Characterization of the fluctuations of an ultrasonic wave passing through a complex environment in order to simplify the modeling

Mirella Aoun [1], Clément Berger [2], Jean Cazalis [3], Thierry Gonon [4], David Lassounon [5]

[1] LMRS, CNRS, UMR 6085, University of Rouen, Avenue de l'Université, 76801 Saint Etienne du Rouvray, France, mirella.aoun@univ-rouen.fr
[2] UMPA, CNRS, UMR 5669, ENS de Lyon, 69364 Lyon, France, clement.berger@ens-lyon.fr
[3] CEREMADE, CNRS, UMR 7534, Université Paris-Dauphine, PSL University, 75016 Paris, France, cazalis@ceremade.dauphine.fr
[4] ICJ, UMR 5208, Ecole Centrale de Lyon, Ecully, France, thierry.gonon@ec-lyon.fr
[5] IRMAR-INSA Rennes, CNRS, UMR 6625, 35708 Rennes, France, enagnon-david.lassounon@insa-rennes.fr

Abstract

Ultrasonic nondestructive testing, also called ultrasonic NDT, is a method for characterizing the thickness or internal structure of material using high-frequency acoustic waves. This technique consists in emitting an ultrasound within the object and detecting the echoes produced by the possible defects. In complex environments, in particular for certain metals, composite materials and concretes, ultrasonic waves are strongly disturbed by the material through which they propagate, which can distort measurements. Modeling these phenomena can help designing control procedures and improving their signal to noise ratio. Because simulation using finite-element methods can be computationally prohibitive, it is important to develop alternative and lighter approaches. In this report, we model these fluctuations in the same manner as dynamic speckle sequences, using stochastic processes. These methods have parameters that we fit with respect to the simulations generated by the CEA.

Keywords: **ultrasound, Gaussian processes, kriging, speckles, noise, wave fluctuations**

Contents

1 Introduction **2**
1.1 Ultrasonic testing 2
1.2 Our framework 3

2 Gaussian processes based approach **4**
2.1 Initial model 4
2.2 Refinements 5
2.3 Drawbacks 7

3 Dynamic simulation with smoothing **7**
3.1 A first implementation 7
3.2 Modifications to enhance temporal behavior 8

4 Quantitative criteria **9**
4.1 Dissimilarity measures 10
4.2 Parameter optimization 11

5 Conclusion and further topics of research **13**

1 Introduction

This document summarizes a work that has been realized in four days for the CEA (Commissariat à l'énergie atomique et aux énergies alternatives) during the SEME (Semaine d'Etudes Mathématiques-Entreprises) organized by IP Paris (Institut Polytechnique de Paris) and AMIES (Agence pour les Mathématiques en Interaction avec l'Entreprise et la Société) in February 2022. We would like to thank our CEA supervisors, Vincent Dorval and Edouard Demaldent, for their support during this week. We also thank Samuel Amstutz as well as the whole staff of the SEME for providing us with this opportunity.

1.1 Ultrasonic testing

Many efforts are being made by the electric power industry in order to ensure security and to extend the life of electric power stations such as nuclear plants or hydroelectric dams. For this purpose, it is crucial to improve the detection of defects (inclusions, blisters, defects in sticking, etc.) or damages in these structures. However tests must not affect the integrity of the materials and thus require the use of so-called nondestructive testing methods.

Among these methods, ultrasonic testing is frequently used because it has many advantages: ease of implementation, possibility of working on a single side of the part to be inspected (no need for access to the second side), and the ability to pass through large thicknesses of material depending on the working frequency. Moreover, the existence of relationships between the propagation of ultrasound and the characteristics of the material allows its characterization. Therefore, the use of ultrasound offers the possibility, without any deterioration, on the one hand of characterizing materials in order to know their elastic properties, and on the other hand of inspecting parts to check their integrity. A challenge of this method comes from the heterogeneity of the materials which are being put to the test. Typically, concrete walls contain multiple aggregates at random places (see figure 1) which disturb the propagation of the ultrasonic waves.

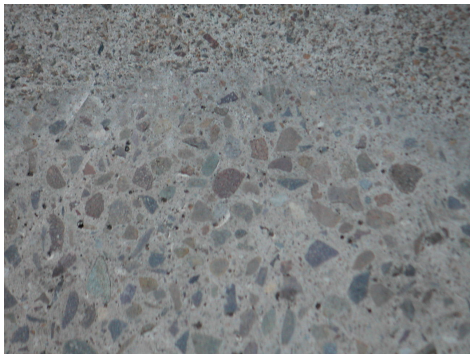


Figure 1: Surface of concrete showing multiple aggregates (image from wikipedia.org).

Numerical simulations of wave propagation through complex materials by finite-element methods are introduced by the ATHENA code developed by EDF and INRIA. ATHENA simulates the propagation of ultrasonic waves in a plane of symmetry of heterogeneous complex anisotropic materials with applications on stainless welds [1, 2, 5]. The CEA uses a more recent code developed in 2019 by Imperiale and Demaldent [4]. However these simulations require the exact positions of the heterogeneities, which are in general not available. In addition, these methods are computationally expensive which prevents their use in a large scale. A way out consists in focusing on the averaged ultrasonic field, which makes it possible to reduce the problem to the easier case of a homogeneous material. Nevertheless, because this approach does not take the noise induced by the heterogeneities into account, it is not precise enough for industrial purposes. One could improve the procedure by modeling the fluctuations around the average wave.

