, France
* Corresponding author (o.patrouix@estia.fr)

Keywords: *Robotics, NDT, 3D surface approximation, force/position hybrid control*

1 General Introduction

The carbon fiber parts are taking more and more importance in the aeronautical field as aircraft makers are chasing the weight to comply with some of the environmental constraints. Due to certifications, composite parts, like metallic ones, need to be quality controlled. To reduce production cost, composite parts must be controlled during the machining process [1], and nowadays industrial have a growing interest in this process automation with robotic systems [2][3]. But this implies specific constraints: for example, the shape of a body panel without its stiffeners differs from CAD model however geometric parameters have to be verified.
We will present in this article the methodology used for the automation of the Non Destructive Testing (NDT) focusing on 3 main topics:

- Misalignment between CAD based robot trajectories and product to be tested,
- Local 3D surface approximation and sensor based robot control,
- Hybrid force-position robot control.

2 Industrial context

2.1 Description of the current usage

The NDT system consists of a 6 axes manipulator robot and a NDT head sensor. The tool path is generated on the basis of the CAD data of part. The process uses some control-skill constraints regarding the position/orientation of the tool vs the composite surface to be controlled.
The tool needs to be in contact with the surface, must be normal to it (sensor surface parallel to the surface to be analyzed), and velocity also needs to be controlled as well as its position monitoring depending on the process step.
In a first step the control is made using a predefined set of path and speed profiles that allows first detection of structure defaults. Then, based on this first grid, a position control allows finest analysis on specific areas. Thus during a first phase a multibeam sensor is used to detect on a given path the location of possible defaults (doubts), then in a second step a more precise detection is made using a directive sensor around the suspicious areas. This last step requires the possibility to control the rotation of the sensor head to explore the composite stacking.
The NDT control task is described using sensor frame shown in figure 1. The industrial implementation is made using the assumption that the part shape is identical to the CAD one and the workshop robot cell is also identical to the CAD model used for the off-line programming.

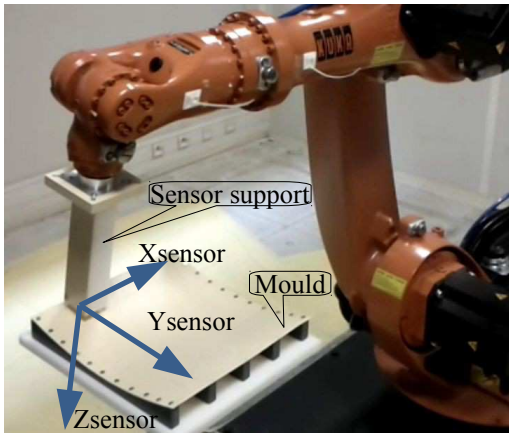

Fig. 1: NDT Task in sensor frame

Unfortunately manufactured parts, and especially composite parts, can be subject to distortions. Indeed, even if simulation are used to anticipate spring back, if the mold is adapted in consequence and if the process of curing is adapted to reduce possible occurrences of distortion during the resin vitrification, the composite material is subject to distortion.
Thus, in the best case, the resulting structure is slightly different from the initial CAD model (meaning 0.2mm/m). But majority of part presents a form tolerance of 0.5-1mm/m. Of course when talking about large structure, that are not stiffened, this value can reach higher values becoming centimetric or decimetric.
If sub-millimetric default could be managed by specific procedure or components, higher default imply the use of specific mounting structure and consequently the complexity and cost of the control process.

2.2 Observations

Because of the implementation, some problems appear during the automated control like overloaded forces between the tool and the composite part, loosing tool-part contact during the process, inaccurate orientation, leading to wrong data and errors.

3 Force Control for NDT

We firstly describe the NDT using the force-position duality, then the robot modeling to achieve the hybrid control and finally the control loop.

3.1 Modeling NDT task

The NDT task needs a dual approach between force/position and between torque/orientation. This dual approach is described in several robotics papers [4].

In the sensor frame, the directions controlled in displacement are the translations along the x and y axis and the rotation around the z axis. Around these axes the torque generated by displacement and surface distortions must allow the sensor to be normal to the surface, keeping the sensor surface parallel to the material surface.

This can be achieved by using a compliant system or using a force/torque controlled system.

The use of a compliant system and its degrees of freedom (Fig. 2) is efficient to avoid torque or force to be applied as constraints on the system. The inconvenient is that there is no possibility to manage the orientation/torque and position/force of compliant directions.

Using a force/torque controlled system offers a more suitable solution because it allows a coupled control of force and position as well as coupled control of torque and orientation.

