QNDE2019-1234

A TOOLBOX FOR ULTRASOUND SIGNAL PROCESSING AND IMAGE RECONSTRUCTION

Thiago A. R. Passarin,[*] **Hector L. de Moura,
Vitor O. Silva**, **Filipe B. G. Cândido**,
Karoline A. Sato, **Andréia R. de Souza**,
Marco T. L. Guerreiro, **Giovanni A. Guarneri**,
Daniel R. Pipa
Department of Electrical Engineering
Federal University of Technology - Paraná - Brazil
Curitiba, PR 80230-901
Email: passarin@utfpr.edu.br

Gustavo P. Pires
Petrobras Center for
Research and Development (CENPES)
Rio de Janeiro, RJ
Brazil

ABSTRACT

We present a toolbox for ultrasound signal processing and image reconstruction whose main goal is the detection and mapping of internal corrosion in underwater oil pipelines. The software is developed in Python and is divided in modules for (i) graphical interface, (ii) inspection file reading, (iii) signal pre-processing, (iv) inspection parameters estimation (v) external surface detection (for immersion inspection) and (vi) imaging. This paper describes the development of key features and indicates the chosen implementation of algorithms. The current state of the development is presented and the next steps towards the final goal of corrosion mapping are set.

1 INTRODUCTION

Detection and mapping of corrosion is an important and challenging aspect in the maintenance of oil production systems. Corrosion of the internal surface occurs in pipelines as a consequence of mechanical and chemical phenomena such as abrasion and hydrogen attack. Knowledge about the dimensions of the corrosion allows technicians to decide whether a pipe is or is not structurally stable under the operation conditions. A nondestruc-tive way to assess the state of internal surfaces is through ultrasound array imaging. The following sections describe a toolbox developed to perform image reconstruction from array ultrasound inspection data. Section 2 lists the tools and software libraries used in the development of the toolbox. The methods for graphical interface, reading of inspection files, signal pre-processing, estimation of inspection parameters, external surface detection and image reconstruction are described in section 3. Finally, section 4 contains a brief conclusion and directions for the next steps on the development of the toolbox.

2 TECHNOLOGIES USED

We chose Python 3 as the programming language for the development of the toolbox. This language has shown good results dealing with algebraic and matrix calculations with packages such as NumPy and SciPy, which are used in the algorithms for parameter estimation and imaging. In addition, the Numba compiler is utilized to optimize the code for better performance. The graphical interface was built with PyQt5 and PyQtGraph, a PyQt package with useful functionalities for data visualization.

---

[*]Address all correspondence to this author.

3 DEVELOPMENTS

3.1 Inspection Data Reading

The toolbox is capable of importing inspection data from different sources and casting it to a default structure. Currently, the toolbox supports importing data from the Multix++ and OmniScan inspection systems, as well as from the CIVA simulator.

CIVA is an NDT simulator developed by CEA and partners [1]. Simulation files are divided in subdirectories, which contain the inspection parameters in `xml` files while raw A-scan data are stored in binary files. The `xml` files are read with dedicated libraries and the binary files are first decompressed and subsequently decoded to `float32` format. The toolbox is able to read simulation files and extract inspection, specimen and transducer parameters.

Multix++ is a commercial inspection system from M2M, capable of performing immersion and contact tests, in sweep or full-matrix capture configurations [2]. Inspection parameters are also stored in `xml` files, while raw A-scan data is saved in binary files. The binary files also provide the sound velocity in the material and in the coupling medium.

OmniScan MX2 is a commercial inspection system designed by Olympus, capable of performing sweeping inspections [3]. Inspection data can be read with the NDT Data Acess Library, which is a dedicated software with programmed functions to read data acquired with Olympus instruments. The toolbox has a plugin that uses this library to read inspection data, which can extract inspection and raw A-scan data.

3.2 Pre-processing Options

The toolbox has built-in pre-processing options like obtaining the mean value of multiple A-scans for noise reduction, mean removal for a single A-scan and an option to add Gaussian noise to evaluate algorithm performance. Additional pre-processing options are in development and the user is also able to make custom codes for pre-processing.

