

Optimisation de la précision et de la robustesse de la mesure  
d'épaisseur des matériaux corrodés par l'intégration des  
réseaux de neurones Transformer

par

Thibault SENDRA

MÉMOIRE PAR ARTICLES PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE  
SUPÉRIEURE COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA  
MAÎTRISE AVEC MÉMOIRE EN GÉNIE MÉCANIQUE  
M. Sc. A.

MONTRÉAL, LE 12 MAI 2025

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC



Thibault Sendra, 2025



Cette licence Creative Commons signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette oeuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'oeuvre n'ait pas été modifié.

**PRÉSENTATION DU JURY**

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Pierre Bélanger, directeur de mémoire  
Département de génie mécanique à l'École de technologie supérieure

M. Martin Viens, président du jury  
Département de génie mécanique à l'École de technologie supérieure

M. Matthew Toews, membre du jury  
Département de génie des systèmes à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 24 AVRIL 2025

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE



## **REMERCIEMENTS**

En premier lieu, je tiens à exprimer ma gratitude au Professeur Pierre Bélanger pour les opportunités qu'il m'a offertes, ainsi que pour son soutien et sa confiance tout au long de ce projet de recherche.

Je remercie également tous les membres du laboratoire PULÉTS : Antoine, Aubin, Aurélien, Éloi, Paul, Lucas, Tony, Rafaël, Maxime, Thibault et Wala'a. Leur présence et leur aide ont rendu cette expérience non seulement plus enrichissante, mais aussi bien plus agréable.

Un grand merci aux membres de Nucleom et d'Eddify, en particulier Sevan, Hugo, Étienne et Marco, dont l'engagement et l'intérêt pour mon projet ont donné un véritable sens à cette recherche. Leur soutien constant a été une source précieuse de motivation.

Enfin, je souhaite remercier ma famille ainsi que mon colocataire pour leur soutien indéfectible tout au long de ma maîtrise au Québec. Leur présence, même à distance, a été essentielle.



# **Optimisation de la précision et de la robustesse de la mesure d'épaisseur des matériaux corrodés par l'intégration des réseaux de neurones Transformer**

Thibault SENDRA

## **RÉSUMÉ**

L'évaluation précise de l'épaisseur des matériaux corrodés représente un enjeu majeur en contrôle non destructif (CND), en raison de la complexité des signaux ultrasonores et de la superposition des échos. Face aux limites des architectures de réseaux de neurones convolutifs (CNN) utilisées dans la littérature, ce travail explore l'intégration des réseaux de neurones Transformer (TNN) afin d'améliorer la résolution et la fiabilité des mesures d'épaisseur.

Entraînée et validée sur un dataset combinant des simulations par éléments finis et des données expérimentales issues de mesures sur un bloc corrodé, une première architecture hybride CNN-TNN a été développée. Validée dans un article révisé par les pairs, cette approche a démontré que l'intégration d'un module Transformer introduit un contexte global dans l'apprentissage, permettant ainsi de surpasser les performances des modèles CNN existants. Toutefois, cette approche présentait une complexité computationnelle élevée et un risque de surajustement.

Pour y remédier, une version optimisée, le Lite CNN-TNN, a été conçue en intégrant une couche convolutive en parallèle du mécanisme Multi-Head Self-Attention (MHSA). Cette hybridation a permis de capitaliser sur la capacité des CNN à extraire des caractéristiques locales tout en conservant les avantages du Transformer pour capturer des interactions globales. Cette approche a amélioré la stabilité de l'apprentissage et réduit significativement le nombre de paramètres du modèle.

Par ailleurs, le modèle de simulation a été optimisé en intégrant des signaux réalistes issus d'échos expérimentaux, en ajustant la taille de la sonde et en affinant la méthodologie de labellisation pour mieux refléter les conditions réelles. L'optimisation de la méthode d'apprentissage a également été renforcée grâce à des techniques d'augmentation de données adaptées.

Les résultats obtenus confirment la pertinence de cette approche : le modèle Lite CNN-TNN a atteint un taux de succès de 98,84% sur le dataset expérimental et 97,40% sur le dataset simulé, surpassant le meilleur modèle CNN de la littérature qui a obtenu 98,70% et 95,70% respectivement, tout en étant 24 fois plus rapide.

**Mots-clés:** Apprentissage profond, Contrôle non destructif, Corrosion, Déconvolution, Simulation par éléments finis, Mesure d'épaisseur, Intelligence Artificielle, Réseau de neurones convolutifs, Réseau de neurones Transformer, Temps de Vol, Traitement du signal, Test ultrasonique





# **Optimization of accuracy and robustness in thickness measurement of corroded materials through the integration of Transformer neural networks**

Thibault SENDRA

## **ABSTRACT**

Accurate evaluation of corroded material thickness is a major challenge in nondestructive evaluation (NDE) due to the complexity of ultrasonic signals and overlapping echoes. Given the limitations of Convolutional Neural Network (CNN) architectures found in the literature, this work explores the integration of Transformer Neural Networks (TNN) to improve the resolution and reliability of thickness measurements.

Trained and validated on a dataset combining finite element simulations and experimental data from measurements on a corroded block, an initial hybrid CNN-TNN architecture was developed. Validated in a published, peer reviewed paper, this approach demonstrated that incorporating a Transformer module introduces a global learning context, enabling superior performance compared to existing CNN models. However, this method exhibited high computational complexity and a risk of overfitting.

To address these limitations, an optimized version, Lite CNN-TNN, was designed by integrating a convolutional layer in parallel with the Multi-Head Self-Attention (MHSA) mechanism. This hybridization leveraged the CNN's ability to extract local features while preserving the Transformer's capacity to capture global interactions. This approach improved learning stability while significantly reducing the model's parameter count.

Furthermore, the simulation model was enhanced by incorporating realistic signals from experimental echoes, adjusting probe size, and refining the labeling methodology to better reflect real-world conditions. The learning process was also optimized through tailored data augmentation techniques.

The results confirm the effectiveness of this approach : the Lite CNN-TNN model achieved a success rate of 98.84% on the experimental dataset and 97.40% on the simulated dataset, surpassing the best CNN model from the literature, which reached 98.70% and 95.70% respectively, while being 24 times faster.

**Keywords:** Machine Learning, Nondestructive evaluation, Corrosion, Deconvolution, Finite element simulation, Thickness measurement, Artificial Intelligence, Convolutional Neural Network, Transformer Neural Network, Time-of-Flight, Signal processing, Ultrasonic testing



# TABLE DES MATIÈRES

	Page
INTRODUCTION .....	1
CHAPITRE 1 REVUE DE LA LITTÉRATURE .....	3
1.1 Introduction aux ultrasons dans les solides .....	3
1.1.1 Les ondes volumiques et leur propagation dans les solides .....	3
1.1.2 Le phénomène d’atténuation .....	8
1.1.3 Le comportement des ultrasons aux interfaces .....	9
1.1.3.1 Réflexion et transmission en incidence normale .....	10
1.1.3.2 Réflexion, réfraction et transmission en incidence oblique .....	12
1.2 Le contrôle non destructif par ultrasons .....	13
1.2.1 Les sondes à ultrasons .....	14
1.2.1.1 Les sondes mono-élément .....	15
1.2.1.2 Les sondes multi-éléments .....	17
1.2.2 Le A-scan, B-scan et C-scan .....	18
1.2.3 Le temps de vol .....	20
1.2.4 Le phénomène de superposition des échos .....	21
1.2.5 L’inspection de pièces corrodées .....	23
1.3 Les méthodes de mesure d’épaisseur .....	24
1.3.1 La méthode de seuil .....	25
1.3.2 La corrélation croisée .....	27
1.3.3 Le Matching Pursuit (MP) .....	28
1.3.4 La déconvolution .....	28
1.4 Les réseaux de neurones et leurs architectures .....	31
1.4.1 Les fondements de l’apprentissage profond .....	32
1.4.2 Le réseau de neurones à propagation avant (FFN) .....	32
1.4.3 Les couches intermédiaires .....	34
1.4.3.1 Les fonctions d’activation .....	34
1.4.3.2 La normalisation en Batch (Batch Normalization) et la normalisation en couche (Layer Normalization) .....	36
1.4.3.3 Le dropout .....	36
1.4.4 Le réseau de neurones Convolutif (CNN) .....	37
1.4.5 Le réseau de neurones Transformer (TNN) .....	40
1.4.6 L’optimisation des modèles .....	42
1.5 Conclusion .....	43
CHAPITRE 2 ON THE USE OF A TRANSFORMER NEURAL NETWORK TO DECONVOLVE ULTRASONIC SIGNALS .....	45
2.1 Abstract .....	45
2.2 Introduction .....	46
2.3 Materials and Methods .....	49

2.3.1	Problem formulation .....	49
2.3.2	Database generation .....	51
2.3.2.1	Simulation database .....	51
2.3.2.2	Experimental database .....	56
2.3.2.3	Training datasets .....	56
2.3.3	Deep Neural Network architecture .....	57
2.3.3.1	Pre-processing module .....	58
2.3.3.2	Transformer encoding module .....	59
2.3.3.3	Post-processing module .....	61
2.3.4	Training .....	61
2.3.4.1	Training and batching .....	61
2.3.4.2	Optimizer .....	62
2.3.5	Comparison to other architectures and methods .....	62
2.3.6	Evaluation methodology .....	63
2.4	Results .....	64
2.4.1	Transformer study .....	64
2.4.1.1	Convolution kernel size .....	65
2.4.1.2	Architecture size .....	66
2.4.1.3	Ablation studies .....	66
2.4.1.4	Database composition .....	67
2.4.2	Comparative results .....	68
2.5	Discussion .....	69
2.5.1	Convolution kernel size .....	69
2.5.2	Size .....	71
2.5.3	Ablation .....	71
2.5.4	Database composition .....	72
2.5.5	Comparative results .....	74
2.5.6	Error analysis .....	76
2.6	Conclusion .....	78
CHAPITRE 3 AMÉLIORATION DU MODÈLE .....		81
3.1	Overfit et généralisation du Transformer .....	81
3.1.1	Utilisation d'une séquence aléatoire fixe (seed) .....	81
3.1.2	Identification du problème .....	82
3.1.3	R-drop .....	84
3.1.4	Transformer Lite .....	86
3.1.5	Résultats .....	88
3.2	Amélioration de la simulation .....	89
3.2.1	Signal émis .....	90
3.2.2	Corrosion et taille de sonde .....	93
3.2.3	Résultats .....	97
3.3	Méthodologie d'apprentissage .....	98
3.3.1	Concaténation contre apprentissage par transfert (Transfer Learning) .....	98

3.3.2	Augmentation de données .....	101
3.3.3	Résultats .....	105
3.4	Conclusion .....	105
CONCLUSION ET RECOMMANDATIONS .....		109
BIBLIOGRAPHIE .....		111



## LISTE DES TABLEAUX

	Page
Tableau 2.1	Tabular summary of the datasets distribution. .... 57
Tableau 2.2	Number of parameters, number of FLOPS, Success Rate (SR) and MAE between actual and predicted distances shown in Fig.2.12 of the different methods on the simulation and the experimental test datasets. .... 69
Tableau 3.1	Méthodes combinées ou concaténées selon le numéro du dataset. ....103
Tableau 3.2	Nombre de paramètres, complexité de calcul (FLOPS), taux de succès (SR) et erreur absolue moyenne (MAE) des différentes architectures sur les datasets de validation expérimental et de test simulé. ....106





## LISTE DES FIGURES

	Page
Figure 1.1	Illustration des deux modes de propagation ondulatoires ..... 7
Figure 1.2	Réflexion et transmission d'une onde incidente normale à la frontière ... 10
Figure 1.3	Réflexion et réfraction d'une onde incidente longitudinale oblique à la frontière de deux solides avec $v_i > v_{tL}$ ..... 12
Figure 1.4	Schéma d'une sonde piézoélectrique mono élément ..... 16
Figure 1.5	Schéma d'une sonde multi-éléments à 12 transducteurs piézoélectriques ..... 17
Figure 1.6	Schéma exemple d'un A-scan ..... 19
Figure 1.7	Schéma exemple d'un B-scan ..... 19
Figure 1.8	Schéma exemple d'un C-scan ..... 20
Figure 1.9	Exemple de détermination du temps de vol grâce au maximum des échos ..... 21
Figure 1.10	Exemples d'A-scans avec des échos superposés. .... 22
Figure 1.11	Simulation d'un A-scan (a) sur un bloc corrodé (b). .... 24
Figure 1.12	Représentation de la méthode de (a) seuil fixe, (b) seuil variable, et (c) maximum de l'enveloppe. .... 26
Figure 1.13	Exemples de l'effet de convolution. .... 29
Figure 1.14	Schéma d'un réseau de neurone à propagation avant. .... 33
Figure 1.15	Graphique des fonctions d'activation (a) ReLU et (b) Tanh. .... 35
Figure 1.16	Schéma de l'effet du dropout sur des couches de neurones d'un FFN. ... 37
Figure 1.17	Schéma de l'action d'une fenêtre de convolution dans une couche de réseau de neurones convolutif. .... 38
Figure 1.18	Exemple simple d'architecture d'un réseau de neurones convolutif. .... 39
Figure 1.19	Schématisation de l'encodeur de l'architecture Transformer avec 2 têtes et $d_{model} = 2$ . .... 42

Figure 2.1	Example of the labeling methodology for minimal distance under the probe with : (a) the result coming from probe on the generated block (b). 50
Figure 2.2	Schematic of the block generated for simulation with its mesh. .... 51
Figure 2.3	Schematic of the corrosion profile generation with : (a) a corrosion profile of the corroded block, (b) the FFT of (a), (c) the algorithm used to randomly generate a corrosion profile with similar properties to the experimental profile, and (d) a typical simulated profile. .... 54
Figure 2.4	Pictures of the experimental setup : (a) laser scan of the corroded staircase specimen, (b) example of an acquisition with the multi-element probe mounted on the robot, and (c) the probe's absolute position on the specimen to determine the minimal distances beneath it. 55
Figure 2.5	Proposed CNN-TNN architecture with reference sizes. .... 57
Figure 2.6	Convolution Block details. .... 58
Figure 2.7	Feed Forward Block details. .... 58
Figure 2.8	Success rate according to the kernel size of : (a) the conv 1 and (b) the conv 3 layer on the simulation and the experimental test datasets. .... 65
Figure 2.9	Success rate with respect to the dimension of the model $d_{model}$ with one or two transformers in series on the simulation and the experimental test datasets. .... 66
Figure 2.10	Success rate depending on the removed blocks or modules on the simulation and the experimental test datasets with : (a) unzoomed and (b) zoomed view. .... 67
Figure 2.11	Success rate following the quantity of simulations within the training dataset on the simulation and the experimental test datasets. .... 68
Figure 2.12	Comparative bar chart of (a) the success rates and (b) the MAE of different AI architectures and industrial algorithm on the simulation and the experimental test datasets. .... 68
Figure 2.13	Comparative analysis of various AI architectures on the experimental test dataset : (a) success rates versus the number of parameters, (b) MAE versus the number of parameters, (c) success rates versus the number of FLOPS, and (d) MAE versus the number of FLOPS. .... 70