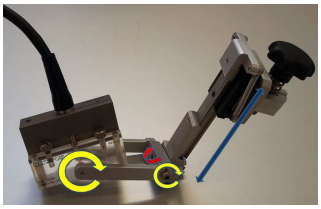

Fig. 2: Standard compliant system and its degrees of freedom.

By using these coupled control modes it is then possible to avoid loosing tool-part contact and also to limit the inaccurate angular positioning of the sensor head. In the meantime, whereas constraint on the sensor head are limited, this one is kept normal to the surface with a controlled (or known) orientation and position.

If the NDT sensor has no privileged direction then the NDT task is a 5 DOF one (one free rotation) otherwise the NDT task is a 6 DOF one.

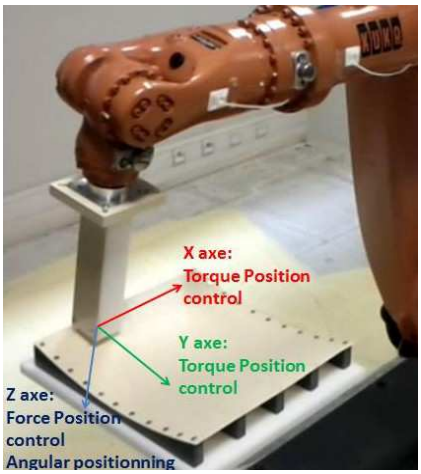

Fig. 3: Force-Torque /Position-orientation control.

3.2 Modeling robot

The classical dynamic model of a robot manipulator can be written in the Lagrangian form [5] but we need to add the external forces due to the robot-environment interaction. So robot model will be represented by equation 1.

$$\Gamma = M(q)\ddot{q} + C(q,\ddot{q})\dot{q} + G(q) + J^T F_{ext} \qquad (1)$$

q representing the joint variable n-vector, Γ the vector of generalized forces acting on the robot manipulator, $M(q)$ the inertia matrix, $C(q,\dot{q})$ represents Coriolis/centripetal forces and $G(q)$ the gravity vector. The external forces F_{ext} represents the interaction forces and J is the Jacobian matrix of the effector in the world frame.

Industrial robot controllers consider only the equation without external forces. So, we have to implement equation 1 with a hybrid command to control position and force [6].

3.3 Sensor-based control

3.3.1 Force sensor

To fulfill the testing contact constraints, the force F_z and the torque T_x and T_y have to be controlled. Consequently, we use a 6 DOF force/torque sensor. This sensor is mounted on the robot wrist between the sensor head and the manipulator.

3.3.2 External hybrid control

On industrial robot, because of warranty limitations, torques/forces must be controlled through an

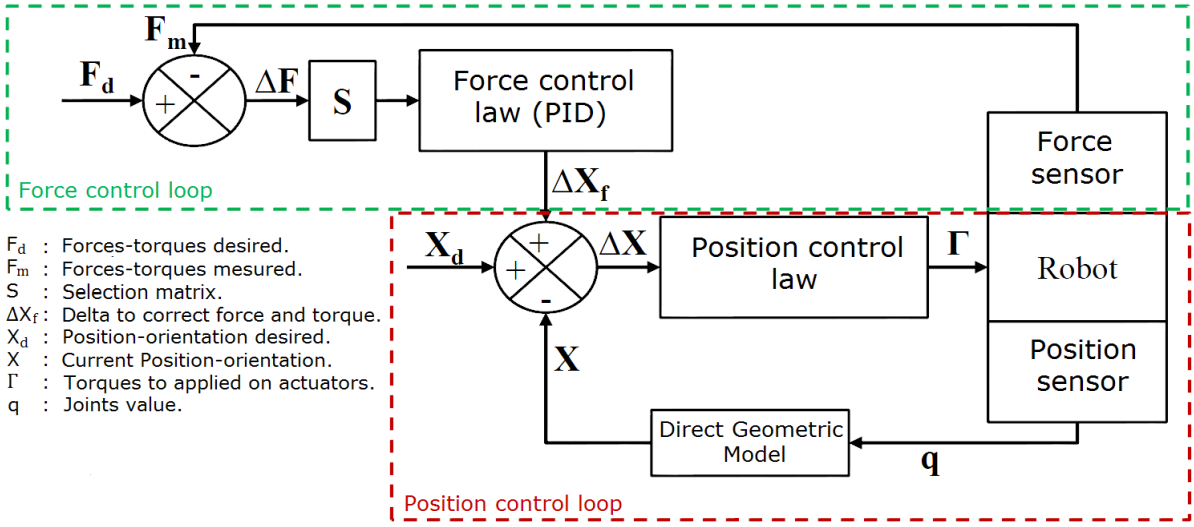

Fig. 4: External hybrid force control (green area).

external loop (green area on Fig. 4). In this way, the tool-path can be followed (industrial position control loop) while applying a controlled torque/force on a chosen tool axis (external force control loop) [4]. These two synchronized control loops make the hybrid control (force/position).