1.2 Our framework

In this work, we propose to randomly simulate wave profiles which are similar to simulations given by a finite-element based solver.

The simulations are performed on a 2D rectangular grid. We will denote space coordinates by x and z and the time by t. We follow a planar wave $u(x, z, t) \in \mathbb{R}^2$ evolving along the z axis, parallel to the x axis. This work is dedicated to the simulation of the deformation along the z axis, denoted by u_z. There are 100 points in z and 200 points in x. Because the finite-element solver has an adaptative time discretization, the number of time steps depends on the simulation. There are roughly between 100 and 200 time steps in the data.

As mentioned earlier, the wave averaged along the x axis can be easily simulated. Thus we will focus only on the remaining fluctuations caused by the aggregates. Finally, at each time t, the space can be roughly divided into three different areas. Starting from the top, we have a zone with $u_z = 0$ since the wave has not reached this zone yet. Getting down we meet the front wave, corresponding to the wave we would have if the material was homogeneous. Close to this area, the fluctuations are composed of many small disorganized patches. Finally, under this area lies the so-called *coda*, which corresponds to the residues of the wave. This setup is described in Figure 2. In this work, it is assumed that the coda is the main contributor to the noise measured during the testing procedures. As a consequence,

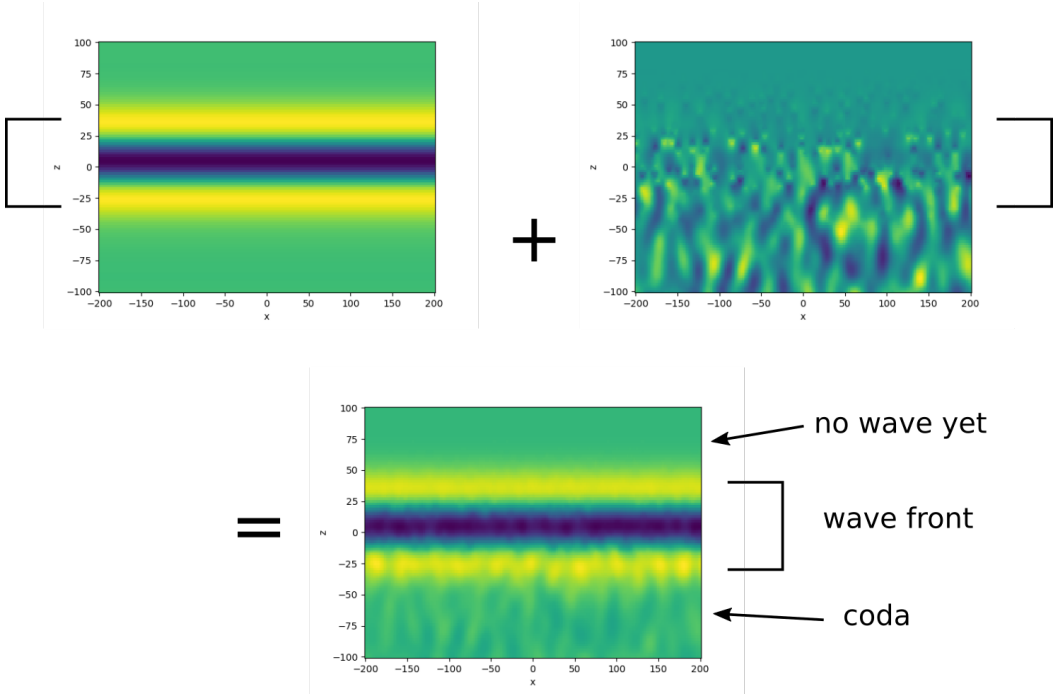


Figure 2: Scheme of the setup. The two upper images show the decomposition of the wave between its average and the noise resulting from the aggregates. The decomposition of the space between the three areas is also presented.

we mainly focus on its modeling.

This paper is presented as follows: in Section 2, we present a first method based on Gaussian processes. Another method based on a dynamic update is presented in Section 3. Finally we propose different indicators in Section 4 in order to get quantitative evaluations of the performances of the methods proposed in this report.

2 Gaussian processes based approach

2.1 Initial model

The first method is inspired from the Gaussian process regression formalism [6, 7]. The z component of the residual fluctuations u_z is supposed to be the realization of a Gaussian process $U_z(x, z, t)$ depending on the space and time variables:

$$U_z(x, z, t) = Y(x, z, t) \cdot \mathbb{1}_{]0,+\infty[}(z - vt)\,.$$

The coda is modeled by the process Y. We have denoted by v the group velocity in the direction z of the ultrasonic wave and we compute it using the simulations provided by the CEA. The function $\mathbb{1}_{]0,+\infty[}(z - vt)$ is equal to 1 if $z - vt \in]0, +\infty[$ and 0 otherwise. In other words, we set the noise to zero in the area $\{z > vt\}$ that the wave front has not reached yet.