Figure 2.14	Distance and deconvolution prediction of the experimental sample 106 with : (a) TNN, (b) Chapon's CNN, (c) Shpigler's CNN, (d) Thresholding technique, (e) ResNet-50, and (f) EfficientNet-B0. ....	75
Figure 2.15	Failure cases of the TNN's deconvolution predictions on experimental sample : (a) 622, (b) 6, (c) 489, and (d) 600 .....	77
Figure 3.1	Graphique comparatif du taux de succès en fonction des époques avec modifications du modèle CNN-TNN. ....	83
Figure 3.2	Schéma illustrant le principe du R-Drop : l'entrée $x$ traverse deux fois le modèle représenté à gauche, générant deux sorties $y_1$ et $y_2$ qui diffèrent en raison du sous-modèle induit par un dropout avec un taux $p = 0.5$ illustré sur l'image de droite. ....	85
Figure 3.3	Graphique comparatif du taux de succès en fonction des époques avec ( $\alpha = 0.5$ ) et sans ( $\alpha = 0$ ) R-drop. ....	85
Figure 3.4	Représentation de l'architecture Transformer Lite proposé, optimisée pour la déconvolution. ....	87
Figure 3.5	Graphique comparatif du taux de succès en fonction des époques des différentes architectures avec un taux de R-drop $\alpha = 0.5$ . ....	87
Figure 3.6	Graphique comparatif du taux de succès en fonction des époques regroupant les méthodes appliquées pour répondre au problème de surapprentissage. ....	88
Figure 3.7	Graphique exemple des signaux d'émission (a) Gaussian Double, (b) Nakagami et (c) du 64 <sup>ème</sup> élément de la sonde multi-élément. ....	91
Figure 3.8	Taux de succès en fonction de la méthodologie de génération de signal. .	92
Figure 3.9	Profils de corrosion réel .....	94
Figure 3.10	Schématisation des considérations de labellisation : corrosion réelle (a-c), avec largeur adaptée (width) (d-f), et avec ouverture de faisceau (BO) (g-i). (a), (d) et (g) représentent le bloc simulé avec la zone de labellisation. (b), (e) et (h) montrent le profil de corrosion utilisé et la position de la distance minimale pour la labellisation. (c), (f) et (i) illustrent le A-scan simulé avec la labellisation associée à la distance minimale. ....	95
Figure 3.11	Taux de succès en fonction de la méthodologie de corrosion, de la taille de la sonde et de la méthode de labellisation. ....	96

Figure 3.12	Taux de succès en fonction de la méthodologie de simulation. ....	97
Figure 3.13	Taux de succès en fonction du nombre de simulations et de la méthodologie d'apprentissage avec (a) 100 données expérimentales et (b) 1200 données expérimentales. ....	100
Figure 3.14	Méthodes d'augmentation de données : (a) ajout de bruit (SNR), (b) modification de l'ordre du filtre passe-bande (Redu), (c) décalage temporel (Shift) et (d) compression ou étirement du signal (StrComp) .	101
Figure 3.15	Comparaison du taux de succès en fonction du numéro de combinaison décrit tableau 3.1 entre la combinaison des méthodes et leur concaténation. ....	104
Figure 3.16	Taux de succès (a,b) et MAE (c,d) des différentes architectures sur les datasets (a,c) de validation expérimental et (b,d) de test simulé en fonction de la complexité de calcul (MFLOPS). ....	108

## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AI	Intelligence Artificielle (Artificial Intelligence)
BO	Ouverture du faisceau (Beam Opening)
CND	Contrôle Non Destructif (Nondestructive Evaluation)
CNN	Réseau de Neurones Convolutifs (Convolutional Neural Network)
DA	Augmentation de données (Data Augmentation)
EEG	Électroencéphalogramme
EMAT	Sonde acoustique électromagnétique (Electromagnetic acoustic Transducer)
FEA	Analyse par éléments finis (Finite Element Analysis)
FFN	Réseau de Neurones à propagation avant (Feedforward Neural Network)
FLOPS	Nombre d'opérations flottantes par seconde (Floating Point Operations Per Second)
FMC	Capture de Matrice Complète (Full Matrix Capture)
GAN	Generative Adversarial Networks
IA	Intelligence Artificielle
L	Grand (Large)
MAE	Erreur Absolue Moyenne (Mean Absolute Error)
MHSA	Multi-Têtes d'Auto-Attention (Multi-Head Self Attention)
ML	Apprentissage machine (Machine Learning)
MP	Matching Pursuit
MSE	Erreur Quadratique Moyenne (Mean Squared Error)
NCC	Corrélation croisée normalisée (Normalized Cross-Correlation)
NDE	Contrôle Non Destructif (Nondestructive Evaluation)
OMP	Orthogonal Matching Pursuit
OS	Système d'opération (Operating System)
ReLU	Rectified Linear Unit
Redu	Réduction de bruit

R-Drop	Abandon Régularisée (Regularized Dropout)
RNN	Réseau de Neurones Récurrent (Recurrent Neural Network)
ROI	Région d'intérêt (Region Of Interest)
S	Petit (Small)
Shift	Décalage temporel (Time Shift)
SMP	Support Matching Pursuit
SNR	Rapport Signal sur Bruit (Signal-to-Noise Ratio)
SR	Taux de succès (Success Rate)
StrComp	Étirement ou Compression (Stretch Compression)
TFM	Méthode de focalisation en tous points (Total Focusing Method)
TL	Apprentissage par transfert (Transfer Learning)
TNN	Réseau de Neurones Transformer (Transformer Neural Network)
ToF	Temps de Vol (Time-of-Flight)
ViT	Vision Transformer

## LISTE DES SYMBOLES ET UNITÉS DE MESURE

$u$	Déplacement de la particule (m)
$t$	Temps (s)
$V$	Vitesse de propagation de l'onde (m/s)
$\lambda$	Longueur d'onde (m)
$\rho$	Densité massique (kg/m <sup>3</sup> )
$A$	Amplitude du signal
$T_{ij}$	Composante du tenseur de contrainte de la face normale à la direction i dans la direction j (Pa)
$S_{ij}$	Composante du tenseur de déformation de la face normale à la direction i dans la direction j (sans unité)
$c_{ijkl}$	Tenseur d'élasticité liant les composantes ij du tenseur de contrainte aux composantes kl du tenseur de déformation (Pa)
$\delta$	Symbole de Kronecker (sans unité)
$E$	Module de Young (Pa)
$\nu$	Coefficient de Poisson (sans unité)
$\vec{\nabla}$	Opérateur Nabla
$\alpha$	Coefficient d'atténuation (m <sup>-1</sup> )
$I$	Intensité acoustique (W/m <sup>2</sup> )
$f$	Fréquence (MHz)
$Z$	Impédance acoustique (Rayl)
$v$	Vitesse de phase dans le milieu (m/s)
$R$	Coefficient de réflexion en énergie (sans unité)
$T$	Coefficient de transmission en énergie (sans unité)
$\theta_{i,r,t,L,T}$	Angle de l'onde : incidente, réfléchie, transmise, Longitudinale, Transversale (rad)
$d$	Épaisseur de la pièce (m)

$d_{min}$	Épaisseur minimale de la pièce sous la sonde (m)
$d_{predicted}$	Épaisseur prédite (m)
$d_{exp}$	Épaisseur réelle (m)
$ToF$	Temps de vol (Time-of-Flight) (s)
$SNR$	Rapport Signal sur Bruit (dB)
$y(t)$	Signal reçu
$x(t)$	Signal émis
$h(t)$	Réponse impulsionnelle du signal
$n(t)$	Bruit du signal
$f(\vec{x})$	Fonction réalisée par l'Intelligence Artificielle
$n_{filtres}$	Nombre de filtres de la couche convolutive
$k$	Dimension des filtres de la couche convolutive
$d_{model}$	Dimension du modèle Transformer
$d_k$	Dimension ajusté par le nombre de têtes du Transformer
$n_{head}$	Nombre de têtes de la couche Multi-Head Attention
$Q, K, V$	Matrices Query, Key et Value
$W$	Matrices de poids des couches d'Intelligence Artificielle
$MAE$	Erreur absolue moyenne (mm)
$MSE$	Erreur quadratique moyenne (mm <sup>2</sup> )
$L_x$	Largeur du bloc généré en simulation (mm)
$L_z$	Hauteur du bloc généré en simulation (mm)
$F_{sampling}$	Fréquence d'échantillonnage (MHz)
$Len$	Longueur du signal
$n_{cycles}$	Nombre de cycles de l'écho
$NCC$	Corrélation croisée normalisée (%)
$\sigma_{tot}$	Amplitude maximale de la corrosion
$ROI$	Région d'intérêt (sans unité)



$p$	Taux d'abandon (drop rate)
$\beta_1, \beta_2, \epsilon$	Constantes du modèle d'optimisation Adam
SR	Taux de succès (%)
$D_{KL}$	Divergence de Kullback-Leibler
$\Theta$	Paramètres du modèle de génération de signal Gaussien double ou Nakagami



## INTRODUCTION

Le contrôle non destructif (CND) vise à inspecter l'intégrité des matériaux sans compromettre leur utilisation future. Parmi les différentes techniques, les ultrasons occupent une place prépondérante en raison de leur efficacité, de leur rapidité et de leur facilité d'utilisation. Ils sont notamment employés pour l'imagerie, la caractérisation des matériaux et la mesure d'épaisseur. Ce mémoire se concentre sur cette dernière application, bien que les avancées proposées puissent être étendues à d'autres domaines.

La mesure d'épaisseur ultrasonique est essentielle dans des secteurs comme l'industrie pétrolière, le nucléaire et l'inspection des pipelines offshore, où elle permet de surveiller l'usure des parois soumises à l'érosion et à la corrosion. Le principe repose sur l'émission d'un pulse ultrasonique à travers le matériau. L'onde se réfléchit sur le fond de la pièce ainsi que sur ses irrégularités avant d'être captée par la sonde. En analysant le temps de vol et connaissant la vitesse de propagation, il est possible d'estimer l'épaisseur restante. Toutefois, la présence de corrosion entraîne des réflexions complexes et des interférences entre les échos, rendant l'interprétation du signal difficile. Dans ce contexte, il est crucial de déterminer avec précision la distance minimale sous la sonde afin d'évaluer l'état réel de la structure. Or, les algorithmes actuels peinent à extraire cette information avec fiabilité.

Les réseaux de neurones convolutifs (CNN) ont récemment démontré leur efficacité pour la séparation des échos superposés et l'estimation de l'épaisseur. Cependant, ces architectures présentent des limitations : leur fonctionnement repose sur l'analyse locale du signal, ce qui limite leur capacité à appréhender sa structure globale, essentielle pour une interprétation fine. Pour compenser cette lacune, il est nécessaire d'empiler plusieurs couches convolutives, ce qui entraîne une augmentation exponentielle du coût de calcul et complexifie l'entraînement du modèle.

Pour dépasser ces contraintes, les réseaux de neurones Transformer (TNN), initialement développés pour le traitement du langage et récemment appliqués à d'autres domaines, apparaissent comme une alternative prometteuse. Contrairement aux CNN, ces modèles permettent d'exploiter l'ensemble du signal sans restriction locale, tout en maintenant une efficacité computationnelle élevée.

Ce travail explore ainsi l'intégration des Transformers dans la mesure d'épaisseur par ultrasons, avec l'objectif d'améliorer la précision et la fiabilité des estimations. L'hypothèse avancée est que l'introduction d'un contexte global, absent des CNN, permettrait une meilleure interprétation du signal et une détection plus fiable de l'épaisseur minimale, tout en optimisant l'efficacité computationnelle.

Pour valider cette hypothèse, plusieurs étapes ont été suivies. Tout d'abord, une méthodologie a été élaborée pour générer des données simulées et expérimentales, garantissant un cadre fiable d'évaluation. Ensuite, une architecture hybride combinant CNN et Transformer a été conçue et analysée afin d'exploiter au mieux les avantages des deux approches. Face à certaines limitations identifiées, des améliorations ont été apportées, notamment par l'optimisation du modèle et de la méthodologie d'apprentissage, intégrant des techniques spécifiques d'augmentation de données.

Ces travaux ont conduit au développement d'une nouvelle architecture hybride, le CNN-TNN, capable d'exploiter simultanément les caractéristiques locales et globales des signaux ultrasonores. Son optimisation a abouti au Lite CNN-TNN, réduisant la complexité computationnelle tout en améliorant la généralisation. En parallèle, un dataset expérimental, enrichi par des simulations basées sur des profils corrodés réels, a été construit. Ensuite, une méthodologie d'entraînement adaptée a été mise en place, intégrant des techniques d'augmentation de données et le R-Drop pour limiter le surajustement. Enfin, le modèle a été évalué et comparé aux architectures CNN existantes, démontrant des gains en précision et en efficacité computationnelle.

# **CHAPITRE 1**

## **REVUE DE LA LITTÉRATURE**

### **1.1 Introduction aux ultrasons dans les solides**

Les ultrasons sont des ondes mécaniques dont la fréquence dépasse 20kHz, non audibles par l'humain, qui se propagent dans divers milieux matériels tels que les liquides, les fluides ou les solides. Leur comportement varie en fonction des propriétés du milieu traversé ou des interfaces rencontrées. Grâce à des modèles mathématiques, leur propagation peut être modélisée et prédite avec précision.

Dans le domaine du contrôle non destructif (CND), les ultrasons jouent un rôle fondamental. Ils permettent de mesurer l'épaisseur des matériaux, de détecter des défauts internes et d'évaluer les propriétés mécaniques des solides inspectés. Ce domaine a été largement exploré et résumé dans différents ouvrages tels que Cheeke (2002), Shull (2002) et Krautkrämer, Krautkrämer & Grabendörfer (1977).

Cette première partie explore le comportement des ultrasons dans les solides, en mettant l'accent sur les mécanismes physiques sous-jacents. Cette analyse fournira les bases nécessaires pour comprendre les enjeux de l'utilisation des ultrasons dans la mesure d'épaisseur des matériaux inspectés.

#### **1.1.1 Les ondes volumiques et leur propagation dans les solides**

Dans cette partie, nous suivrons le raisonnement partagé par Cheeke (2002). Pour établir les équations des ondes tridimensionnelles dans les solides, il est nécessaire de commencer par le cas unidimensionnel.