4 Robot trajectory from 3D surface

We describe here the process for generating the robot trajectory from control points based on the surface.

4.1 Robot trajectory

The robot trajectory for the NDT testing is splitted into several elementary movements. Each movement is defined using a target point P_{target} (5 or 6 DOF definition) and a displacement nature: joint space, Cartesian space (linear, circular...).

From the surface definition, we can compute the target point P_{target}. For our example the NDT rules say that:

- sensor should be at a distance d_{NDT} from a surface point ($P_{Surface}$),
- sensor should be normal (perpendicular) to the surface ($N_{Surface}$),

then P_{target} is computed as follow (eq. 2):

$$\begin{bmatrix} X_{P_{Target}} \\ Y_{P_{Target}} \\ Z_{P_{Target}} \end{bmatrix}_{R_{Sensor}} = \begin{bmatrix} X_{P_{Surface}} \\ Y_{P_{Surface}} \\ Z_{P_{Surface}} \end{bmatrix}_{R_{Sensor}} + d_{NDT} \cdot \begin{bmatrix} X_{N_{Surface}} \\ Y_{N_{Surface}} \\ Z_{N_{Surface}} \end{bmatrix}_{R_{Sensor}} \quad (2)$$

Thus, the robot program can be automatically generated from the robot trajectory control points.

4.2 Local 3D-Surface approximation

As the real part shape could differ from the CAD when attempting to match them because of the acquisition process (measurement accuracy, noise...), a surface approximation procedure must be used to customize the off line computed theoretical trajectory. We use a sensor providing us a set of 3D colored points. If a background is present then the points cloud is partitioned into two subsets: the object and the background using a classical computer vision approach (Fig. 5).

Fig. 5: Original shape to approximate

Because of the acquisition process, the data cloud is not dense and some holes need to be filled to have a exploitable view of the surface (Fig. 6). With this constraints, it's difficult to reconstruct a surface with classical methods needing control points like the NURBS or others. So, we propose a neural network approximation [7] which allows also surface characterization (like surface normal calculation).

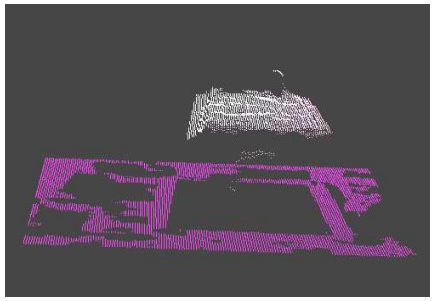

Fig. 6: 3D points cloud.

The neural network architecture is linked to the shape of the surface to be approximated (this type of problem is called "regression"). In our testing on large parts, the surface curvatures are usually large (i.e. radius above 1m). The neural net inputs are the x and y coordinates of the 3D points in the sensor frame and the output is the z coordinate. The coordinate system can be changed to any other frame like the robotic cell one.

Due to our problematic, we chose a feed forward (FF) neural network type, i.e. neurons are connected in the sense oriented from inputs to outputs (the neural architecture is shown in figure 7). Such a structure have a stable behavior and a good fault tolerance. It exists other powerful neural networks as RBF (Radial Basis Function [8]) or ANFIS (Adaptive-Network-based Fuzzy Inference Systems [9]), but they are computationally slow.

For our feed forward network, the architecture chosen is:

- 2 hidden layers with 10 neurons and tan-sigmoïd transfer function,
- 1 output layer with 1 neuron and pure linear transfer function.

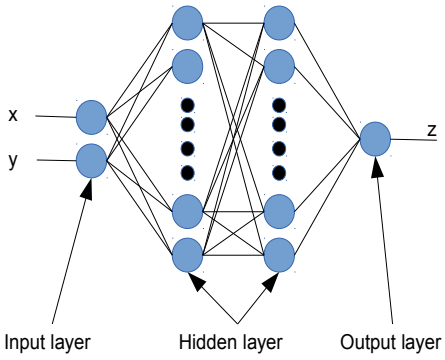

Fig. 7: Neural Feed Forward Network architecture.

In order to make the neural more independent from the training data set, we chose to scale the input $[X_{Net}\, Y_{Net}]$ and output Z_{Net} to -1 to +1 range (eq. 3) where Max and Min means the range of the real value $[X_{Real}\, Y_{Real}\, Z_{Real}]$.

$$X_{net} = 2\frac{X_{Real} - X_{min}}{X_{Max} - X_{Min}} - 1$$
$$Y_{net} = 2\frac{Y_{Real} - Y_{min}}{Y_{Max} - Y_{Min}} - 1 \qquad (3)$$
$$Z_{net} = 2\frac{Z_{Real} - Z_{min}}{Z_{Max} - Z_{Min}} - 1$$

The neural network is trained using Levenberg-Marquardt method (LM) and Mean Square Error (MSE) function as cost function to be optimized. This training method has been chosen because of its performances in our context.