The process U_z is simulated by generating a path of Y and multiplying it by the Heavyside function $\mathbb{1}_{]0,+\infty[}(z - vt)$. The path of Y is generated on a grid $\mathbb{G} = \{(x_i, z_i, t_i),\ 1 \leq i \leq N\}$

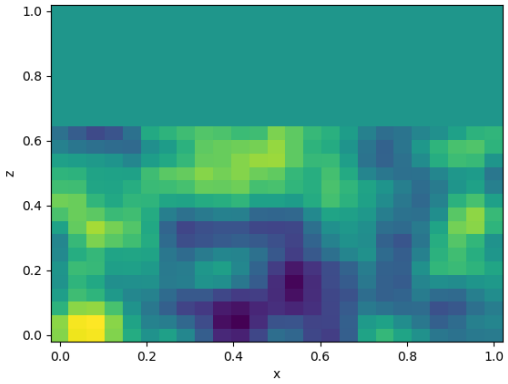


Figure 3: Example of simulation using Gaussian processes at one time step. Coding has been done in `R`.

of the input space (x, z, t). The law of Y is chosen equal to:

$$Y \sim \mathcal{GP}\left(0, k((x,z,t),(x',z',t'))\right) ,$$

where we denote by k the covariance kernel. We choose k as a Matérn kernel with parameter $\frac{5}{2}$ (a usual choice in the Gaussian regression community):

$$\begin{aligned} k((x,z,t),(x',z',t')) \;=\; & \sigma^2 \quad \left(1+\frac{\sqrt{5}|x-x'|}{\theta_x}+\frac{5(x-x')^2}{3\theta_x}\right) \quad \exp\left(-\frac{\sqrt{5}|x-x'|}{\theta_x}\right) \\ & \cdot \quad \left(1+\frac{\sqrt{5}|z-z'|}{\theta_z}+\frac{5(z-z')^2}{3\theta_z}\right) \quad \exp\left(-\frac{\sqrt{5}|z-z'|}{\theta_z}\right) \\ & \cdot \quad \left(1+\frac{\sqrt{5}|t-t'|}{\theta_t}+\frac{5(t-t')^2}{3\theta_t}\right) \quad \exp\left(-\frac{\sqrt{5}|t-t'|}{\theta_t}\right). \end{aligned}$$

The generation of the Gaussian vector $Y(\mathbb{G})$ of law:

$$Y(\mathbb{G}) \sim \mathcal{N}\left(0, k(\mathbb{G},\mathbb{G})\right) ,$$

is done by simulating a standard Gaussian vector $\epsilon \sim \mathcal{N}(0, I)$ and multiplying it by the square-root of the covariance matrix $k(\mathbb{G},\mathbb{G})$ (see Algorithm 1):

$$Y(\mathbb{G}) = \sqrt{k(\mathbb{G},\mathbb{G})} \cdot \epsilon.$$

Notice that the computation of $\sqrt{k(\mathbb{G},\mathbb{G})}$ can be done once and for all before performing multiple simulations. An illustration of a simulation is given in Figure 3, for a $25 \times 25 \times 25$ grid in $[0,1]^3$ where the parameters are arbitrarily set to $\sigma^2 = 1$, $\theta_x = \theta_z = \theta_t = 0.1$, $v = 1$.

2.2 Refinements

- To smooth the wave front, the Heavyside function used for the cutoff can be replaced by a function of the form: $\frac{1}{2} - \frac{1}{\pi}\arctan(\alpha(z - vt))$ with α a parameter to adjust.
- The model parameters may be adjusted numerically, instead of manually, by maximiz-

Algorithm 1 Simulation based on Gaussian processes.

Set $\mathbb{G}$ of size n, σ^2, θ_x, θ_z, θ_t.

Generate $\epsilon \sim \mathcal{N}(0, I_n)$.

Compute $k(\mathbb{G}, \mathbb{G}) = (k((x_i, z_i, t_i), (x_j, z_j, t_j)))_{i,j}$.

Decompose:

$$k(\mathbb{G}, \mathbb{G}) = P \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_n \end{pmatrix} P^T. \tag{2.1}$$

▷ Eigenvalue decomposition

Take the square root:

$$\sqrt{k(\mathbb{G}, \mathbb{G})} = P \begin{pmatrix} \sqrt{\lambda_1} & 0 & \dots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \sqrt{\lambda_n} \end{pmatrix} P^T. \tag{2.2}$$

Compute:

$$Y(\mathbb{G}) = \sqrt{k(\mathbb{G}, \mathbb{G})}\epsilon. \tag{2.3}$$

▷ Simulation of the noise

H vector of size $n \times 1$

for i in $\{1, \cdots, n\}$ **do** ▷ Build Heavyside vector

  if $z_i - vt_i > 0$ **then**

    $H(i) = 0$

  else

    $H(i) = 1$

  end if

end for

Final simulation:

$$U_z(\mathbb{G}) = Y(\mathbb{G})^T H. \tag{2.4}$$

ing the loglikelihood for the observations of one simulation:

$$l(\sigma^2, \theta_x, \theta_z, \theta_t) = -\log|k(\mathbb{G}, \mathbb{G})| - \frac{\boldsymbol{u_z}^T k(\mathbb{G}, \mathbb{G})^{-1} \boldsymbol{u_z}}{2}.$$

- To take into account the fact that the fluctuations may vanish with time and far from the wave front, we can multiply the simulation by $\exp(\beta(z - vt))$ with β a parameter to adjust.