3.3 Inspection Parameter Estimation

So far the Toolbox is able to estimate the sound velocity in water and specimen under inspection, the surface profile and the probe delay. Some of these parameters require specific conditions while others make no assumption on the inspection.

3.4 External Surface Detection

The imaging algorithms described in section 3.5 are based on the calculation of the time of flight between each transducer element and each pixel inside the Region of Interest (ROI). Refraction on the water-steel interface must be taken into account in order to successfully trace the ray path between each element-pixel pair. Consequently, the entry point of each ray on the interface must be calculated. This calculation is performed with a discrete version of the Newton-Raphson algorithm proposed in [4] for this specific problem. From a collection of discrete points on the surface, the algorithm determines which point best approximates Snell's law for a given element-pixel pair, then selects that point as ray entry point. The discrete points are taken from an analytic description of the surface obtained from fitting a circle and line to the B-scan data, for cylindrical and plane surfaces respectively. The fitting procedures use interface points identified by the pulse-echo method [5].

3.5 Imaging Algorithms

The imaging algorithms aims are to provide a graphical representation of the internal acoustic reflexivity of an object from the A-scan signals captured by a non-destructive ultrasonic inspection system. Currently, we have implemented in the toolbox four algorithms widely used by the non-destructive testing community: SAFT, ωk-SAFT [6], TFM [7] and the *wavenumber* [8].

The SAFT and TFM algorithms process the A-scan signals directly in the time-domain, and both are based on the *delay-and-sum* (DAS) operations. In our implementation, DAS operations are carried out for each ROI point using an optimized loop generated by the Numba tool, using the parallelism feature [9].

The ωk-SAFT and *wavenumber* algorithms treat the A-scan signals in the frequency-domain which leads to better performance than SAFT and TFM when reconstructing large images [8]. Additionally, the typical implementations of these algorithms assume that the grids of the reconstructed image and the acquired data are the same. Our implementation allows these grids to be different. This is possible by interpolating the spectral points of the image into the acquired data spectrum.

3.6 Interface

On the toolbox, the user can write their own scripts and execute them using the modules functions, or use the Graphical User Interface (GUI) developed, which can be seen in Fig. 1.

The interface contains the following functionalities presented in the toolbox: (i) loading a file, (ii) pre-processing the data, (iii) selecting the desired ROI, estimating the sound speed in the material, (iv) applying an imaging algorithm, (v) detecting the surface in immersion inspections, (vi) post-processing the image and (vii) visualizing data slices, A-scans and generated images.

A standard for the function call was developed so the interface can identify and use new imaging algorithms, allowing the user to create its own code and test it with the aid of the GUI. The same can be done with pre-processing algorithms.