The depth image size is 640×480 pixels but we only use the data from a Region Of Interest (sub-image) where the part is located. Even in the ROI, we do a sub-sampling in order to extract our date training set.

The figure 8 shows the evolution of the cost function with respect to the training epochs with 1400 points data set. The MSE goal is generally achieved in less than 30 training (epoch). Due to a fast processing, this approach can be used in an on-line robot trajectory generation.

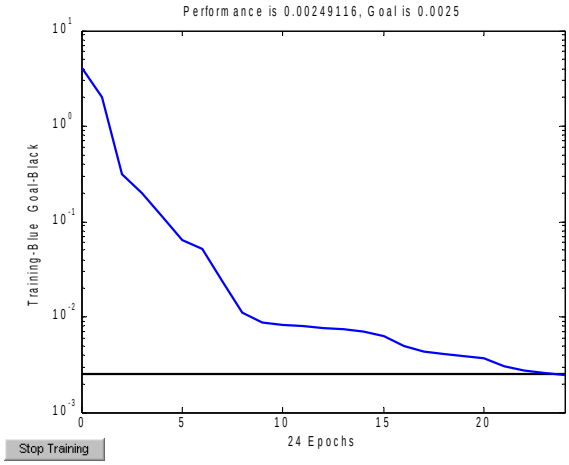

Fig. 8: Fast Neural network Training (goal met in less than 30 epochs).

4.3 Local Normal calculation

To compute the robot trajectory control points, we need to obtain the surface normal to fulfill the NDT task constraints. The surface normal at a given point can be calculated using the neighborhood of the point and a fitting plane method based on the maximum of likelihood for example (Fig. 9).

The fitting plane function calculates a least squares fit to the normal N to a plane through a set of points with coordinates $[x\, y\, z]$ in the form equation 4.

$$N_x \cdot x + N_y \cdot y + N_z \cdot z = D \qquad (4)$$

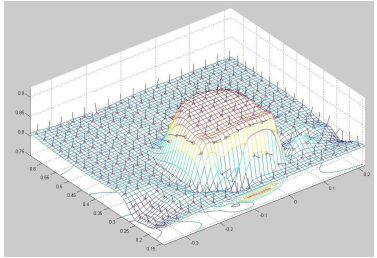

Fig. 9: 3D surface approximation with normal vectors.

Normally N is normalized so that D = 1 unless it is close to zero (a plane goes near coordinate system origin). The neighborhood can be limited to the 8 nearest points (3×3 patch centered on P_i) or extended to 5×5 patch (24 neighbors).

5 Experimental setup and results

In this paragraph, we firstly present the elementary steps from the data acquisition to the robot trajectory generation. Secondly, we show the results on our robotics equipments.

5.1 Elementary procedures

In this experiment, we use a stereo camera (bumblebee camera from Point Grey) or the kinect (from Microsoft) to capture the 3D points. To demonstrate the surface approximation reconstruction with the network, we chose a non favorable case study: big curvature and presence of specular reflexion (Fig. 10).

Fig. 10: Carbon part.

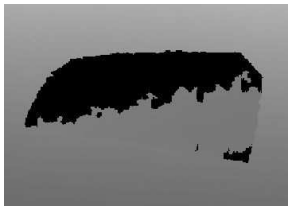

Fig. 11: Depth Image from sensor data.

The figure 11 shows the depth image computed from the sensor information. The distances are represented as gray level pixels and black pixels correspond to unknown distance.

We show the data processing and the neural network approximation. Several surface profiles are used in order to cope with the panel shapes.

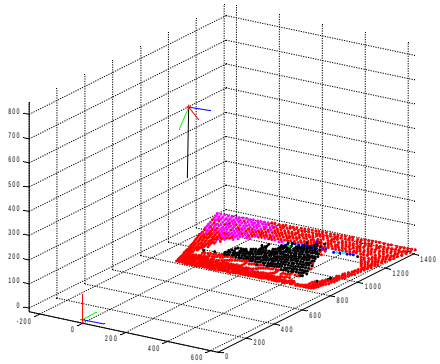

Fig. 12: 3D view of the 3D colored points cloud and sensor frame.

The 3D points cloud (Fig. 12) is used to train the neural network and after learning an approximation cloud can be computed even for inputs not included in the training set (Fig. 13).