2.3 Drawbacks

While the results are promising, the size of $\mathbb{G}$ makes the computations of $\sqrt{k(\mathbb{G}, \mathbb{G})}$ (for the simulations) and $k(\mathbb{G}, \mathbb{G})^{-1}$ (for fitting the parameters) computationally expensive. Truncations of the covariance matrix have been attempted in order to obtain sparser covariance matrices. However, even with these adjustments, the calculation time of $\sqrt{k(\mathbb{G}, \mathbb{G})}$ for a $25 \times 25 \times 25$ grid takes hours. In conclusion, it is not realistic to use this method for bigger grids and multiple different parameters.

3 Dynamic simulation with smoothing

In this section we build upon the work of [3] to simulate the noise. In [3], the authors present a numerical simulation to generate a temporal sequence of dynamic speckle patterns which is based on a model on a superposition of waves from discrete scattering centres.

3.1 A first implementation

Algorithm 2 Dynamic simulation of the noise with space smoothing.

Set $\phi(.,.,t=0)$, $c > 0$ decreasing, $T_{max} > 0$.
for $0 \leq t \leq T_{max} - \Delta t$ **do**
  for all x, z **do**
    Compute:

$$\phi(x, z, t + \Delta t) = \phi(x, z, t) + \mathcal{N}(0, 1)\sqrt{\log c(t + \Delta t) - \log c(t)}). \tag{3.1}$$

▷ Computation of the new phase

    Compute:

$$\tilde{u}_z(x, z, t) = \mathcal{R}\left(\mathcal{F}^{-1}(H_{\omega_0}(\mathcal{F}(e^{i\phi(.,.,t)})))(x, z, t)\right). \tag{3.2}$$

▷ Application of low pass filter

  end for
end for
Compute:

$$u_z(x, z, t) = \frac{\frac{\pi}{2} - \arctan(\alpha(z - vt))}{\pi} \tilde{u}_z(x, z, t). \tag{3.3}$$

▷ Application of arctan to recreate the no-wave area

Let's briefly describe the algorithm proposed in [3]. The idea is to work on the phase ϕ of the wave, with a time iterative update. We start with $\phi(t=0)$ given by sampling uniform (over $[-\pi, \pi]$) random variables on the space grid. Then at each time, we update the phase according to the formula (3.1) in which the phase at each time consists in the previous phase shifted by a centered random variable. The quantity $\sqrt{\log c(t+\Delta t) - \log c(t)}$ allows us to control the variance of the update, where the time function c represents the correlation between the image at time t and the image at time 0. It is calibrated according to the CEA's simulations using the formula:

$$c(t) = \frac{\mathbb{E}[I(0)I(t)] - \mathbb{E}[I(t)]\mathbb{E}[I(0)]}{\sqrt{\mathbb{V}[I(0)]\mathbb{V}[I(t)]}}\,, \tag{3.4}$$

where $\mathbb{E}$ (resp. $\mathbb{V}$) denotes the mean operator (resp. the variance operator).

The resulting image is a white noise evolving with time, with a degree of time correlation. To add spatial correlation, a low pass filter is applied on each image, see (3.2). More specifically, we perform a discrete Fourier transform on the image at each time step then we apply a cutoff function H_{ω_0} on both space variables: given a threshold ω_0 in the frequency domain, for each frequency ω, the corresponding component is set to 0 if $\omega > \omega_0$ and is let unchanged otherwise. We then perform an inverse Fourier transform to get our result.

To finish we have to apply, as in Section 2, a cutoff on each image in order to set u_z to 0 in the area that the wave has not reached yet. As suggested earlier, we use the formula (3.3) based on the arctan function rather than a Heavyside in order to obtain a more regular front. The complete algorithm is summed up in Algorithm 2.

An example of simulation using this algorithm is shown in Figure 4. The resulting image is close to the one shown in Figure 2. Also note that this time it is possible to work on the full space-time grid, within reasonable time (approximately 0.3s).

However when displaying the evolution of the noise, we observe that in our simulations the patches are being irregularly deformed and have no visible motion whereas in the real simulations, the noise behaves more like smudges coming from the front wave and going down.

3.2 Modifications to enhance temporal behavior

To mimic this behavior we made some adjustments to the original algorithm.

We start by applying a low pass not only in space, but also in time. This means that we perform a three-dimensional discrete Fourier transform on the whole sequence of images before applying a filter B (for consistency we changed the notation as it is now a three-dimensional function), as shown in (3.6). This smooths out the time evolution of the noise, making its appearance closer to the ones in the CEA simulations. Since this filter already manages the time correlation, we also remove the function c and simply use a fixed variance σ^2 for the Gaussian random variables used in the phase update.