Dans un solide, chaque particule peut être modélisée par un système masse-ressort. Avec cette hypothèse, d'Alembert, en combinant la loi de Hooke et les équations de Newton, a établi au

18ème siècle l'équation de propagation suivante :

$$\frac{\partial^2 u(x, t)}{\partial x^2} = \frac{1}{V_L^2} \frac{\partial^2 u(x, t)}{\partial t^2} \quad (1.1)$$

Avec  $u$  le déplacement de la particule et  $V_L$  la vitesse de propagation de l'onde selon la direction  $x$ . Les solutions de l'équation 1.1 pour le déplacement sont de la forme :

$$u(x, t) = Ae^{j\left(\omega\left(t - \frac{x}{V_L}\right)\right)} + Be^{j\left(\omega\left(t + \frac{x}{V_L}\right)\right)} \quad (1.2)$$

Le premier terme de l'équation correspond à la propagation de l'onde selon la direction  $+x$  et le second à la propagation selon la direction  $-x$ .

Toutefois, dans un solide, le déplacement des particules ne se limite pas à la direction de propagation  $x$ . D'un point de vue physique, on peut également observer des vibrations transversales, dont le déplacement s'effectue perpendiculairement à la direction de propagation. Des composantes de déplacement peuvent ainsi apparaître selon les axes  $y$  et  $z$ .

En notant  $T$  le tenseur de contrainte et  $S$  le tenseur de déformation sur le système masse-ressort de la particule, on a d'après la seconde loi de Newton suivant la direction  $x_i$  et par symétrie de  $T$  :

$$\sum_j \frac{\partial T_{ij}}{\partial x_j} = \rho \frac{\partial^2 u_i}{\partial t^2} \quad (1.3)$$

Où  $\rho$  est la masse volumique du matériau.

On peut ensuite relier le tenseur de contraintes avec le tenseur de déformations grâce à la loi de Hooke généralisée, qui suit la formulation suivante :

$$T_{ij} = \sum_k \sum_l c_{ijkl} S_{kl} \quad (1.4)$$

Où  $c_{ijkl}$  est le tenseur d'élasticité. En émettant l'hypothèse que le solide est isotrope et homogène, l'équation 1.4 peut s'écrire sous la forme ci-dessous :

$$T_{ij} = (c_{11} - c_{44}) \sum_k S_{kk} \delta_{ij} + c_{44} S_{ij} \quad (1.5)$$

Avec  $\delta_{ij}$  le symbole de Kronecker, qui est nul si  $i \neq j$  et

$$\begin{cases} c_{11} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \\ c_{44} = \frac{E}{(1+\nu)} \end{cases} \quad (1.6)$$

Où  $E$  est le module d'Young et  $\nu$  le coefficient de poisson du matériau.

Comme :

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (1.7)$$

En implémentant 1.5 dans 1.3, le résultat suivant est obtenu :

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \frac{\partial}{\partial x_i} \left[ (c_{11} - c_{44}) \sum_k \frac{\partial u_k}{\partial x_k} \right] + \frac{c_{44}}{2} \sum_j \frac{\partial^2 u_i}{\partial x_j^2} + \frac{c_{44}}{2} \frac{\partial}{\partial x_i} \left( \sum_k \frac{\partial u_k}{\partial x_k} \right) \quad (1.8)$$

Le premier terme correspond au gradient de la divergence du champ de déplacement  $\vec{u}$ , le second est le Laplacien  $\Delta$  de  $\vec{u}$ , et le troisième est également une forme de gradient de divergence.

$$\rho \frac{\partial^2 \vec{u}}{\partial t^2} = (c_{11} - c_{44}) \vec{grad} [div (\vec{u})] + \frac{c_{44}}{2} \Delta (\vec{u}) + \frac{c_{44}}{2} \vec{grad} [div (\vec{u})] \quad (1.9)$$

Cette expression peut être réécrite sous forme vectorielle comme suit :

$$\rho \frac{\partial^2 \vec{u}}{\partial t^2} = (c_{11} - \frac{c_{44}}{2}) \vec{\nabla} (\vec{\nabla} \cdot \vec{u}) + \frac{c_{44}}{2} \nabla^2 \vec{u} \quad (1.10)$$

où  $\vec{\nabla}$  désigne l'opérateur Nabla, défini par :

$$\vec{\nabla} = \left( \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3} \right)_{\vec{i}, \vec{j}, \vec{k}} \quad (1.11)$$

tel que :

- $\vec{\nabla} \cdot \vec{u}$  représente la divergence du champ  $\vec{u}$ ,  $div(\vec{u})$  ;
- $\vec{\nabla}(u)$  est le gradient de scalaire  $u$ ,  $\vec{grad}(u)$  ;
- $\nabla^2 \vec{u}$  est le Laplacien vectoriel de  $\vec{u}$ ,  $\Delta(\vec{u})$ .

En décomposant le vecteur de déplacement  $\vec{u}$  par le gradient d'un scalaire  $\phi$  et le rotationnel d'un vecteur  $\vec{\psi}$  tel que :

$$\vec{u} = \vec{\nabla}\phi + \vec{\nabla} \times \vec{\psi} \quad (1.12)$$

On peut alors, avec l'équation 1.12 et grâce à l'identité de Helmholtz dans l'analyse vectorielle, réécrire l'équation 1.10 sous la forme suivante :

$$\vec{\nabla} \left( \rho \frac{\partial^2 \phi}{\partial t^2} - c_{11} \nabla^2 \phi \right) + \vec{\nabla} \times \left( \rho \frac{\partial^2 \vec{\psi}}{\partial t^2} - \frac{c_{44}}{2} \nabla^2 \vec{\psi} \right) = 0 \quad (1.13)$$

Comme le terme dans la parenthèse gauche est un scalaire et que le terme dans la parenthèse droite est un vecteur, ils sont séparément égaux à 0. La combinaison suivante est alors obtenue :

$$\begin{cases} \frac{\partial^2 \phi}{\partial t^2} = \frac{c_{11}}{\rho} \nabla^2 \phi \\ \frac{\partial^2 \vec{\psi}}{\partial t^2} = \frac{c_{44}}{2\rho} \nabla^2 \vec{\psi} \end{cases} \quad (1.14)$$

On identifie ainsi les équations de propagation comme défini avec l'équation 1.1. La première équation correspond à la propagation de l'onde longitudinale, tandis que la seconde correspond à l'équation de propagation de l'onde transversale. On note  $V_L$  et  $V_T$  la vitesse de l'onde



longitudinale et transversale respectivement, que l'on déduit des équations de propagation 1.14 :

$$\begin{cases} V_L = \sqrt{\frac{c_{11}}{\rho}} \\ V_T = \sqrt{\frac{c_{44}}{2\rho}} \end{cases} \quad (1.15)$$

Enfin, on peut relier les coefficients du tenseur d'élasticité avec les propriétés mécaniques  $E$  et  $\nu$  du matériau :

$$\begin{cases} V_L = \sqrt{\frac{E(1-\nu)}{\rho(1+\nu)(1-2\nu)}} \\ V_T = \sqrt{\frac{E}{2\rho(1+\nu)}} \end{cases} \quad (1.16)$$

Dans les solides homogènes et isotropes, deux types d'ondes peuvent se propager : les ondes longitudinales et les ondes transversales. Les ultrasons sont donc le résultat d'une combinaison de ces deux modes, en fonction de l'excitation impulsionnelle générée par la sonde. La figure 1.1 illustre ces modes de propagation.

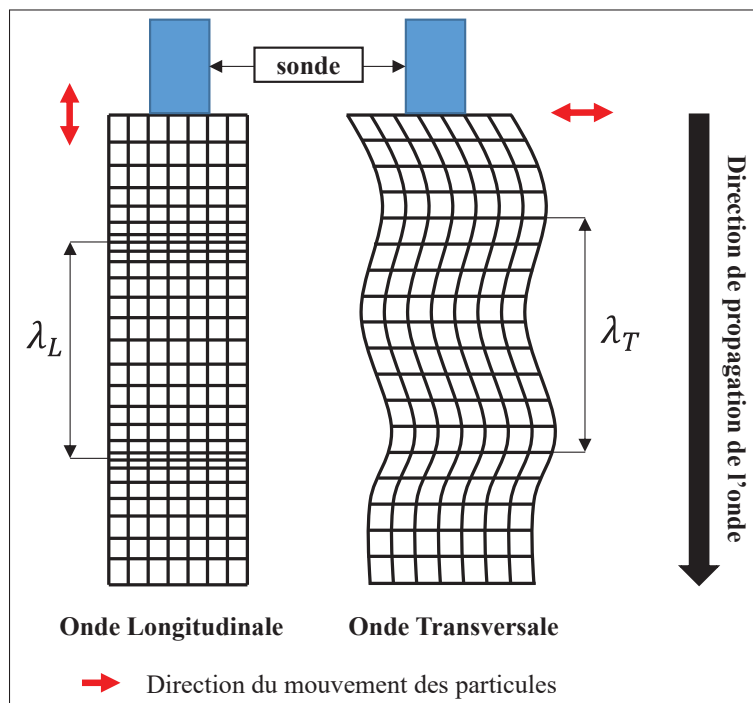


Figure 1.1 Illustration des deux modes de propagation ondulatoires

### 1.1.2 Le phénomène d'atténuation

Lorsque les ultrasons se propagent au sein d'un solide, ils subissent un phénomène d'atténuation régi par le coefficient d'atténuation  $\alpha$ . Ce phénomène résulte de deux effets principaux.

Le premier est l'absorption de l'onde par le solide, qui est proportionnelle au carré de la fréquence de l'onde. Cette absorption dépend également des propriétés thermomécaniques, magnéto-acoustiques et des dislocations du matériau. L'énergie de l'onde est alors absorbée par le solide et convertie en chaleur.

Le second effet est la dispersion de l'onde. La propagation des ultrasons dans un solide repose sur le déplacement des particules du matériau, décrit précédemment. Les impuretés, les joints de grains, les fissures et autres défauts cristallins contribuent à ce phénomène de dispersion, réduisant ainsi l'intensité de l'onde (Wanin, 2001), (Shull, 2002).

Les publications, telles que celles de Cheeke (2002) ou de Shull (2002), décrivent l'atténuation des ultrasons à travers l'intensité acoustique  $I$ . Celle-ci correspond à l'énergie moyenne du flux acoustique traversant une surface  $dA$  pendant un temps  $dt$ . L'intensité acoustique diminue suivant la direction de propagation  $x$  selon une loi exponentielle :

$$I(x) = I_0 e^{-2\alpha x} \quad (1.17)$$

où  $I_0$  représente l'intensité acoustique au niveau de l'émetteur.

Il a été démontré expérimentalement qu'à température ambiante, pour un matériau homogène et isotrope, le coefficient d'atténuation  $\alpha$  varie proportionnellement au carré de la fréquence  $f^2$  (Cheeke, 2002). Cette relation décrit le phénomène d'absorption acoustique, distinct de la dispersion. Sous ces hypothèses, le rapport  $\frac{\alpha}{f^2}$  reste constant. Cette relation met en évidence le rôle central de la fréquence dans l'atténuation des ultrasons et, par conséquent, dans la résolution des acquisitions réalisées pour le contrôle non destructif des matériaux.

La détection des défauts dépend de la longueur d'onde  $\lambda$  de l'onde ultrasonore, qui est inversement proportionnelle à la fréquence. Une fréquence plus élevée permet de détecter des anomalies de plus petite taille, augmentant ainsi la précision du contrôle. Néanmoins, cette augmentation de fréquence entraîne également une atténuation accrue. L'information critique pour l'opérateur peut alors se perdre dans le bruit du signal, ce qui représente un défi majeur dans le domaine du CND par ultrasons (Shull, 2002).

Pour compenser les pertes dues à l'absorption acoustique, une solution intuitive consiste à augmenter l'amplitude de l'onde émise. Toutefois, dans un système linéaire, cette stratégie n'améliore pas le rapport signal sur bruit, car le bruit augmente également avec l'amplitude. De plus, cette approche ne permet pas de contrer les effets de dispersion, qui dégradent le signal de manière significative. En traversant un solide, l'onde ultrasonore interagit avec les irrégularités structurales telles que les défauts cristallins, les porosités ou les variations géométriques. Ces interactions provoquent des réflexions, des réfractions et des conversions de mode (voir section 1.1.3), réduisant l'amplitude des ondes réfléchies utiles. Une partie de l'intensité acoustique est ainsi diffusée hors de la direction d'intérêt, tandis qu'une autre génère des ondes parasites qui viennent perturber le signal utile.

Ce phénomène, nommé « grass » par Shull (2002), voit son amplitude croître proportionnellement avec l'amplitude du signal émis. Pour réduire l'impact de l'atténuation, il est souvent nécessaire de diminuer la fréquence de l'onde, même si cela se fait au détriment de la résolution.

### **1.1.3 Le comportement des ultrasons aux interfaces**

Une fois émise, l'onde ultrasonore se propage, s'atténue et entre en contact avec des changements de milieu. Lorsque l'onde rencontre ces frontières, telles que le fond de la pièce ou des défauts internes, l'onde peut subir plusieurs phénomènes physiques : réflexion, transmission, réfraction ou encore un changement de mode.

Cette section examine en détail ces mécanismes afin de mieux comprendre leurs implications dans le contexte du contrôle non destructif.

### 1.1.3.1 Réflexion et transmission en incidence normale

Commençons par étudier le cas de l'incidence normale (figure 1.2), qui représente le scénario le plus courant dans la mesure d'épaisseur. Pour décrire mathématiquement les phénomènes des ultrasons aux interfaces, les ouvrages de référence tels que ceux de Shull (2002) et Cheeke (2002) utilisent la notion d'impédance acoustique  $Z$ , en analogie avec l'impédance des circuits électriques.

L'impédance acoustique est définie par la relation suivante :

$$Z = \rho v \quad (1.18)$$

où  $\rho$  est la densité massique du matériau et  $v$  la vitesse de phase dans le milieu.

Ces paramètres peuvent être facilement obtenus à partir de tables de données, ce qui simplifie les calculs.

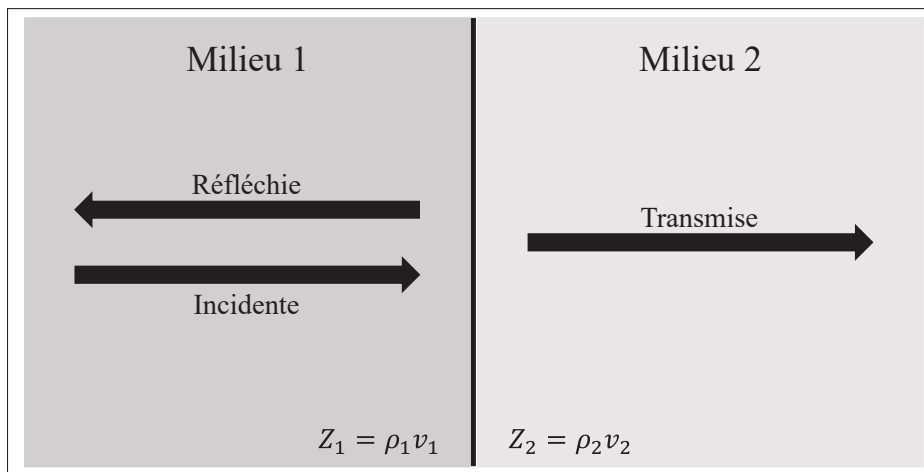


Figure 1.2 Réflexion et transmission d'une onde incidente normale à la frontière

Pour quantifier les proportions d'énergie acoustique transmises et réfléchies lorsqu'une onde ultrasonore rencontre une interface, on définit les coefficients de réflexion  $R$  et de transmission

$T$  en énergie comme suit :

$$\begin{cases} R = \frac{I_r}{I_i} \\ T = \frac{I_t}{I_i} \end{cases} \quad (1.19)$$

où  $I_i$ ,  $I_r$  et  $I_t$  représentent respectivement les intensités acoustiques incidente, réfléchie et transmise au niveau de l'interface.