The network allows the calculation of the Z coordinate for a 2D point $[X\,Y]$ even if the network has not been trained on this point.

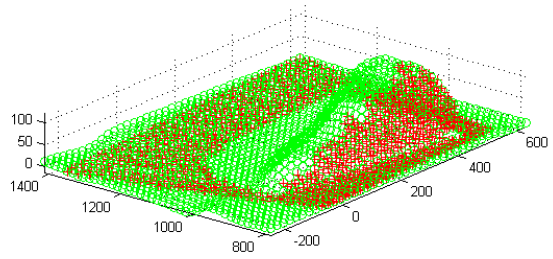

Fig. 13: Surface approximation (in red cross training set and green approximated points).

Using the point neighborhood, the surface normal can be computed (Fig. 14).

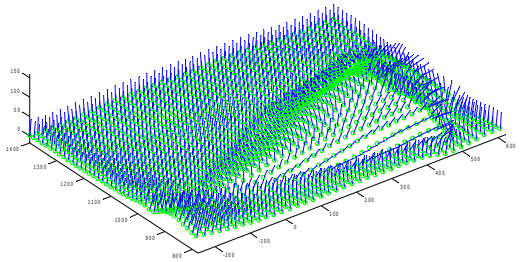

Fig. 14: Normal computation from NNet surface approximation.

Using the NDT rules and the Neural net results, the robot position and tool orientation can be calculated. Thus, the robot trajectory control points and Z_{tool} axis are obtained. Positions (respectively Z_{tool}) are shown as red circles (respectively as red line) on figure 15.

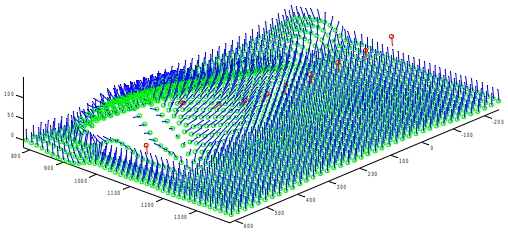

Fig. 15: Robot trajectory control points (in red) with tool Z axis.

The figure 16 shows a zoom-in of the trajectory control points.

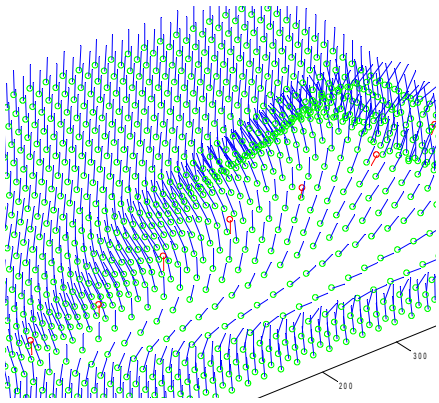

Fig. 16: Zoom in on robot control points (in red).

5.2 Experiments on the robot cell

The robotic cell of ESTIA is used for testing. It consists of a KUKA KR6 robot, a NDT sensor support and molds with different curvatures. RSI module allows communication between Sensor-PC and the robot (Fig. 17).

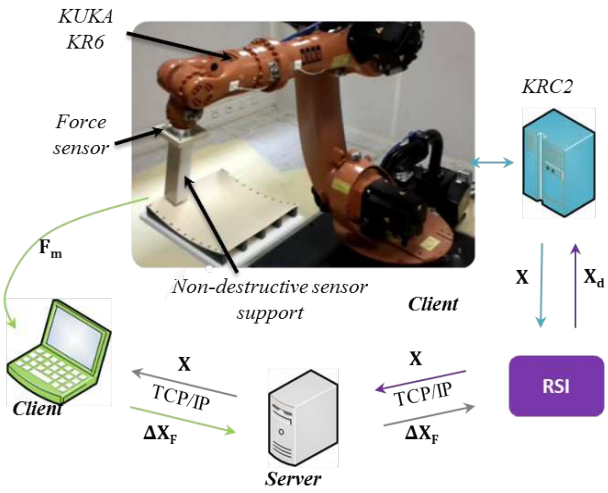

Fig. 17: Robot cell for NDT.

In this test, the ND task is implemented using the force-position approach. The additional external loop is based on modeling the behavior of a spring. The data logging is done to check if the NDT constraints are respected.

We firstly present the results on the 3D surface approximation and how this approach suits the industrial cases when the part flexibility is high (part deformation) or when the surface is unfoldable or even unknown.

Secondly, we show how the hybrid-control scheme allows us to achieve the constraints of the NDT and is helpful to carry on automated testing.

5.2.1 Aeronautical part approximation

For our experiment, we use a landing gear door panel. We have in our workshop the mold (Fig.18 and 19) and a test panel made using our robotic fiber placement machine.