Another modification has been implemented in order to recreate the smudges going down, as described earlier. In the phase update, we take a combination between the previous update (3.1) and the value of the phase two space steps above. This allows the information coming from above to partially propagate downwards. This is done by a convex combination of both

Algorithm 3 Dynamic simulation of the noise with space-time smoothing and added space correlations.

Set $\phi(.,.,t=0)$, $corr \in [0,1]$, $\sigma_{corr} \geq 0$, $\sigma \geq 0$, $T_{max} > 0$.
for $0 \leq t \leq T_{max} - \Delta t$ **do**
  for all x, z **do**
    Sample: $corr_{eff} = \mathcal{N}(corr, \sigma^2_{corr})$
    Compute:

$$\phi(x,z,t+\Delta t) = (1 - corr_{eff}) \left(\phi(x,z,t) + \mathcal{N}(0,\sigma^2) \right) + corr_{eff} \times \phi(x,z+2\Delta z,t). \quad (3.5)$$

▷ Computation of the new phase

    Compute:

$$\tilde{u}_z(x,z,t) = \mathcal{R}\left(\mathcal{F}^{-1}(B(\mathcal{F}(e^{i\phi})))(x,z,t) \right). \quad (3.6)$$

▷ Application of low pass filter

  end for
end for
Compute:

$$u_z(x,z,t) = \frac{\frac{\pi}{2} - \arctan(\alpha(z - vt))}{\pi} \tilde{u}_z(x,z,t). \quad (3.7)$$

▷ Application of arctan to recreate the no-wave area

quantities, using a coefficient $corr_{eff}$, see (3.5). This coefficient is chosen to be random, which in practice provides better results. A heuristic explanation is that the smudges do not always propagate as they can be cut out by other smudges and aggregates, so it seems natural for the combination to be stochastic. This algorithm is detailed in Algorithm 3.

The Figure 4 displays a comparison between two images simulated by Algorithm 2 and Algorithm 3. At fixed time, both give similar results. However, Algorithm 3 seems to produce better time evolving signals, with more time regularity and the exhibition of smudges going downwards. A simulation takes roughly 0.5 seconds to be done with this algorithm. While it is still long, it is a significant improvement compared to the finite-element solver. Note that there are many different free parameters that we did not explicit. As a first approach, these are chosen by trials and fails, based on the visual appearance of the results.

4 Quantitative criteria

The method proposed in the previous section seems promising. This model have many parameters and until now, the fitting has only been made by visualization. Of course, this is not satisfying and we would like to dispose of quantitative criteria to evaluate the accordance of the signals generated by our model with respect to the simulations provided by the CEA.

In this section, we propose many *dissimilarity measures* and we compare them through their capacity to discriminate wave profiles provided by the finite-element solver according to the input parameters (frequency f of the incoming wave, proportion p and size r of the aggregates). Afterward, we use the best dissimilarity measure in the implementation of an optimization algorithm, namely the *Nelder-Mead method*, in order to fit the parameters of

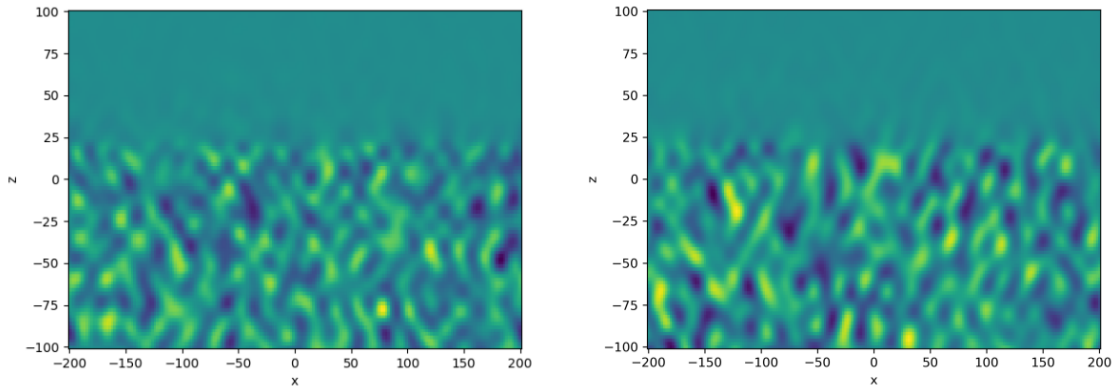


Figure 4: Two noise simulations using the dynamic method. The left image has been generated using Algorithm 2 and the right image using Algorithm 3. Both codes have been written in `python`, using `scipy.fftpack` for the discrete Fourier transforms.

the generative model.