Il est possible de démontrer que ces coefficients sont donnés par les expressions suivantes :

$$\begin{cases} R = \left( \frac{Z_2 - Z_1}{Z_2 + Z_1} \right)^2 \\ T = \frac{4Z_1 Z_2}{(Z_1 + Z_2)^2} \end{cases} \quad (1.20)$$

Ces relations respectent la conservation de l'énergie, ce qui implique :

$$R + T = 1 \quad (1.21)$$

Cependant, les coefficients de réflexion  $r$  et de transmission  $t$ , exprimés en amplitude, sont plus couramment utilisés en raison de leur simplicité. On peut montrer que :

$$\begin{cases} r = \sqrt{R} = \frac{Z_2 - Z_1}{Z_2 + Z_1} \\ t = \sqrt{\frac{T \cdot Z_2}{Z_1}} = \frac{2Z_2}{Z_1 + Z_2} \end{cases} \quad (1.22)$$

Dans le cas où l'impédance acoustique  $Z_2 \ll Z_1$ , il n'y a pratiquement aucune onde transmise du milieu 1 vers le milieu 2. Ce phénomène est courant pour des interfaces solides-air ou eau-air, comme décrit par Cheeke (2002). Au contraire, si  $Z_2 = Z_1$ ,  $r = 0$  et  $t = 1$ , ce qui signifie que l'onde est totalement transmise sans réflexion.

### 1.1.3.2 Réflexion, réfraction et transmission en incidence oblique

Le cas de l'incidence normale représente une situation idéale et simplifiée, qui n'est généralement pas représentative des conditions réelles. En pratique, les frontières des matériaux ne sont jamais parfaitement planes. Prenons l'exemple d'un bloc d'acier : lors de l'usinage, aucune de ses surfaces ne sera parfaitement plane, ce qui introduira des défauts géométriques susceptibles d'influencer l'angle d'incidence de l'onde transmise.

Dans ces conditions, de nouveaux phénomènes physiques doivent être pris en compte. Comme dans le cas d'une incidence normale, l'onde incidente sera partiellement réfléchie et partiellement transmise. Toutefois, l'onde transmise changera de direction de propagation, un phénomène connu sous le nom de réfraction. De plus, une conversion d'un type d'onde à un autre peut se produire, affectant à la fois l'onde réfléchie et l'onde réfractée. Ce phénomène, appelé conversion de mode (Shull, 2002), consiste, par exemple, en la transformation d'une onde incidente longitudinale en une onde réfractée longitudinale et une onde réfractée transversale, chacune ayant une direction de propagation différente. La figure 1.3 illustre ce phénomène.

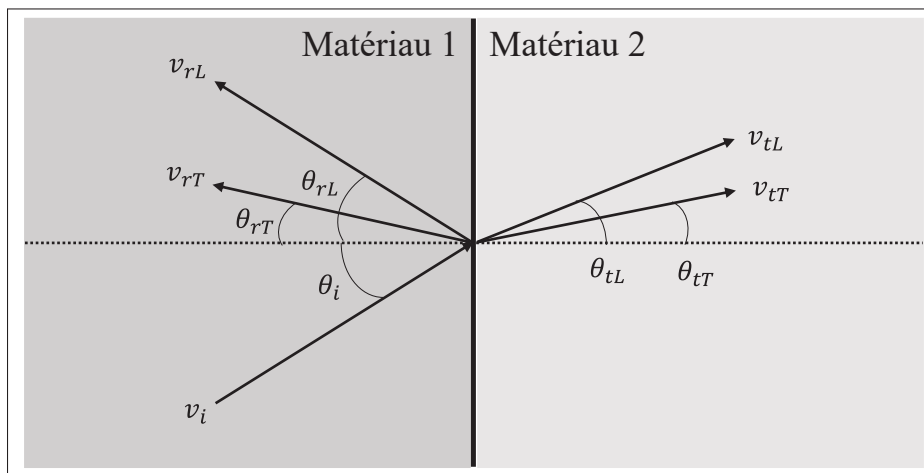


Figure 1.3 Réflexion et réfraction d'une onde incidente longitudinale oblique à la frontière de deux solides avec  $v_i > v_{tL}$

La direction de propagation des ondes réfléchies et réfractées peut être prédite grâce à la loi de Snell (Shull, 2002), exprimée comme suit :

$$\frac{\sin(\theta_i)}{v_i} = \frac{\sin(\theta_{rL})}{v_{rL}} = \frac{\sin(\theta_{rT})}{v_{rT}} = \frac{\sin(\theta_{tL})}{v_{tL}} = \frac{\sin(\theta_{tT})}{v_{tT}} \quad (1.23)$$

Dans cette équation,  $i$ ,  $r$ , et  $t$  désignent respectivement l'onde incidente, réfléchie, et réfractée, tandis que  $L$  et  $T$  indiquent les ondes longitudinales et transversales.  $v$  représente la vitesse de l'onde dans le matériau, et  $\theta$  correspond à l'angle de propagation de l'onde par rapport à la normale à la frontière au point d'incidence.

Les équations pour déterminer l'énergie transmise et réfléchie deviennent plus complexes que dans le cas d'incidence normale (éq. 1.20). Elles dépendent du mode de l'onde incidente, de son angle  $\theta_i$  et de la nature de l'interface (solide-solide, liquide-solide, solide-gaz etc.) (Cheeke, 2002).

Considérons une interface solide-solide où la vitesse de l'onde longitudinale est supérieure dans le milieu 1 par rapport au milieu 2, comme illustré dans la figure 1.3. Les vitesses des ondes transversales étant toujours inférieures à celles des ondes longitudinales (voir éq. 1.15), on obtient donc que  $\theta_T < \theta_L$ . Dans ce cas particulier, quel que soit l'angle d'incidence  $\theta_i$ , quatre ondes résultantes peuvent être observées : une onde réfléchie longitudinale, une onde réfléchie transversale, une onde réfractée longitudinale, et une onde réfractée transversale.

Dans certaines configurations, si la vitesse longitudinale dans le milieu 1 est inférieure à celle du milieu 2, alors  $\theta_i < \theta_{tL}$ . On peut démontrer avec la loi de Snell qu'il existe un angle critique  $\theta_{ic}$ , pour lequel  $\theta_{tL} = 90^\circ$  (Shull, 2002). Au-delà de cet angle critique, l'onde longitudinale réfractée n'existe plus dans le milieu 2. Cette situation est utile pour isoler les ondes transversales.

## 1.2 Le contrôle non destructif par ultrasons

Selon la société américaine du contrôle non destructif (ASNT), le contrôle non destructif (CND) est défini comme « l'inspection d'un objet avec une technologie qui n'affecte pas l'utilisation

future de l'objet » (Shull, 2002). Le CND est omniprésent dans notre quotidien, souvent de manière intuitive et informelle. Par exemple, au supermarché, lorsque nous sélectionnons des fruits, nous inspectons leur apparence, leur couleur ou leur odeur pour évaluer leur qualité. Cela constitue une forme de contrôle non destructif basé sur nos sens. Néanmoins, bien que chaque consommateur utilise des critères similaires, les normes de qualité varient d'une personne à l'autre. Ainsi, même des produits présentant des défauts peuvent être achetés en raison de ce caractère subjectif de l'évaluation.

Dans le domaine industriel, les ultrasons remplacent nos sens pour détecter les défauts au sein d'un objet. L'exemple spécifique de notre étude concerne la mesure d'épaisseur. L'inspecteur, à l'aide des ultrasons, analyse l'onde reçue pour déterminer l'épaisseur du matériau contrôlé. Ce calcul repose sur les paramètres physiques du matériau ( $E$  et  $\nu$ ), ainsi que sur le temps nécessaire pour que l'onde parcoure la distance entre l'émetteur et le récepteur. En fonction des résultats, l'inspecteur décide si la pièce est conforme ou si elle doit être remplacée.

Cependant, comme dans l'exemple du supermarché, la subjectivité intervient dans l'interprétation des données. La détermination du moment exact où l'onde est reçue dépend de l'expérience et des connaissances de l'opérateur.

Bien que des algorithmes aident aujourd'hui les opérateurs à réaliser leur travail, certains cas spécifiques, comme celui de la corrosion, nécessitent une expertise humaine pour interpréter correctement les signaux.

Dans cette section, nous détaillerons les outils et méthodes utilisés par les inspecteurs pour mesurer l'épaisseur avec les ultrasons. Nous examinerons ensuite un cas limite lié à la corrosion, afin de poser les bases du problème principal traité dans ce mémoire.

### **1.2.1 Les sondes à ultrasons**

L'émission et la réception des ondes ultrasonores sont réalisées par des sondes spécifiques. Il en existe quatre types, les sondes piézoélectriques, les sondes électro-magnéto-acoustiques



(EMAT), les transducteur capacitifs et les sondes opto-acoustiques (lasers) (Shull, 2002). Ces sondes reposent sur deux approches distinctes pour générer des ultrasons.

La première méthode, dite "avec contact", utilise des sondes piézoélectriques qui doivent être directement en contact avec la pièce à inspecter. Ces sondes convertissent une excitation électrique en vibrations mécaniques pour produire des ondes ultrasonores.

La seconde méthode, "sans contact", inclut les sondes EMAT et les lasers, qui génèrent des ultrasons en activant directement les particules à la surface de l'objet inspecté. Les sondes EMAT exploitent la force électromagnétique de Lorentz pour générer les vibrations ultrasonores, tandis que les lasers produisent des ondes via un impact thermique localisé, entraînant une expansion rapide de la surface (Wanin, 2001).

Dans ce mémoire, nous concentrerons notre étude sur les sondes piézoélectriques, car elles représentent la technologie la plus couramment utilisée dans l'industrie pour l'émission et la détection des ondes ultrasonores (Shull, 2002).

#### **1.2.1.1 Les sondes mono-élément**

Un palpeur piézoélectrique se compose principalement d'une pastille piézoélectrique, dont l'excitation est réalisée par une tension via l'effet piézoélectrique (Krautkrämer *et al.*, 1977). Une face de protection est interposée entre la pastille et la pièce inspectée afin de protéger le piézo des chocs mécaniques et d'assurer l'étanchéité du boîtier en acier.

Pour optimiser la transmission des ondes ultrasonores, l'épaisseur de cette couche est généralement choisie égale à un quart de la longueur d'onde ( $\lambda/4$ ) dans le matériau de la couche. Cette configuration permet de réduire les réflexions par interférence destructive et de maximiser la transmission d'énergie acoustique vers la pièce inspectée.

Pour limiter les interférences causées par les ondes parasites à l'arrière du palpeur lors de l'émission et de la réception, un bloc amortisseur englobe la pastille et le fil électrique (Wanin, 2001). La figure 1.4 illustre les éléments constitutifs d'une sonde piézoélectrique mono-élément.

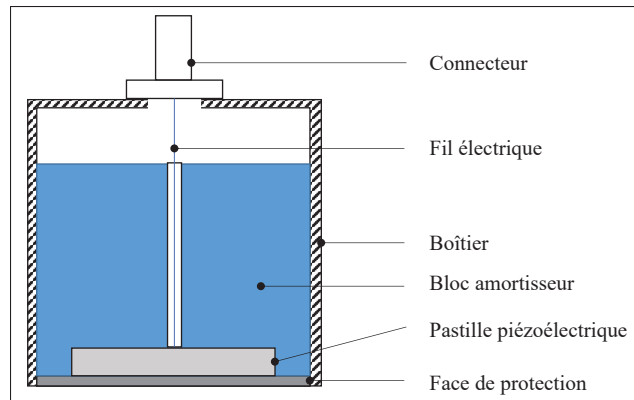


Figure 1.4 Schéma d'une sonde piézoélectrique mono élément

La pastille piézoélectrique joue un double rôle dans les sondes ultrasoniques : elle est capable à la fois d'émettre une impulsion ultrasonore et de la recevoir. En effet, l'énergie acoustique qui retourne sur le palpeur génère des vibrations mécaniques qui excitent le quartz piézoélectrique. Par effet piézoélectrique inverse, la pastille produit alors un signal électrique en réponse à ces vibrations (Krautkrämer *et al.*, 1977). Ce signal est ensuite interprété par le circuit électrique associé (Shull, 2002). Cette propriété permet à la pastille de fonctionner en tant que palpeur émetteur-récepteur.

Les sondes piézoélectriques se distinguent par plusieurs caractéristiques essentielles, notamment leur fréquence d'oscillation ( $F$ ), leur bande passante ( $BW$ ), leurs modes d'excitation (longitudinale ou transversale) et leur diamètre. Le choix de ces paramètres doit être soigneusement adapté à l'application visée afin d'assurer des performances optimales (Krautkrämer *et al.*, 1977).

Enfin, pour émettre une pulsation ultrasonore de la sonde vers le matériau, l'utilisation d'un couplant entre les deux est nécessaire. Comme évoqué dans la partie 1.1.3.1, l'onde ne se transmet pas à travers une interface solide-air car la variation d'impédance acoustique est trop importante. Pour l'émission d'une onde longitudinale, utiliser de l'eau suffit. En revanche, pour transmettre des ondes transversales via un palpeur en contact normal, un couplant spécial à haute viscosité est indispensable (Shull, 2002).

### 1.2.1.2 Les sondes multi-éléments

La sonde mono-élément est limitée au contrôle d'un point. Pour effectuer le contrôle d'une surface, il est donc nécessaire de déplacer la sonde point par point, d'une distance  $dx$  puis  $dy$ , afin de couvrir la zone à inspecter. Afin d'augmenter la productivité et de réduire les incertitudes de déplacement  $dx$ , des sondes multi-éléments sont utilisées (Wanin, 2001).

Ces capteurs multi-éléments sont constitués de plusieurs pastilles piézoélectriques disposées selon un espacement fixe  $dx$ , appelé pitch, et organisées suivant des configurations linéaires, matricielles ou annulaires, selon le cas d'utilisation. Chaque pastille peut être contrôlée individuellement pour émettre ou recevoir un signal. Une représentation schématisée est donnée dans la figure 1.5.