Fig. 18: Robot and landing gear door mold.

Fig. 19: Automated Fiber placement.

Due to part dimensions and the 3D device, the acquisition device could not make the measurement in one shot.

The point-clouds are merged using a dedicated software for 3D measurement (Fig. 20).

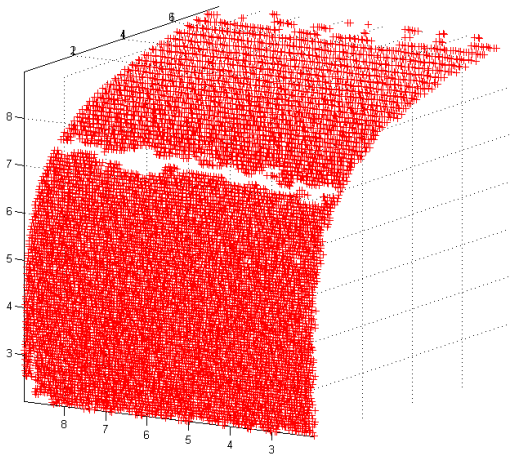

Fig. 20: 3D points cloud from merged acquisitions.

Using these 3D points as entry for our network learning, we manage to meet the precision goal. The figure 21 presents the evolution of the MSE cost function during the training.

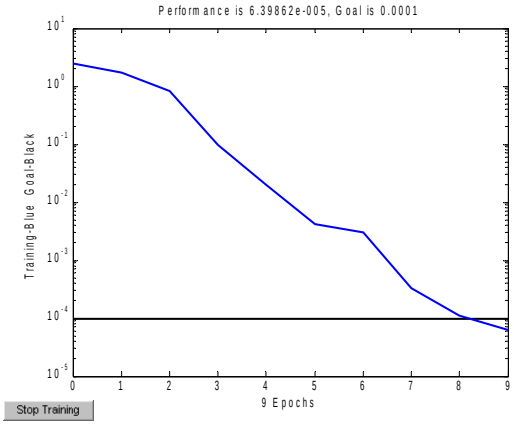

Fig. 21: MSE cost function evolution during learning phase (only 9 epochs to reach the goal).

The trained neural network can now be used as an approximation black box for the trajectory module. The following figure (Fig. 22) shows the approximated surface of the door panel (the step size on X or Y can be changed).
The holes in the original data sample are not present because of the approximation of the Z component by neural network for a given $[X\ Y]$.

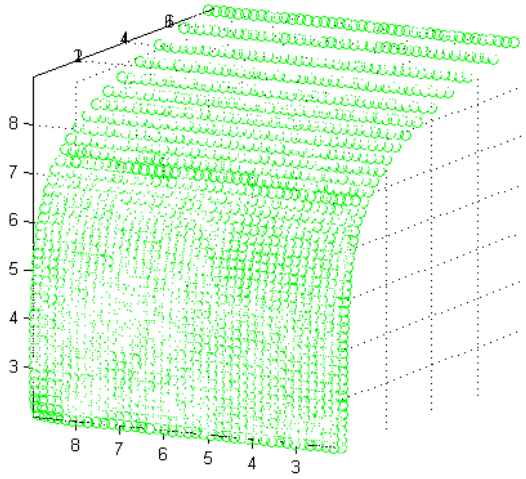

Fig. 22: Neural network surface approximation.

The (Fig. 23) shows this approximate surface over the holed surface.
We are now able to chose a NDT direction (for a given X and an Y range) by computing the robot trajectory control points and the tool Z axis (Fig. 24).

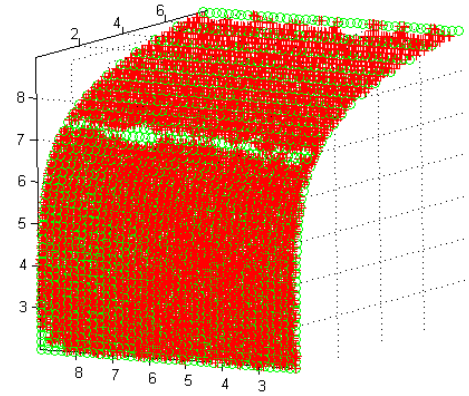

Fig. 23: Neural network surface approximation (green) over 3D points cloud (red).