4.1 Dissimilarity measures

For a simulation $U_i(x,z,t)$ and a direction $\kappa \in \{x,z,t\}$, we denote by $\mathbb{V}_i^\kappa$ the (biased) variance along the axis κ. We also denote $\mathbb{V}_i^{x,z,t}$ the variance along the temporal and both spatial directions. Finally, we denote by $\mathbb{V}_i^x(z_0)$ the variance along the axis x at fixed z_0. In this report, we propose and study the following dissimilarity measures:

$$
\begin{aligned}
\mathrm{d}_1(i,j) &= |\mathbb{V}_i^{x,z,t} - \mathbb{V}_j^{x,z,t}|\,,\\
\mathrm{d}_2(i,j) &= \|\mathbb{V}_i^t - \mathbb{V}_j^t\|_2\,,\\
\mathrm{d}_3(i,j) &= \|(\mathbb{V}_i^t)^T\mathbb{V}_i^t - (\mathbb{V}_j^t)^T\mathbb{V}_j^t\|_2\,,\\
\mathrm{d}_4(i,j) &= \|\mathbb{V}_i^t(\mathbb{V}_i^t)^T - \mathbb{V}_j^t(\mathbb{V}_j^t)^T\|_2\,,\\
\mathrm{d}_5(i,j) &= \|\mathbb{V}_i^x(z_0) - \mathbb{V}_j^x(z_0)\|_2\,,
\end{aligned}
$$

where $\|\cdot\|_2$ stands for the Frobenius norm in the appropriate vector space and $(\mathbb{V}_i^x)^T$ for the transpose of $\mathbb{V}_i^x$. Because they only take the variances and covariances into account, these functions measure the difference of dispersion between two simulations. We could have considered other combinations of variances and covariances distances but it was not possible because of insufficient time.

We want to use one of these dissimilarity measures to fit our generative model. As a consequence, one should choose the measure with respect to its capacity to correctly discriminate the simulations provided by the finite-element solver according to the input parameters f, p and r. In this direction, we construct for each dissimilarity measure a dissimilarity matrix where the entry (i,j) is given by $d(i,j)$, see an example in Figure 5. For each input parameter, there are ten simulations which are listed as follows:

- $i \in \{1,\dots,10\}$ correspond to parameters $f = 5$ kHz, $p = 20$ and $r = 2$;

	$\mathrm{d}_1(i,j)$	$\mathrm{d}_2(i,j)$	$\mathrm{d}_3(i,j)$	$\mathrm{d}_4(i,j)$	$\mathrm{d}_5(i,j)$
ARI	0.41	0.62	0.63	0.63	0.60
AMI	0.52	0.73	0.75	0.74	0.72

Table 1: Comparison of dissimilarity measures. The ARI and AMI scores are averaged over 50 iterations of the spectral clustering algorithm.

- $i \in \{11, \dots, 20\}$ correspond to parameters $f = 5$ kHz, $p = 5$ and $r = 4$;
- $i \in \{21, \dots, 30\}$ correspond to parameters $f = 5$ kHz, $p = 10$ and $r = 4$;
- $i \in \{31, \dots, 40\}$ correspond to parameters $f = 5$ kHz, $p = 20$ and $r = 4$;
- $i \in \{41, \dots, 50\}$ correspond to parameters $f = 5$ kHz, $p = 40$ and $r = 4$;
- $i \in \{51, \dots, 60\}$ correspond to parameters $f = 10$ kHz, $p = 20$ and $r = 4$;
- $i \in \{61, \dots, 70\}$ correspond to parameters $f = 15$ kHz, $p = 20$ and $r = 4$.

In addition, we compare the simulations to white noises filtered by a low pass filter, which are labeled by $i \in \{71, \dots, 80\}$.

We convert the dissimilarity matrix into a similarity matrix by applying the operator $x_{i,j} \mapsto -x_{i,j} + \max_{k,\ell} x_{k,\ell}$ to each entry. To evaluate the capacity of a measure to discriminate the input parameters, we apply spectral clustering to these similarity matrices and we compute the corresponding adjusted rand index (ARI) and adjusted mutual information (AMI), see Table 1. Except for $\mathrm{d}_1(i,j) = |\mathbb{V}_i^{x,z,t} - \mathbb{V}_j^{x,z,t}|$, the scores obtained for the proposed measures are alike (around 0.55 for the ARI and around 0.75 for the AMI). This suggested that the d_2, d_3, d_4 and d_5 are more or less equivalent. In the sequel, we choose d_5 for fitting our generative model. This is motivated by the fact that it will be easier to extract the coda with this dissimilarity measure.

In Figure 6 is drawn the graph of the function $t \mapsto \mathbb{V}^x(z_0, t)$ averaged on the ten simulations samples for each input parameters. An interpolation procedure has been applied in order to normalize the time scale. The graph exhibits a long-time behavior corresponding to the coda whereas small times correspond to the wave front propagation which generates a totally different noise regime. The long-time behavior seems sufficient to distinguish different input parameters.

4.2 Parameter optimization

In this subsection, we explain how to fit the parameters of the generative model according to the simulations provided by the CEA with the input parameters $f = 5$ kHz, $p = 20$ and $r = 4$.