Cette technologie permet de réaliser de l'imagerie en exploitant la capacité des pastilles à fonctionner de manière indépendante. Cependant, dans le cadre de notre étude, qui se concentre sur la mesure d'épaisseur, nous restreignons l'utilisation des sondes multi-éléments à une configuration linéaire dans laquelle chaque élément émet et reçoit son propre signal, de manière autonome. Ainsi, le premier élément traite uniquement le signal qu'il émet, et il en va de même pour les autres éléments constituant la sonde.

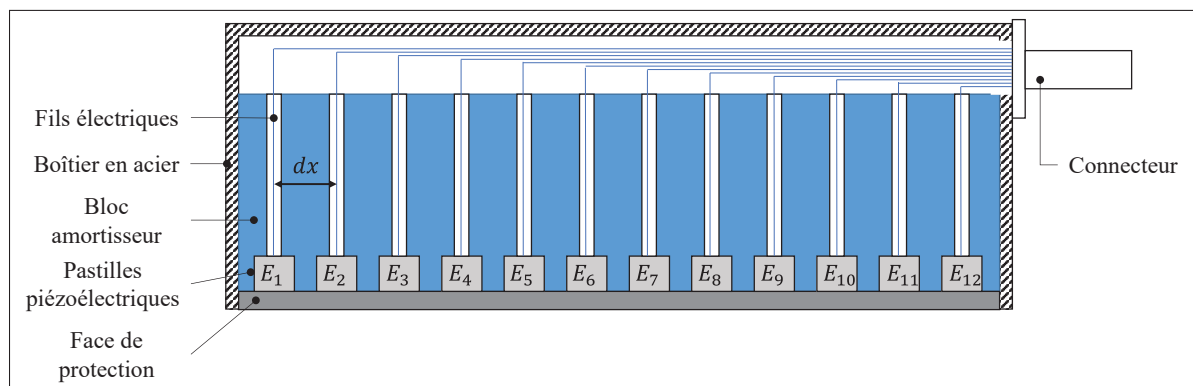


Figure 1.5 Schéma d'une sonde multi-éléments à 12 transducteurs piézoélectriques

Le fonctionnement des sondes multi-éléments repose sur un principe séquentiel. L'élément  $E_1$  (figure 1.5) commence par émettre une pulsation ultrasonore, puis il se reconfigure pour agir en tant que récepteur. Après un délai  $\Delta t$ , l'élément  $E_2$  émet à son tour une pulsation. Ce processus se poursuit pour chaque élément de la sonde, permettant ainsi d'obtenir, à partir d'une seule acquisition, 12 signaux distincts dans une direction.

Le choix du délai  $\Delta t$  est déterminé de manière à éviter toute superposition temporelle des signaux issus des différents éléments. Il est généralement fixé en fonction de la durée du signal émis (environ 2 à 3 fois la durée de la pulse) et du temps de vol aller-retour dans le matériau inspecté. Ce délai dépend donc de la vitesse de propagation de l'onde dans le matériau, de la profondeur maximale à inspecter, et du type d'onde utilisée.

### **1.2.2 Le A-scan, B-scan et C-scan**

La méthode ultrasonore utilisée pour mesurer l'épaisseur des matériaux dans cette étude est le « pulse-echo », où la sonde agit comme émetteur et récepteur (Shull, 2002).

Positionnée sur le matériau à l'aide d'un couplant, la sonde émet une onde ultrasonore lorsqu'une impulsion électrique excite les pastilles piézoélectriques. Cette onde, appelée impulsion d'émission, se propage dans le matériau, s'atténue et se réfléchit sur les défauts ou le fond de la pièce. Une partie de cette onde revient à la sonde sous forme d'échos de réflexion, qui sont visualisés sur un oscilloscope pour analyser les propriétés du matériau (Wanin, 2001).

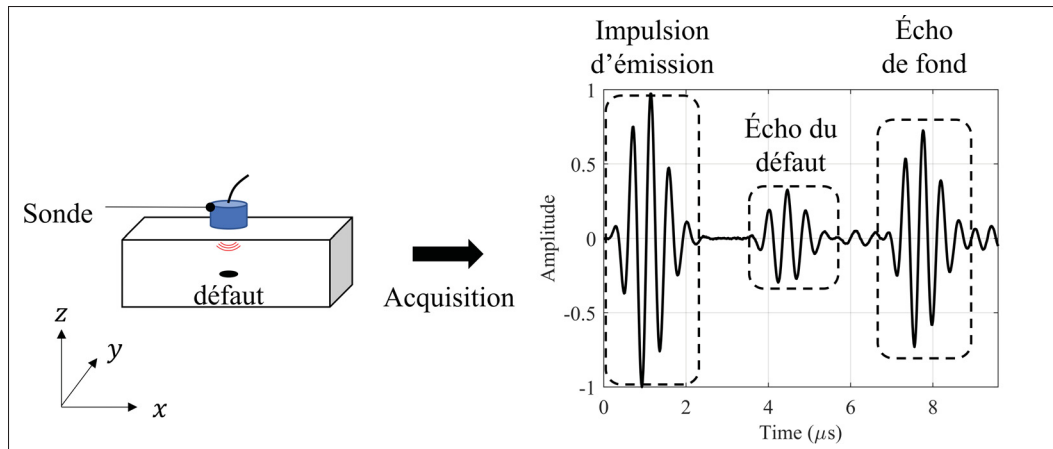


Figure 1.6 Schéma exemple d'un A-scan

Grâce à une intensité initiale suffisante, plusieurs échos peuvent être observés malgré une atténuation exponentielle. Une mesure unique, appelée A-scan, correspond à l'émission et à la réception d'une impulsion à un point spécifique (Fig. 1.6) (Shull, 2002).

Le B-scan résulte de plusieurs A-scans obtenus le long d'un axe de balayage donné (Fig. 1.7) (Shull, 2002).

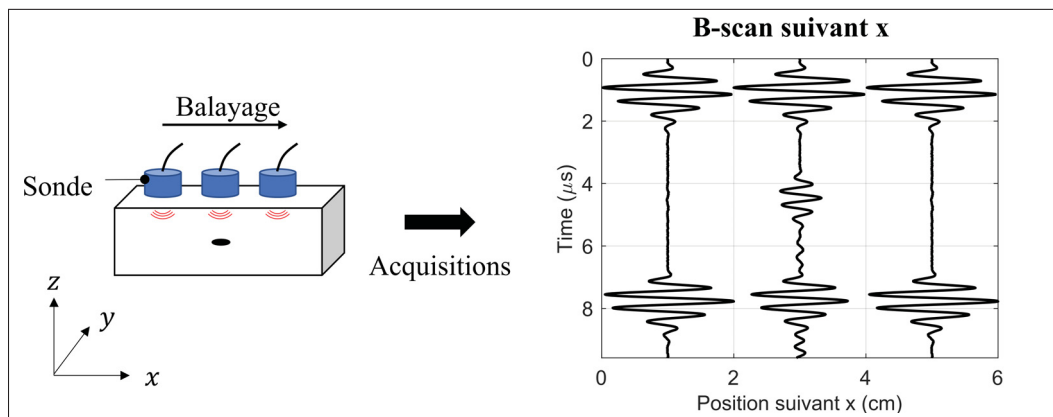


Figure 1.7 Schéma exemple d'un B-scan

Le C-scan permet l'inspection complète d'une surface en combinant une série de B-scans acquis le long d'un axe de déplacement. Chaque B-scan fournit une coupe transversale du

matériau selon un plan  $(x, z)$  ou  $(y, z)$ , et l'ensemble des coupes successives, acquises pas à pas perpendiculairement à ce plan, permet de reconstituer une image en deux dimensions  $(x, y)$  de la surface inspectée. Cette méthode fournit une vue d'ensemble détaillée de la zone inspectée, permettant d'évaluer globalement la conformité de l'épaisseur de l'objet. Le C-scan est généralement représenté sous forme de cartographie 2D en couleur ou en niveau de gris, suivant une échelle d'épaisseur, ou d'amplitude (Fig. 1.8) (Shull, 2002).

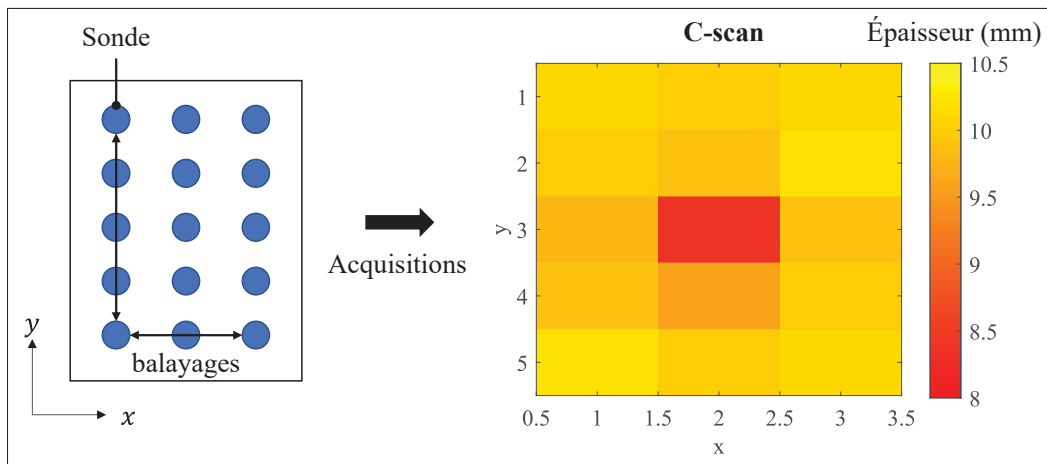


Figure 1.8 Schéma exemple d'un C-scan

### 1.2.3 Le temps de vol

Le temps de vol correspond au temps de transit de l'impulsion ultrasonique entre son émission et sa réception par la sonde (Shull, 2002). Sur un A-scan, il s'agit de l'intervalle de temps séparant deux échos (figure 1.9). Connaissant la vitesse  $V$  de propagation de l'onde dans le matériau (longitudinale ou transversale), l'épaisseur  $d$  de la pièce peut être déterminée à partir du temps de vol  $ToF$  selon la relation suivante :

$$d = \frac{V \cdot ToF}{2} \quad (1.24)$$

L'onde parcourant un aller-retour dans la pièce, le temps de vol représente le double de l'épaisseur.

Bien que d'autres méthodes existent pour évaluer l'épaisseur en utilisant des paramètres comme l'amplitude — via l'atténuation du signal —, la phase — qui varie avec la distance parcourue —, ou encore la fréquence — par l'analyse des résonances — (Shull, 2002), cette étude se concentre sur la détermination du temps de vol, méthode privilégiée par nos partenaires industriels.

Le calcul précis du temps de vol à partir d'un A-scan est un enjeu central. Plusieurs approches analytiques existent, mais elles présentent certaines limitations, qui seront discutées en détail dans la section 1.3. Un exemple typique de mesure est illustré figure 1.9.

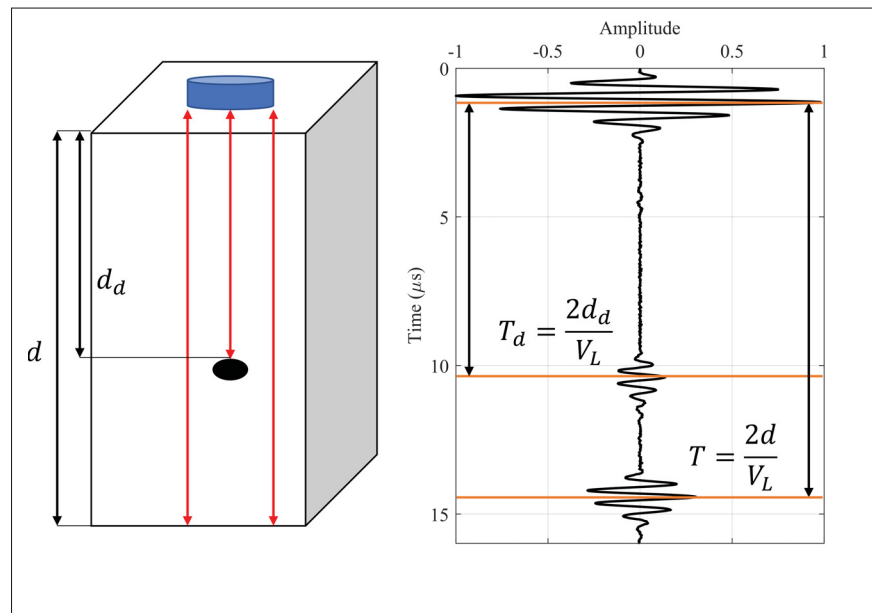


Figure 1.9 Exemple de détermination du temps de vol grâce au maximum des échos

Le temps de propagation  $T$  et  $T_d$  est ici calculé à partir de la différence temporelle entre le maximum de l'impulsion initiale et du premier écho.

#### 1.2.4 Le phénomène de superposition des échos

Lorsque le temps de vol entre les défauts ou le fond de la pièce et le récepteur dépasse la largeur de l'impulsion ultrasonore (comme illustré à la figure 1.9), il est facile de séparer les échos et de

mesurer les différents temps de vol. En revanche, si le temps de vol est inférieur à la largeur de l'impulsion, les échos se superposent, rendant difficile leur séparation et la mesure précise des temps de propagation (Shpigler, Mor & Bar-Hillel, 2022).

Le phénomène de superposition des échos est illustré dans la figure ci-dessous.