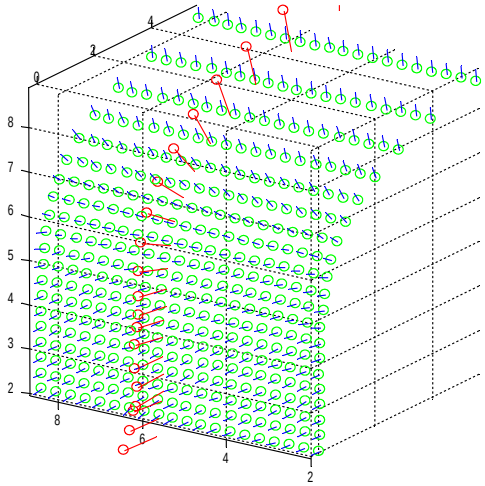

Fig. 24: Robot trajectory generation with control points (red).

5.2.2 Hybrid control

In order to do the robotic experiment, we have scaled down the part size to suit the ESTIA robotic cell. We use a fiber glass plate where we have include known and visible default (blue patch on figure 25). In this experiment the position on the plate is known (using the teaching facility of the robotics environment) but the orientation is not calibrated.

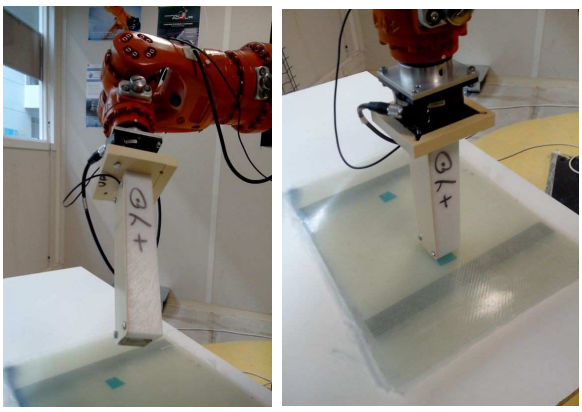

Fig. 25: ESTIA robotic cell and panel.

We compute the cardinal spline based on the control points [10] and this path (Fig. 26 and 27) is split into three phases (sampled into elementary segments according to the desired maximal velocity and the RSI loop frequency):

1. approach in position control,
2. NDT task in force control,
3. departure in position control.

The robot movement on each segment is a linear approximation. During this displacement, the force and torque linked to the contact constraints are controlled by the external hybrid-control.

On our experimental setup, we have logged the contact force (Fz) during the three phases. During the NDT phase, the contact force follows our desired value of 3N. The result is presented on figure 26.

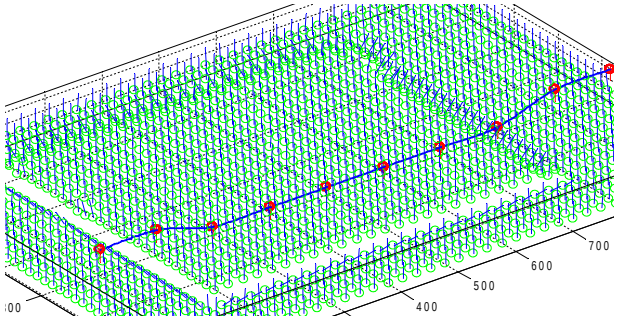

Fig. 27: Robot trajectory based on Cardinal spline.

6 Conclusions and future works

To complete NDT task, we need to control the contact forces and torques and we are facing the industrial constraints (the use of industrial robot controller and components). We propose to implement an external hybrid command. The robot trajectory is locally generated by a surface approximation using a neural network.

We have presented the computer sciences approach from 3D points acquisition to surface approximation on a test part and an industrial one. We have also presented the robot trajectory computation based on geometrical data (position on the surface and being normal to the surface) and the robot path passing through the control points. All this processing has been tested on our robotics facility on a simple case (plate).