Because we are only interested on modeling the coda, we fit only with respect to the long-time behavior that is, for times between 50 and 90 in Figure 6. Then we use the L^2-norm on this zone to perform an optimization on the parameters of Algorithm 3 using the Nelder-Mead algorithm, a classical method for non smooth problems. Because of the duration of the simulations, we had to limit ourselves to 600 steps. While this does not

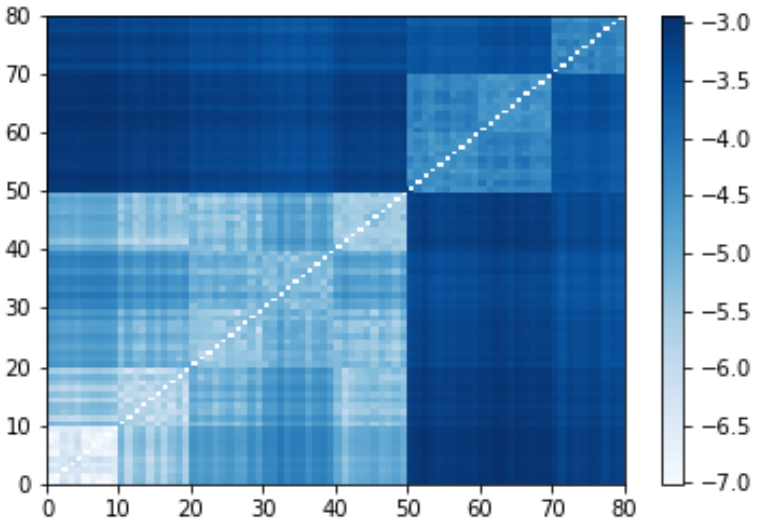


Figure 5: Dissimilarity matrice for $\mathrm{d}(i,j) = \|\mathbb{V}_i^x(z_0) - \mathbb{V}_j^x(z_0)\|_2$. There are ten samples for each input parameter. For $i \in \{1,\dots,50\}$ (resp. $i \in \{51,\dots,60\}$, resp. $i \in \{61,\dots,70\}$) the frequency is set to $f = 5$ kHz (resp. $f = 10$ kHz, resp. $f = 15$ kHz). The samples $i \in \{71, 80\}$ correspond to white noises filtered by a low pass filter.

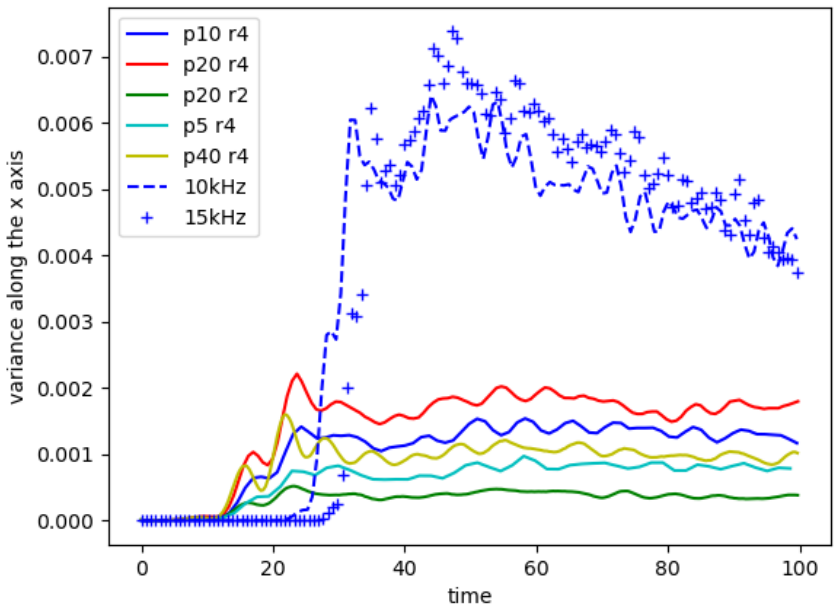


Figure 6: Graphs of the function $t \mapsto \mathbb{V}^x(z_0, t)$ averaged on samples for each input parameter.

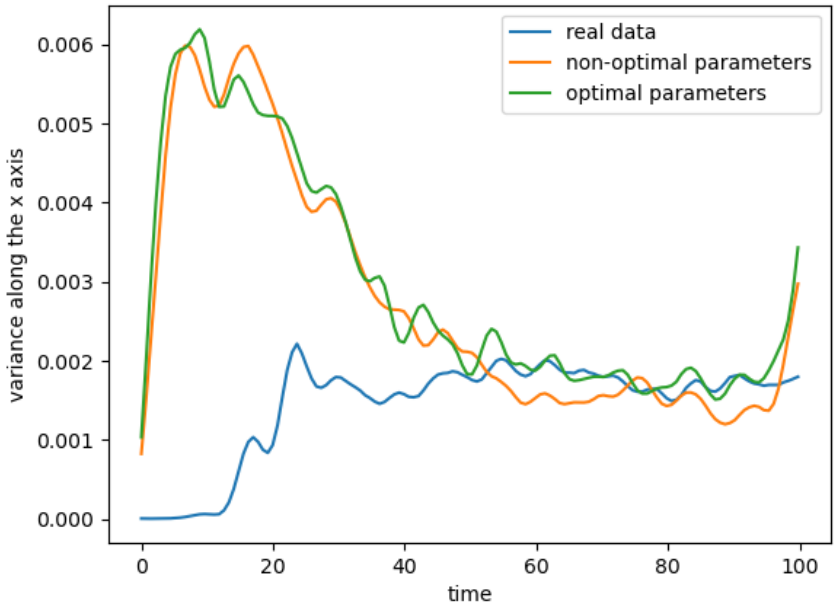


Figure 7: Graphs of the function $t \mapsto \mathbb{V}^x(z_0, t)$ for real data and two simulations provided by the algorithm 3, one with non-optimal parameters and the other one after a numerical optimization.

allow for the convergence of the algorithm, Figure 7 shows that it provides a significant improvement in terms of curve-fitting. As expected, we observe an important discrepancy between our simulations and the data in small times. Also, in the last times, we observe a slope of the curves corresponding to our generative model. We believe that it is a boundary effect related to numerical issues, probably linked with the low pass filter.