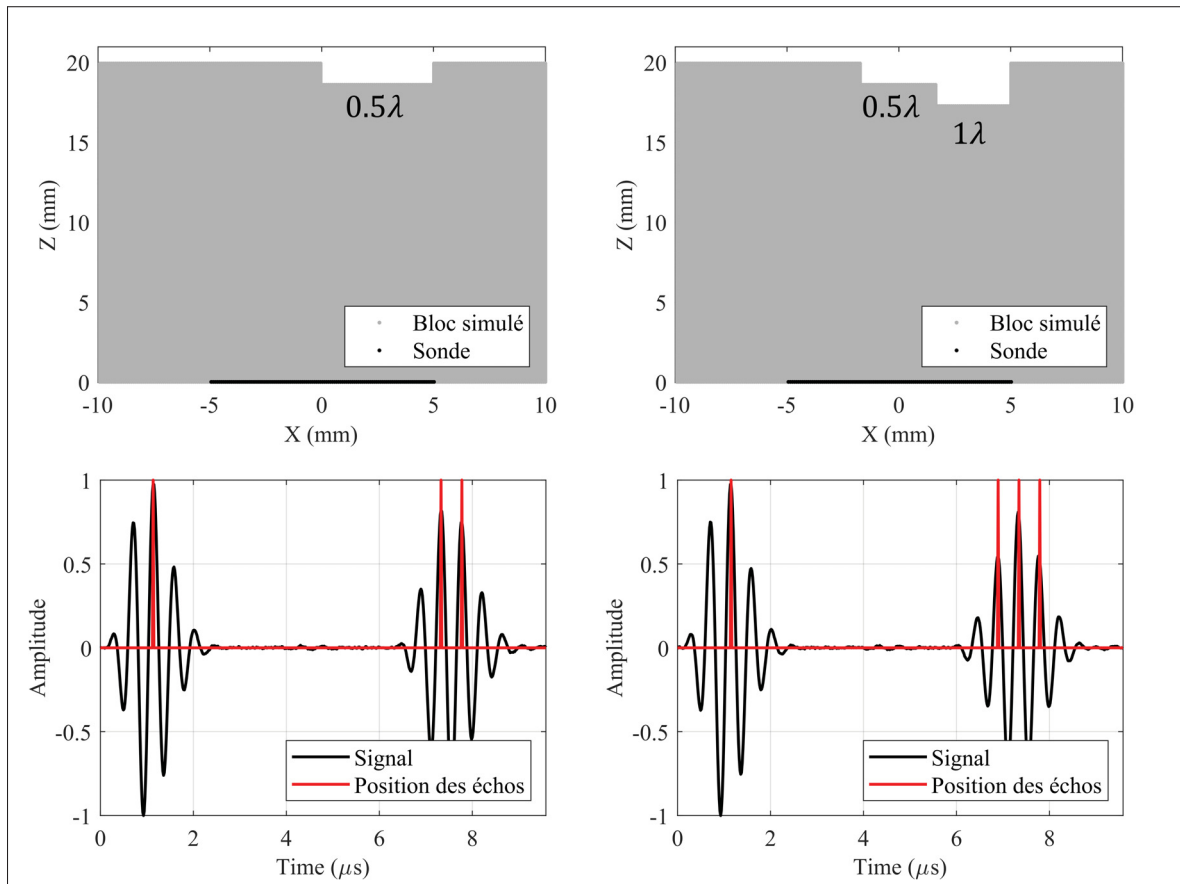


Figure 1.10 Exemples d'A-scans avec des échos superposés.

Les exemples sur la figure 1.10 sont simplifiés, car l'amplitude de chaque écho est sensiblement la même. En effet, l'onde se réfléchit sur des surfaces de même dimension, donc l'intensité réfléchie est la même pour chaque fond de pièce. Toutefois, cette configuration est peu fréquente dans la réalité. Les études menées par Shpigler *et al.* (2022) soulignent que, dans des conditions plus authentiques, l'amplitude et la forme des différents échos fluctuent car les caractéristiques des surfaces sur lesquelles ces derniers se réfléchissent varient.



Deux approches peuvent être envisagées pour répondre au problème de superposition.

La première est d'accroître la fréquence d'émission de l'onde pour réduire la largeur d'impulsion et augmenter la résolution. Cependant, cette méthode comporte deux limites majeures : l'atténuation, qui croît proportionnellement au carré de la fréquence, et l'effet de dispersion.

Lorsque la fréquence augmente, la longueur d'onde diminue, rendant l'onde plus sensible aux hétérogénéités du matériau (telles que grains, pores ou inclusions). Ces interactions avec les irrégularités locales peuvent provoquer des diffusions multiples et des changements de mode d'onde, générant ainsi des interférences parasites qui dégradent l'information utile (Mor, Azoulay & Aladjem, 2010).

La deuxième alternative repose sur des techniques d'analyse numérique, telles que les algorithmes de traitement du signal ou l'utilisation d'intelligences artificielles entraînées pour des cas spécifiques (Shpigler *et al.*, 2022). Cela permet d'atteindre une résolution axiale de l'ordre de  $0.5\lambda$  (Chapon, Pereira, Toews & Belanger, 2021). La section 1.3 présente ces méthodes en détail.

### **1.2.5 L'inspection de pièces corrodées**

L'inspection de l'épaisseur d'une pièce devient complexe lorsque la nature de la surface réfléchissante est inconnue, comme dans le cas des tubes ou réservoirs de pétrole. Une surface corrodée peut alors introduire de fortes incertitudes sur l'épaisseur mesurée par ultrasons. En effet, la rugosité accrue entraîne une forte superposition des échos, rendant leur analyse difficile sans un traitement avancé. Pour répondre aux normes de sécurité, il est crucial de déterminer la distance minimale, mais ce paramètre se retrouve noyé dans les multiples échos superposés de différentes amplitudes. Cela contraint les inspecteurs à recourir à des méthodes alternatives pour décider si une pièce doit être remplacée (Cantero-Chinchilla, Simpson, Ballisat, Croxford & Wilcox, 2023).

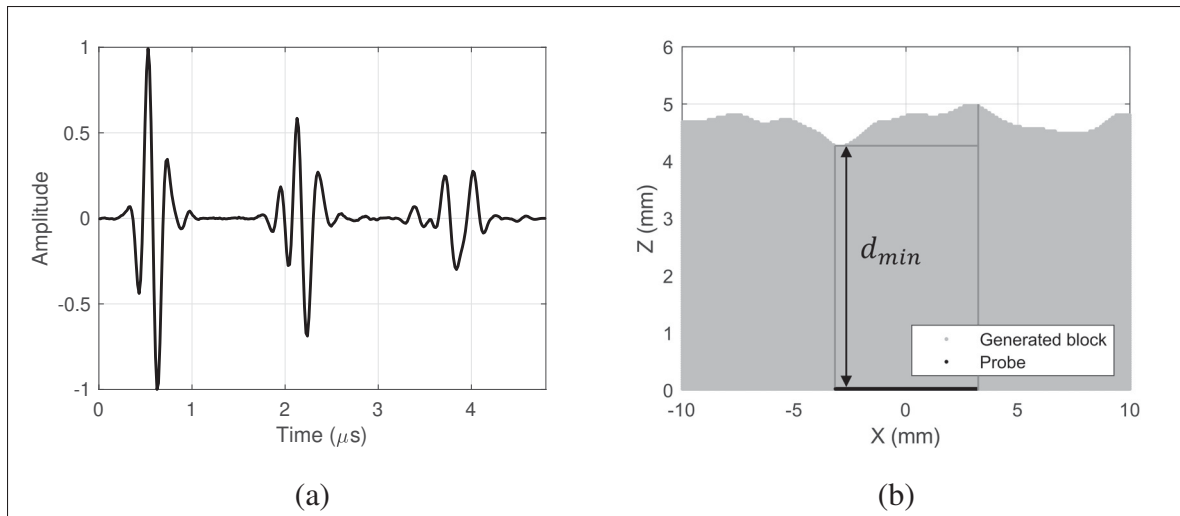


Figure 1.11 Simulation d'un A-scan (a) sur un bloc corrodé (b).

La figure 1.11 illustre le cas d'un bloc fortement corrodé, où l'écho d'émission n'est plus discernable des deux échos de réception. Dès le premier écho, l'amplitude du signal est significativement réduite, même après un faible parcours de l'onde. Cette atténuation s'explique par la faible fraction de l'intensité initiale qui parvient à se réfléchir efficacement vers la sonde.

Quant au second écho, une forte superposition avec d'autres réflexions déforme considérablement sa signature, le rendant difficilement comparable au premier.

Étant donné que seule la réception est exploitée pour la détermination de l'épaisseur, il devient particulièrement complexe d'identifier avec précision le temps de propagation associé à la distance minimale ( $d_{min}$ ). Cet exemple illustre les défis majeurs abordés dans ce mémoire.

### 1.3 Les méthodes de mesure d'épaisseur

Pour mesurer l'épaisseur d'une pièce en un point donné, il est nécessaire de connaître la vitesse de propagation de l'onde, longitudinale ou transversale, et son temps de parcours, déterminé à partir de l'A-scan. Comme évoqué précédemment, la principale difficulté réside dans l'estimation précise du temps de vol.

Lorsque les échos sont clairement distincts, des algorithmes simples suffisent pour obtenir des résultats fiables. En revanche, dans le cas de superpositions, il devient difficile d'extraire les temps de propagation à partir du signal brut. Différentes méthodes algorithmiques plus complexes existent pour différencier et séparer les échos. Toutefois, face à des cas plus délicats, comme celui de la corrosion, les motifs à distinguer s'avèrent beaucoup plus subtiles. L'intelligence artificielle a alors été introduite avec succès, offrant souvent des résultats plus fiables que les algorithmes analytiques développés jusqu'alors (Shpigler *et al.*, 2022).

Dans cette section, nous retracerons l'évolution des méthodes employées pour la détermination du temps de vol, afin de souligner la nécessité d'intégrer des architectures innovantes répondant aux défis actuels.

### **1.3.1 La méthode de seuil**

La méthode de seuil, ou « thresholding », est l'une des premières techniques employées pour obtenir le temps de vol dans le monde de l'industrie. Encore utilisée aujourd'hui, cette méthode consiste à détecter l'écho lorsque son amplitude dépasse un seuil prédéfini (Marioli *et al.*, 1992). Dès 1974, elle était implémentée à l'aide de circuits électroniques (Frederiksen & Howard, 1974). Bien que simple et économique, cette méthode manque de précision, notamment en cas de forte atténuation entre les échos, comme illustré figure 1.12a (Marioli *et al.*, 1992).

Pour améliorer la précision, Fox, Khuri-Yakub & Kino (1983) ont proposé un seuil ajustable, par exemple en détectant chaque écho lorsque son amplitude dépasse un quart de son maximum (figure 1.12b). Toutefois, ces méthodes montrent rapidement leurs limites face à des signaux superposés, où la distinction entre échos devient difficile.

Pour répondre à cette problématique, Audoin & Roux (1996) ont introduit l'utilisation de l'enveloppe du signal grâce à la transformée d'Hilbert (Parrilla, Anaya & Fritsch, 1991). Leur méthode repose sur la corrélation croisée entre échos pour extraire le temps de vol, ce qui permet d'utiliser l'ensemble de la forme d'onde et améliore la précision, en particulier lorsque les signaux sont bruités.

Alternativement, il est également possible d'estimer le temps de vol en mesurant le délai entre les maximums successifs de l'enveloppe (figure 1.12c). Cette approche est plus simple à mettre en œuvre, mais elle est généralement plus sensible au bruit et aux déformations locales du signal, ce qui peut affecter la fiabilité de la mesure.

Poursuivant ces travaux, Egaña, Seco & Ceres (2008) ont étudié le seuil optimal et une normalisation sélective de l'enveloppe pour améliorer la précision. Malgré sa robustesse, cet algorithme reste limité pour les signaux superposés.

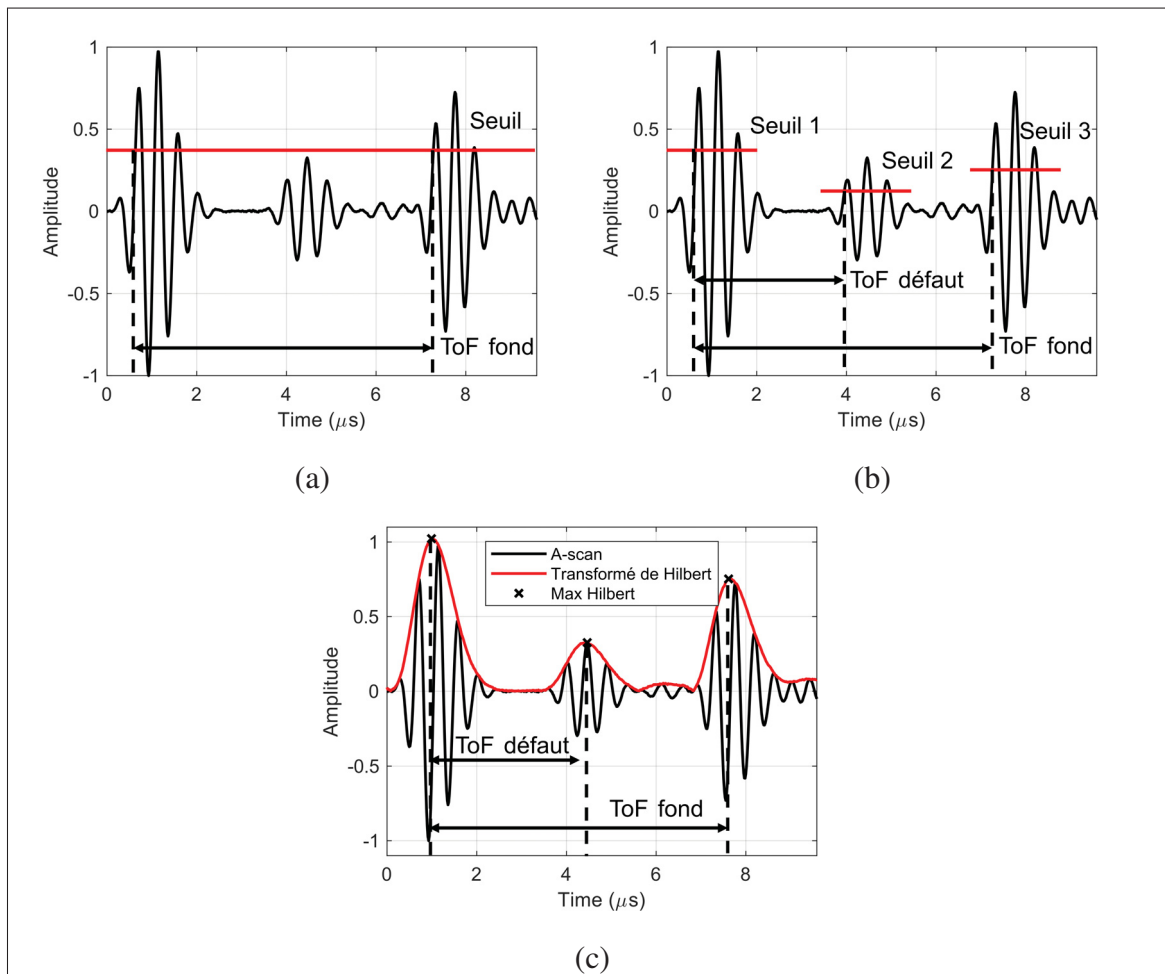


Figure 1.12 Représentation de la méthode de (a) seuil fixe, (b) seuil variable, et (c) maximum de l'enveloppe.

### 1.3.2 La corrélation croisée

La corrélation croisée a été développée pour réduire les erreurs significatives associées à la méthode de seuil, notamment dans des conditions de faible rapport signal/bruit (SNR) ou de fréquence d'échantillonnage réduite (Queiros, Correa Alegria, Silva Girao & Cruz Serra, 2010). Le SNR, ou « Signal to Noise Ratio », est défini comme suit :

$$SNR_{dB} = 20 \cdot \log_{10} \left( \frac{A_{signal}}{A_{bruit}} \right) \quad (1.25)$$

où  $A$  représente l'amplitude du signal ou du bruit.