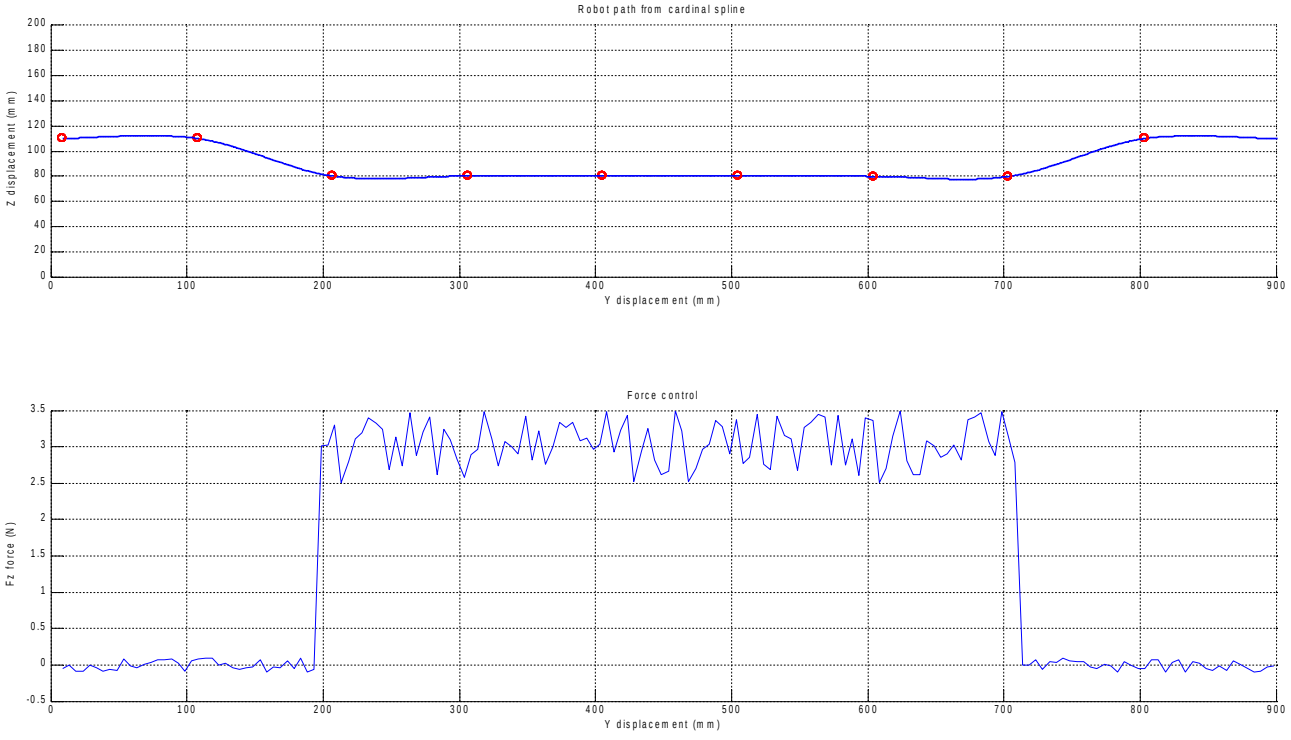

Fig. 26: Monitored contact force vs robot trajectory.

As future work, the surface approximation need to be ported to "real time" computing system and robot-environment model improved to be more realistic.

As we bought a new robotic cell for cutting and edging composites material panels (based on the KR240 KUKA robot and a KRC4 controller), we have planed to migrate the experiment as the RSI loop period is 4ms on this new device compared to 12ms on the current one. This is an easy 3 times factor gain which will permit a better force-control and a velocity improvement. We also need to perform the coupling of the NDT equipment on the external PC to be able to store synchronized data and being able to display NDT 3D maps.

7 References

[1] R.Oster "Non-destructive testing methodologies on helicopter fiber composite components challenges today and in the future", *18th World Conference on Nondestructive Testing*, 16-20 April 2012, Durban, South Africa.

[2] R. Bogue "The role of robotics in non-destructive testing", *The Industrial Robot*, 2010, 37(5), 421-426.

[3] T. P. Sattar and A.-A. Brenner "Robotic system for inspection of test objects with unknown geometry using NDT methods", *The Industrial Robot*, 2009, 36(4), 340-343.

[4] D. Xiao, B.K. Ghosh, X. Ning, T.J. Tarn "Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment", *Control Systems Technology*, IEEE Transactions, vol.8, no.4, pp.635-645, Jul 2000.

[5] M. M. Fateh "Dynamic Modeling of Robot Manipulators in D-H Frames", W*orld Applied Sciences Journal* 6 (1): 39-44, 2009 ISSN 1818-4952, IDOSI Publications 2009.

[6] J. De Schutter, H. Van Brussel "Compliant Robot Motion II. A Control Approach Based on External Control Loops". *The International Journal of Robotics Research*, 7(4): 18–33, August 1988.

[7] J. Barhak, A. Fischer "Adaptive reconstruction of freeform objects with 3D SOM neural network grids", *Computer Graphics and Applications*, 2001. Proceedings. Ninth Pacific Conference on Computer Graphics and Applications, vol., no., pp.97,105, 2001.

[8] M. Qinggang, B. Li, H. Holstein, Y. Liu "Parameterization of point-cloud freeform surfaces using adaptive sequential learning RBFnetworks", *Pattern Recognition*, Volume 46, Issue 8, August 2013, Pages 2361-2375.

[9] J.S. Jang "ANFIS: Adaptive-Network-based Fuzzy Inference Systems", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 23, No. 3, pp. 665-685, May 1993.

[10] B. Cao, G.I. Dodds, G.W. Irwin "Constrained time-efficient and smooth cubic spline trajectory generation for industrial robots", *Control Theory and Applications, IEE Proceedings*, vol.144, no.5, pp.467,475, September 1997.