This optimization procedure could be improved with another choice of optimization algorithm (Nelder-Mead algorithm was chosen for implementation convenience) or with more data corresponding to the coda. However, this work indicates that it should be possible to optimize the parameters of our algorithm to adjust any of the parameters provided by the CEA.

5 Conclusion and further topics of research

In this work, different approaches to sample random fluctuations of ultrasonic wave within heterogeneous materials have been proposed. A first approach based on Gaussian processes gave promising results but has been discarded for computational cost issues. A second approach, inspired by speckle simulations, gave better and faster results. The addition of a low pass filter in time and an enforced space correlation along the z axis have been a key to obtain realistic time-evolving simulations. Finally different metrics have been proposed in order be able to quantify the quality of the simulations. They have been shown not only to be able to discriminate between different wave and material parameters, but also to identify

noise that is not the result of an ultrasound propagation. Thus this allowed us to perform an optimization on the parameters of our model in order to get a better fit on the CEA simulations.

Multiple directions can be taken for future works. We hereby mention some of them.

A first work would have to be done on the code in order to get faster simulations. The current code has been written in `python`, but other languages could be explored in order to fasten the Gaussian sampling as well as the computations of the discrete Fourier transforms, which are the two components which take most of the simulation time. Note that by nature the algorithm has a high potential for parallel computations. This would allow for a significantly faster parameter optimization and faster statistical studies afterwards.

For the moment we only considered one neighbour in the update (3.5), but studies have to be made in order to find which exact neighbours should be taken into account in order to properly retrieve the behavior of the downward smudges.

A Fourier transform could be performed on the data in order to identify some characteristic modes or other properties that could nourish the pool of quantitative criteria.

Other wavelet transforms could be considered, not only to establish metrics, but also to modify the low pass filter. A suitable wavelet basis could act as a selector on the simulation in order to recreate with more fidelity the smudges observed in the data.

Finally, while the front wave area is not the most useful for detection, this is where the coda comes from. Thus, working on a proper modelization of this area could lead to significant improvements in the quality of the rest of the simulation. In this area the noise is composed of small patches created by the shocks between the front wave and the aggregates. These are very short-time patches which means that a dynamic procedure as described here might not be the best lead for this part. In particular, the use of a Generative Adversarial Network (GAN) should be considered. To briefly describe GANs, they are Neural Networks (NN) composed of one NN which has the role of creating a random vector and another one which is supposed to establish if it comes from a real set of data or if this is a fake. The alternative optimizations of both NN allows for high quality creations by the first one. This process is commonly used in image or text generation. In our case, as we have multiple time steps to simulate, it would not be realistic to train a GAN to generate the coda for the full simulation (since we have very limited data). However, successive generations of a few images of front wave noise would be possible since one CEA simulation provides multiple data of this nature. Some work would then have to be done in order to smoothly go from one small sequence to another.

References

[1] Chassignole, B. *Influence de la structure métallurgique des soudures en acier inoxydable austénitique sur le contrôle non destructif par ultrasons.* These de doctorat, Lyon, INSA, Jan. 2000.

[2] Chassignole, B., El Guerjouma, R., Ploix, M.-A., and Fouquet, T. Ultrasonic and structural characterization of anisotropic austenitic stainless steel welds: Towards

a higher reliability in ultrasonic non-destructive testing. *NDT & E International 43*, 4 (2010), 273–282.

[3] Federico, A., Kaufmann, G., Galizzi, G., Rabal, H., Trivi, M., and Arizaga, R. Simulation of dynamic speckle sequences and its application to the analysis of transient processes. *Optics communications 260*, 2 (2006), 493–499.

[4] Imperiale, A., and Demaldent, E. A macro-element strategy based upon spectral finite elements and mortar elements for transient wave propagation modeling. application to ultrasonic testing of laminate composite materials. *International Journal for Numerical Methods in Engineering 119*, 10 (2019), 964–990.

[5] Ploix, M.-A. M. *Étude de l'atténuation des ondes ultrasonores. Application au contrôle non destructif des soudures en acier inoxydable austénique.* Theses, INSA de Lyon (France), Dec. 2006.

[6] Santner, T. J., Williams, B. J., Notz, W., and Williams, B. J. *The design and analysis of computer experiments*, vol. 1. Springer, 2003, pp. 145–245.

[7] Williams, C. K., and Rasmussen, C. E. *Gaussian processes for machine learning*, vol. 2. MIT press Cambridge, MA, 2006.