Les travaux de Parrilla *et al.* (1991) et Marioli *et al.* (1992) ont introduit cette méthode pour la mesure du temps de vol. Son principe repose sur la corrélation croisée entre le signal émis  $e(t)$  et le signal reçu  $r(t)$ , ou entre deux échos reçus. Le maximum de cette corrélation croisée correspondant à une approximation du temps de vol (Parrilla *et al.*, 1991). En pratique, la forme exacte du signal émis est inconnue. Le premier écho est donc utilisé comme référence pour corréler avec le reste du signal. La corrélation croisée pour des signaux discrets s'exprime par :

$$s(k) = \sum_{i=1}^N e(k+i) \cdot r(i) \quad (1.26)$$

et le temps de vol peut être estimé en mesurant l'intervalle entre les indices des deux maximums successifs de la corrélation croisée  $s$ , puis en divisant par la fréquence d'échantillonnage  $F_{ech}$  :

$$ToF_{correlation} == \frac{|i_{\max,2} - i_{\max,1}|}{F_{ech}} \quad (1.27)$$

où  $i_{\max,1}$  et  $i_{\max,2}$  désignent respectivement les indices des premier et second maximums détectés sur la corrélation croisée.

La corrélation croisée est efficace à faible SNR car elle exploite à la fois l'amplitude et la phase du signal pour estimer le temps de vol (Ferrara, Algazi & Liu, 1992). Même si la corrélation

nécessite un signal de référence, ce dernier peut être extrait d'une zone saine ou d'un échantillon similaire (Hoseini, Wang & Zuo, 2012). Toutefois, en cas de forte superposition, la déformation des échos peut altérer la précision de la corrélation et fausser l'estimation du délai.

### 1.3.3 Le Matching Pursuit (MP)

En 1993, Mallat & Zhang (1993) proposent de décomposer le signal en une combinaison linéaire parcimonieuse d'ondelettes, sélectionnées dans un dictionnaire afin de correspondre au mieux à la structure du signal. Cette approche, appelée Matching Pursuit, permet de déconstruire et d'interpréter plus aisément le signal, mais son algorithme initial requiert des ressources computationnelles importantes.

Pour réduire cette complexité, Tropp & Gilbert (2007) introduisent l'« Orthogonal Matching Pursuit » (OMP), qui réduit la complexité algorithmique de quadratique à logarithmique. En 2010, Mor *et al.* (2010) élaborent le « Support Matching Pursuit » (SMP), une méthode optimisée pour la décomposition des échos ultrasonores superposés, intégrant les phénomènes physiques sous-jacents au signal.

Bien que ces méthodes basées sur des dictionnaires se révèlent efficaces, elles présentent une limite majeure : l'augmentation de la variance des formes d'échos nécessite l'élargissement du dictionnaire. Une taille accrue ralentit l'algorithme et accroît le risque de fausses détections (Shpigler *et al.*, 2022).

### 1.3.4 La déconvolution

Pour résoudre les problèmes de résolution engendrés par la superposition des échos, Carpenter & Stepanishen (1984) ont introduit la déconvolution des échos pour le CND par ultrasons.

En termes discrets, le problème peut être formulé comme suit (Sin & Chen, 1992). Soit  $x(k)$  une référence du signal émis,  $y(k)$  le signal pulse-écho reçu, et  $n(k)$  le bruit. La relation entre

ces variables est décrite par :

$$y(k) = x(k) * h(k) + n(k) \quad (1.28)$$

où  $h(k)$  représente la réponse impulsionnelle, et «  $*$  » le produit de convolution, défini discrètement par :

$$x(k) * h(k) = \sum_{i=-\infty}^{+\infty} h(k-i) \cdot x(i) \quad (1.29)$$

La figure 1.13 illustre l'effet de ce produit convolutif entre le signal émis et la réponse impulsionnelle. L'objectif est de déterminer  $h(k)$ , qui traduit la position et l'amplitude des échos. La déconvolution permet alors d'isoler la réponse impulsionnelle, où chaque impulsion correspond à un écho distinct. Cela facilite la séparation des échos superposés et l'identification précise de leurs amplitudes.

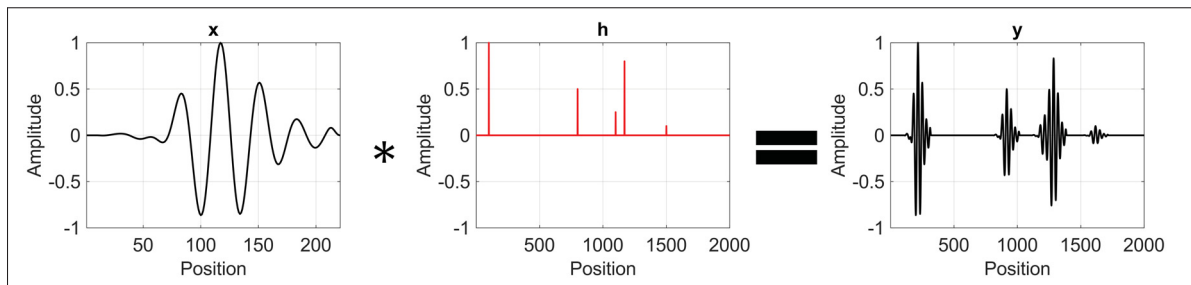


Figure 1.13 Exemples de l'effet de convolution.

La méthode la plus naïve consiste à filtrer le bruit  $n$  ou à le considérer comme négligeable face à  $x$ , puis d'effectuer la transformée de Fourier du signal reçu approximatif  $\tilde{y}$ .

$$\tilde{Y}(k) = FFT(x(k) * h(k)) = X(k) \cdot H(k) \quad (1.30)$$

La réponse impulsionnelle  $h$  est alors obtenue par transformée de Fourier inverse :

$$h(k) = IFFT \left( \frac{\tilde{Y}(k)}{X(k)} \right) \quad (1.31)$$

Cette méthode, appelée filtrage inverse (Biraud, 1976), présente toutefois des limitations significatives. En effet, la moindre variation dans la réponse du système peut amplifier fortement les erreurs. De plus, lorsque le bruit est principalement dû à la diffusion des ultrasons dans le matériau, il partage inévitablement la même bande passante que le signal utile, rendant son élimination particulièrement difficile. Cela limite la résolution de  $h$ , surtout lorsque le SNR est faible (Biraud, 1976). De plus, l'incertitude sur le signal émis  $x$ , souvent approximatif, introduit des erreurs supplémentaires dans la déconvolution.

Pour augmenter la résolution de la déconvolution, divers algorithmes de traitement du signal ont été développés, notamment dans des domaines tels que la sismologie et la géologie. Sin & Chen (1992) ont adapté certaines de ces techniques pour le CND par ultrasons, obtenant des résultats prometteurs avec le filtre de Wiener et la méthode des moindres carrés  $L_2$ .

Les progrès dans la puissance de calcul des ordinateurs ont permis l'émergence d'algorithmes plus avancés, comme les déconvolutions « blind » (Nandi, Mampel & Roscher, 1997) ou « sparse » (Wei, Huang & Que, 2009), capables d'atteindre une résolution allant jusqu'à  $0,68\lambda$ , où  $\lambda$  représente la longueur d'onde de l'onde ultrasonore (Soussen, Idier, Carcreff, Simon & Potel, 2012). Cependant, ces méthodes restent sensibles à un faible SNR, ce qui limite leur efficacité dans des conditions extrêmes, telles que celles associées aux matériaux fortement corrodés.

Les recherches récentes se tournent vers l'Intelligence Artificielle (IA), notamment avec l'utilisation des réseaux de neurones. Ces derniers visent à approximer la fonction permettant de passer du signal reçu à sa déconvolution, en s'appuyant sur des paramètres appris grâce à un entraînement supervisé avec des données labellisées. Par exemple, Chapon *et al.* (2021) ont introduit une architecture de réseau de neurones convolutifs capable de déconvoluer la superposition de deux échos avec un SNR allant jusqu'à 5 dB. Leur IA atteint une résolution de



0,65 $\lambda$  pour un SNR de 20 dB et de 0,5 $\lambda$  pour un SNR de 40 dB. Toutefois, cette approche reste limitée à la détection de 2 échos pour un signal, une fréquence et un type de bruit spécifiques, ce qui réduit sa flexibilité et son applicabilité en milieu industriel.

Shpigler *et al.* (2022) ont étendu leur architecture d'IA et enrichi leur base de données en se concentrant sur la détection d'un nombre aléatoire d'échos superposés, avec des amplitudes et des formes variées. Ils ont démontré que leur IA était plus robuste que les algorithmes de Matching Pursuit développés jusqu'alors, en particulier dans des situations de forte superposition d'échos. Leur jeu de données repose sur une superposition d'échos gaussiens de formes différentes pour chaque A-scans. Toutefois, chaque écho généré dans un A-scan donné voit uniquement son amplitude modifiée avant d'être superposé. Cela limite leur capacité à simuler les variations réelles de la forme des échos, résultant des diffusions et atténuations que subit le signal lorsqu'il interagit avec des irrégularités dans la pièce.

Dans une autre approche, Cantero-Chinchilla *et al.* (2023) ont proposé une nouvelle architecture IA visant à estimer la distance minimal ou moyenne entre une surface corrodée et la sonde. Leur base de données repose cette fois sur des A-scans simulés numériquement à partir de surfaces corrodées, avec des signaux émis générés aléatoirement en respectant des paramètres réalistes. Bien que les résultats obtenus sur ces données simulées soient excellents, leur IA présente des limites lorsqu'elle est appliquée à des données réelles. Ce problème résulte du manque de données expérimentales labellisées dans leur base de données d'entraînement.

En dépit des progrès réalisés dans l'application de l'intelligence artificielle pour la détermination de l'épaisseur des matériaux par ultrasons, aucune approche actuelle n'a encore démontré une robustesse suffisante pour traiter des cas complexes tels que ceux rencontrés en présence de corrosion.

#### **1.4 Les réseaux de neurones et leurs architectures**

Notre étude vise à développer une architecture d'IA accompagnée d'une base de données labellisées dédiée à la déconvolution des signaux ultrasonores. Comme l'impulsion se déforme

au cours de la propagation en raison de l'atténuation, de la diffusion et de la superposition d'échos, notre modèle intégrera une dépendance temporelle via l'ajout du contexte global, afin de mieux tenir compte de l'évolution de la forme du signal en fonction du délai d'écho.

L'objectif est de permettre une détermination précise de l'épaisseur des matériaux, y compris dans des situations complexes comme celles où les surfaces sont corrodées. Dans cette partie, nous introduirons les concepts fondamentaux de l'IA qui serviront à concevoir une architecture répondant spécifiquement à ces besoins.

#### 1.4.1 Les fondements de l'apprentissage profond

L'apprentissage profond est une sous-catégorie de l'apprentissage machine. Mitchell (1997) donne la définition suivante : un programme apprend d'une expérience  $E$  pour accomplir une tâche  $T$  avec une performance mesurée  $P$ . Ainsi, la tâche  $T$ , évaluée par  $P$ , s'améliore grâce à l'expérience  $E$ .

Dans notre étude, la tâche  $T$  correspond à la déconvolution du signal ultrasonore. La performance  $P$  de cette tâche est mesurée par une fonction de perte, qui évalue l'écart relatif entre le résultat attendu et celui obtenu en sortie de l'IA. L'expérience  $E$  est représentée par un ensemble de A-scans labellisés, où chaque A-scan est associé au résultat déconvolué attendu. Ce processus relève d'un apprentissage dit supervisé (Goodfellow, Bengio & Courville, 2016).

L'objectif principal est d'approximer une fonction  $f^*$  capable de répondre à la tâche  $T$  le plus précisément possible. Si  $\vec{x}$  représente le vecteur associé à un A-scan, nous cherchons à approximer  $f^*$  de telle manière que  $\vec{y} = f^*(\vec{x})$  corresponde à la déconvolution de  $\vec{x}$ .

#### 1.4.2 Le réseau de neurones à propagation avant (FFN)

Le premier modèle que nous allons présenter est le réseau de neurones à propagation avant, ou « Feed Forward Network » (FFN), qui constitue l'une des bases de l'apprentissage profond (Goodfellow *et al.*, 2016).

Un réseau de neurones à propagation avant est composé de plusieurs neurones, chacun représentant une fonction. Ces neurones sont connectés de manière unidirectionnelle, d'où le nom de propagation avant. Le réseau se forme en superposant une série de couches. La première couche est la couche d'entrée, tandis que la dernière correspond à la couche de sortie. Les couches situées entre l'entrée et la sortie sont appelées les couches cachées. Un exemple de ce type de réseau est illustré dans la figure 1.14.

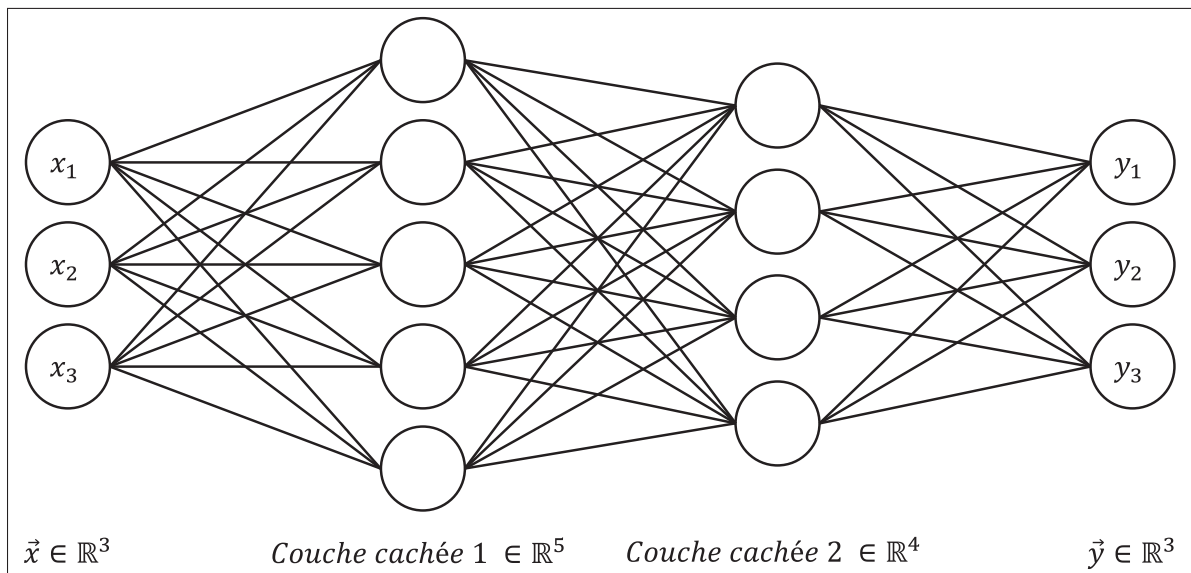


Figure 1.14 Schéma d'un réseau de neurone à propagation avant.