Fig. 1 shows an example of visualization, where a slice of the raw data is shown in the top-left image, an A-scan is shown in the bottom-left image and the result of a TFM is shown in the right image. Notice the red line denoting the detected flat surface.
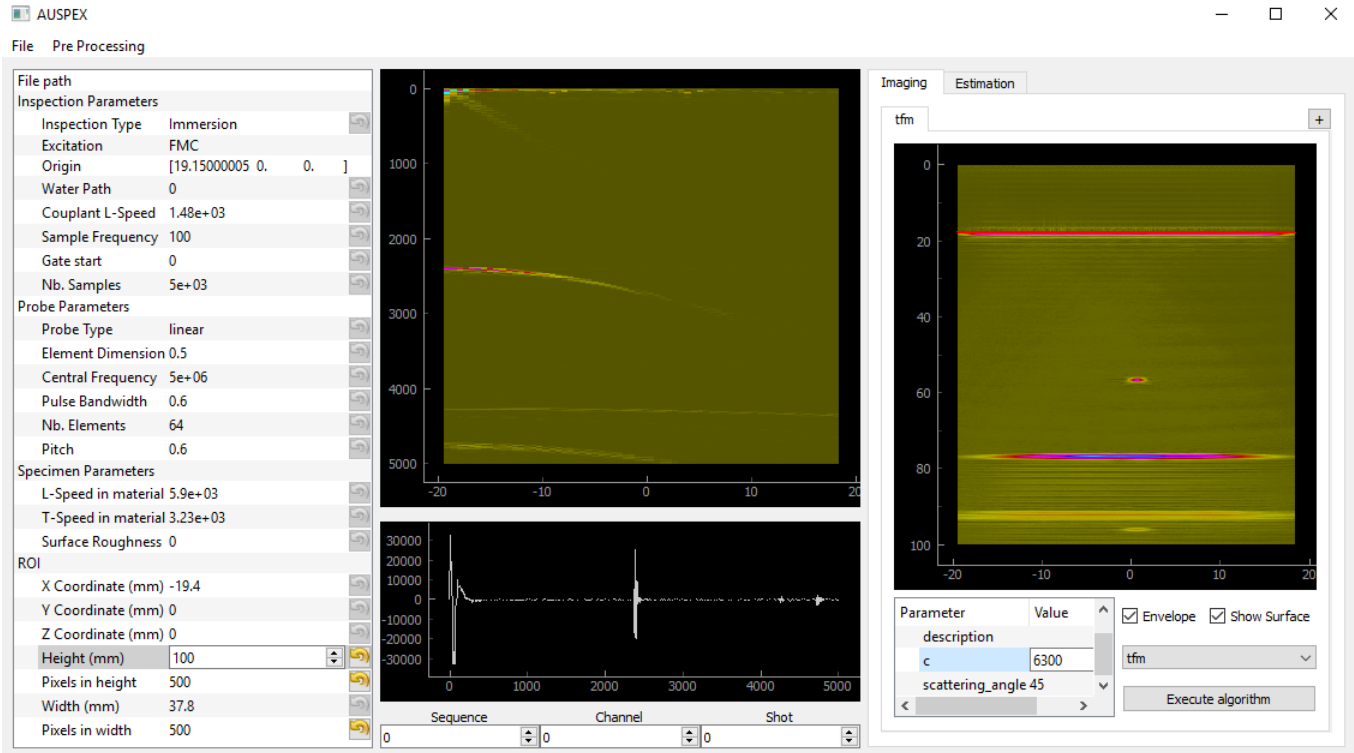
Copyright © 2019 by ASME

FIGURE 1. The developed user interface.

4 CONCLUSION AND FUTURE DEVELOPMENTS

The toolbox can successfully read data from state-of-the-art equipment and simulator and reconstruct images with the algorithms listed in section 3.5. Additionally, users can plug their own implementations of those and other imaging algorithms.

The next functionality to be implemented in the toolbox is the representation of the internal corroded surface (e.g. with marching cubes). Another functionality to be studied and implemented is the possibility to save the generated images with the used parameters and pre-processing, then loading the created files.

ACKNOWLEDGMENT

REFERENCES

[1] TECH, C., 2017. *CIVA NDE 2017 User's Manual*. CEA TECH, www-civa.cea.fr/.

[2] NDT, M., 2019. *M2M NDT*. M2M NDT, www.m2m-ndt.com.

[3] Olympus NDT, I., 2012. *NDT Data Access Library User's Manual*. Olympus NDT, Inc., www.olympus-ims.com, 10.

[4] Parrilla, M., Ibáñez, A., Camacho, J., and Fritsch, C., 2007. "Fast focal law computing for non-destructive testing with phased arrays". In International Congress on Ultrasonics, pp. 1–4.

[5] Camacho, J., Cruza, J. F., Brizuela, J., and Fritsch, C., 2014. "Automatic dynamic depth focusing for ndt". *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control,* **61**(4), pp. 673–684.

[6] Stepinski, T., 2007. "An implementation of synthetic aperture focusing technique in frequency domain". *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on,* **54**(7), pp. 1399–1408.

[7] Holmes, C., Drinkwater, B. W., and Wilcox, P. D., 2005. "Post-processing of the full matrix of ultrasonic transmit–receive array data for non-destructive evaluation". *NDT & E International,* **38**(8), pp. 701–711.

[8] Hunter, A. J., Drinkwater, B. W., and Wilcox, P. D., 2008. "The wavenumber algorithm for full-matrix imaging using an ultrasonic array". *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on,* **55**(11), pp. 2450–2462.

[9] Doglio, F., 2015. *Mastering Python High Performance*. Packt Publishing.