Chaque neurone suit le modèle du perceptron, développé par Rosenblatt (1958). Ces neurones représentent des fonctions, appelées fonctions d'activation, avec un biais. Les connexions entre les neurones sont associées à des poids. La fonction d'activation est une fonction non linéaire qui permet d'établir une transformation non linéaire entre l'entrée  $\vec{x}$  et la sortie  $\vec{y}$ .

Prenons l'exemple d'un neurone  $i$  de la couche 1. Soit  $g_i^1$  sa fonction d'activation,  $b_i^1$  son biais, et  $w_{ij}^1$  les poids des connections associées aux entrées  $x_j$  avec  $j \in [1, 3]$ . La sortie du neurone est alors donnée par :

$$y_i^1 = g_i^1 \left( b_i^1 + \sum_{j=1}^3 w_{ij}^1 \cdot x_j \right) \quad (1.32)$$

Le nombre de paramètres à déterminer pour approximer la fonction  $f^*$  peut alors devenir conséquent si la dimension des entrées et des sorties est élevée. Dans notre cas, un A-scan contient 1200 points. Ce nombre est défini en fonction de la fréquence d'échantillonnage, de la vitesse de propagation de l'onde dans le matériau, de la taille maximale des échos et de l'épaisseur maximale de la pièce inspectée. Avec un réseau composé de trois couches de même dimension, environ 5 760 000 paramètres devraient être ajustés. Ainsi, ce type de réseau seul ne convient pas pour traiter efficacement les A-scans (LeCun & Bengio, 1995).

### **1.4.3 Les couches intermédiaires**

Avant de présenter les autres architectures, il est important de comprendre certains concepts clés liés aux couches intermédiaires d'un réseau de neurones. Ces concepts incluent les fonctions d'activation, la normalisation en batch ou en layer, ainsi que le dropout. Chacune de ces techniques joue un rôle essentiel dans le fonctionnement et l'efficacité d'un réseau de neurones.

#### **1.4.3.1 Les fonctions d'activation**

Comme mentionné précédemment, les fonctions d'activation sont utilisées pour introduire une non linéarité dans la fonction  $f^*$  que nous cherchons à approximer. Elles jouent également un rôle crucial dans la rétropropagation du gradient (Dubey & Jain, 2019), un concept que nous aborderons dans une prochaine partie. Bien que les fonctions d'activation soient souvent associées aux neurones, il est plus précis de les considérer comme une couche intermédiaire entre chaque couche de neurones.

Diverses fonctions d'activation sont utilisées en fonction de l'application, parmi lesquelles on trouve la tangente hyperbolique (Tanh), la sigmoïde, la ReLU et ses variantes, ainsi que la fonction linéaire ou Softmax.

L'une des fonctions d'activation les plus célèbres est la « Rectified Linear Unit » (ReLU), introduite par Hahnloser, Sarpeshkar, Mahowald, Douglas & Seung (2000) pour les circuits électriques, puis adoptée dans les réseaux neuronaux à partir de 2011 (Agarap, 2018). La fonction

ReLU est définie comme suit :

$$ReLU(z) = \begin{cases} 0 & \text{si } z \leq 0 \\ z & \text{si } z > 0 \end{cases} \quad (1.33)$$

Cependant, la fonction ReLU présente un inconvénient majeur : elle peut provoquer la "mort" de certains neurones. En effet, pour les entrées négatives, la sortie est nulle, empêchant ainsi la réactivation du neurone pendant l'entraînement (Lau & Hann Lim, 2018).

La seconde fonction d'activation d'intérêt est la tangente hyperbolique, définie par :

$$Tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1} \quad (1.34)$$

La tangente hyperbolique est une fonction logistique qui présente une forte similitude avec le taux d'activation des neurones biologiques. Elle offre un bon compromis entre linéarité et non-linéarité dans le comportement du réseau (Lau & Hann Lim, 2018). Contrairement à la fonction sigmoïde, la fonction Tanh est centrée sur zéro, ce qui permet d'éviter le ralentissement de l'apprentissage observé avec la sigmoïde. Les deux fonctions sont représentées graphiquement dans la figure 1.15.

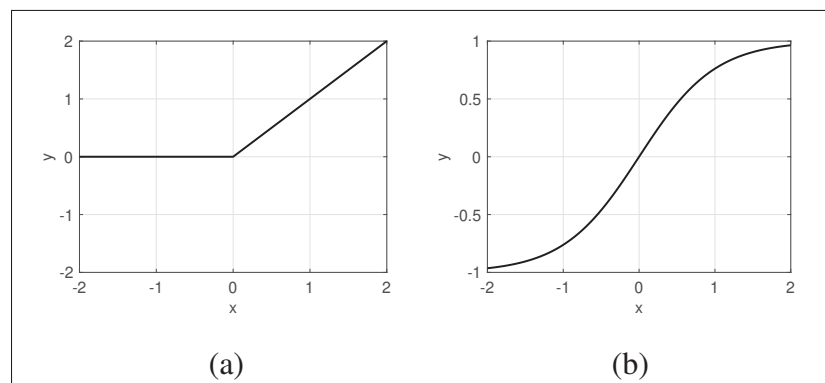


Figure 1.15 Graphique des fonctions d'activation (a) ReLU et (b) Tanh.

#### 1.4.3.2 La normalisation en Batch (Batch Normalization) et la normalisation en couche (Layer Normalization)

Il est bien établi que la normalisation des entrées, de manière à obtenir une moyenne nulle et un écart type constant, contribue à améliorer l'entraînement des réseaux de neurones (Montavon, Orr & Müller, 2012). Afin d'étendre cette pratique aux couches cachées entre l'entrée et la sortie, Ioffe & Szegedy (2015) ont introduit la normalisation en batch (batch normalization). Cette technique permet non seulement d'accélérer l'apprentissage, mais aussi d'améliorer la généralisation du modèle (Bjorck, Gomes, Selman & Weinberger, 2018). En pratique, la couche de normalisation est généralement placée entre la couche de neurones et la fonction d'activation (Thakkar, Tewary & Chakraborty, 2018).

Lors de l'entraînement, les données sont envoyées en lots, appelés batches. La normalisation en batch normalise les données en fonction de l'ensemble des entrées reçues dans le batch, ce qui rend cette méthode dépendante de la taille du batch. En revanche, la normalisation en couche (layer normalization) propose une approche différente en normalisant chaque entrée du batch indépendamment. Cette technique est particulièrement adaptée pour des architectures telles que les Recurrent Neural Networks (RNN) ou l'architecture Transformer, que nous aborderons dans ce mémoire (Ba, Kiros & Hinton, 2016).

#### 1.4.3.3 Le dropout

En général, plus un modèle est grand, plus il est performant pour accomplir la tâche pour laquelle il a été entraîné. Cependant, un modèle plus large est aussi plus susceptible de souffrir du surentraînement. Ce phénomène, appelé overfitting, se produit lorsque le modèle s'adapte trop spécifiquement aux données d'entraînement, ce qui limite sa capacité à généraliser sur de nouvelles données.

Pour contrer cette limitation, la technique du dropout a été introduit par Srivastava, Hinton, Krizhevsky, Sutskever & Salakhutdinov (2014). Le principe est le suivant : lors de l'entraînement, certains neurones sont aléatoirement désactivés avec une probabilité  $p$ , correspondant au taux

de désactivation. Ainsi, certains neurones sont ignorés à chaque itération d'apprentissage, ce qui rend les neurones activés aléatoires à chaque époque. Une fois l'entraînement terminé, tous les neurones sont réactivés pour effectuer les prédictions. La figure 1.16 illustre l'effet du dropout sur un réseau de neurones à propagation avant.

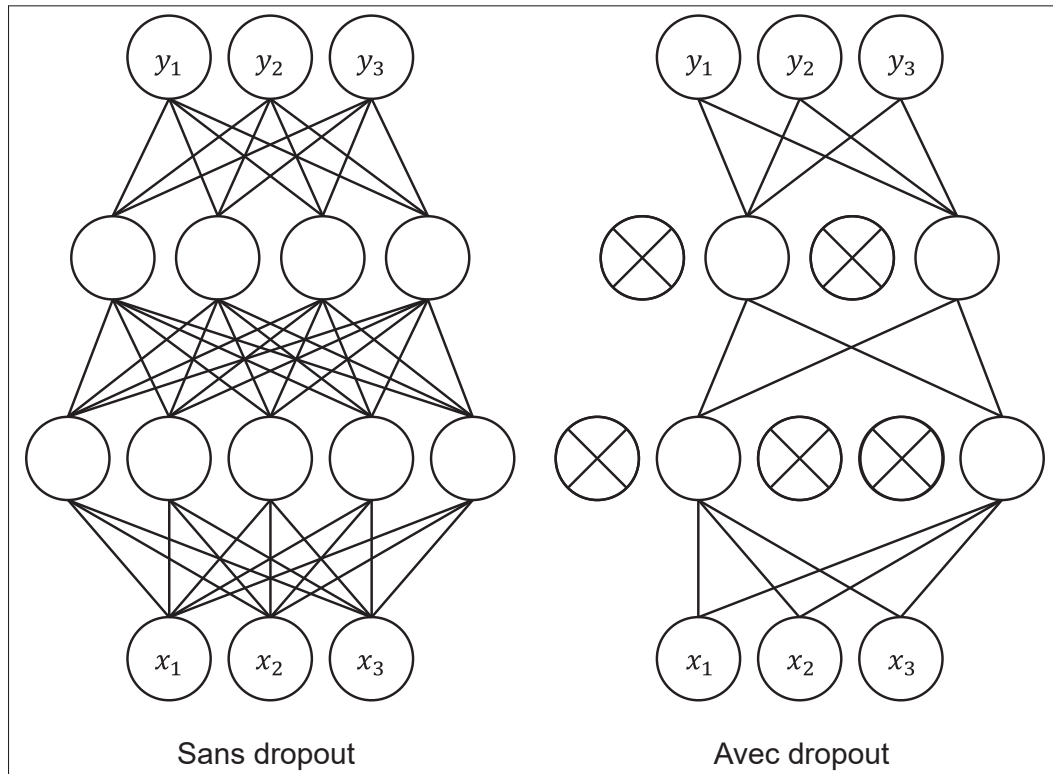


Figure 1.16 Schéma de l'effet du dropout sur des couches de neurones d'un FFN.

#### 1.4.4 Le réseau de neurones Convolutif (CNN)

Le CNN est spécialement conçu pour traiter des données topologiques quadrillées, comme des signaux temporels ou des images. Il permet de résoudre certains des problèmes rencontrés par les réseaux de neurones traditionnels, notamment l'interaction dense entre chaque entrée et chaque sortie pour chaque couche de neurones (Goodfellow *et al.*, 2016).

En essence, un CNN est un réseau de neurones à propagation avant, mais il adopte une topologie différente avec des poids partagés, en utilisant l'opérateur de convolution (équation 1.27).

Un CNN est constitué de couches convolutives, dans lesquelles l'entrée est convoluée avec des filtres appelés kernels. Ces filtres, définis par un ensemble de poids, parcourent l'entrée pour l'analyser. Chaque filtre peut être interprété comme une carte de caractéristiques, permettant de détecter ou d'accentuer une propriété spécifique, comme les contours d'une image (LeCun & Bengio, 1995). L'utilisation de filtres partagés entre les neurones réduit considérablement le nombre de paramètres à apprendre, comparé aux réseaux entièrement connectés.

Par exemple, considérons un A-scan  $\vec{x} \in \mathbb{R}^6$  traversant une couche convolutive avec un filtre  $\vec{h} \in \mathbb{R}^2$ , composé de deux poids  $\alpha$  et  $\beta$ . La sortie  $\vec{y}$  de cette couche aura une dimension  $n = \dim(\vec{x}) - \dim(\vec{h}) + 1 = 5$ . L'opération de convolution réalisée par le filtre sur l'entrée est illustrée dans la figure 1.17.

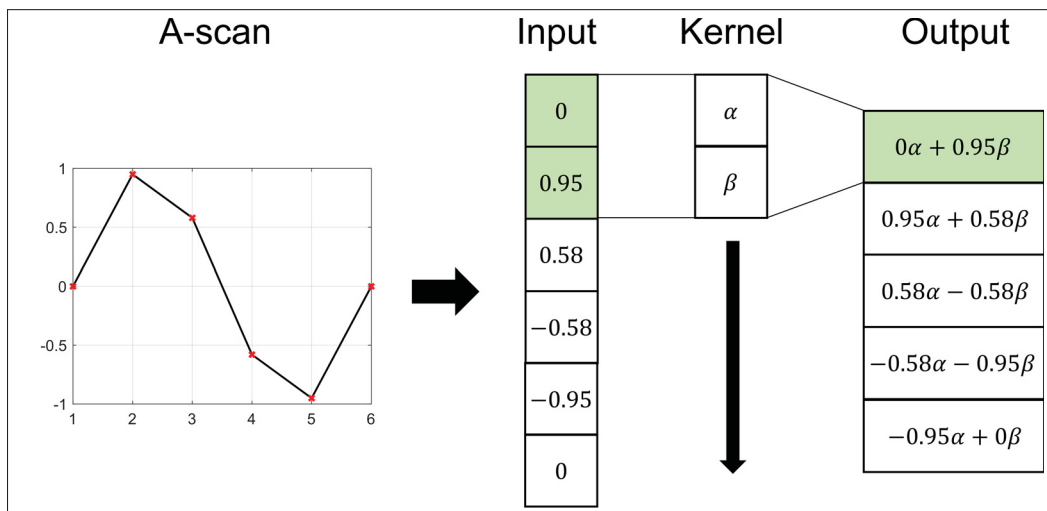


Figure 1.17 Schéma de l'action d'une fenêtre de convolution dans une couche de réseau de neurones convolutif.

La sortie de l'opération de convolution entre les filtres et l'entrée est ensuite transformée par une fonction d'activation, garantissant ainsi la non-linéarité du modèle.



En résumé, une couche convolutive est définie par :

1. Le nombre de filtres ( $n_{filters}$ ) : chaque filtre est utilisé pour extraire une caractéristique spécifique de l'entrée.
2. La dimension des filtres ( $k$ ) : elle correspond à la taille de la fenêtre glissante appliquée sur l'entrée.
3. La fonction d'activation : elle transforme la sortie de l'opération de convolution pour permettre l'apprentissage de relations complexes.

Pour une entrée  $\vec{x} \in \mathbb{R}^n$ , une couche convolutive produit une sortie  $\vec{y} \in \mathbb{R}^{(n-k+1) \times n_{filters}}$ , composée de  $n_{filters}$  vecteurs, chacun de dimension  $n - k + 1$ . Le nombre total de paramètres à optimiser est  $k \cdot n_{filters}$ . La figure 1.18 illustre l'architecture d'un réseau de neurone convolutif simple, uniquement composé de couches convolutives.

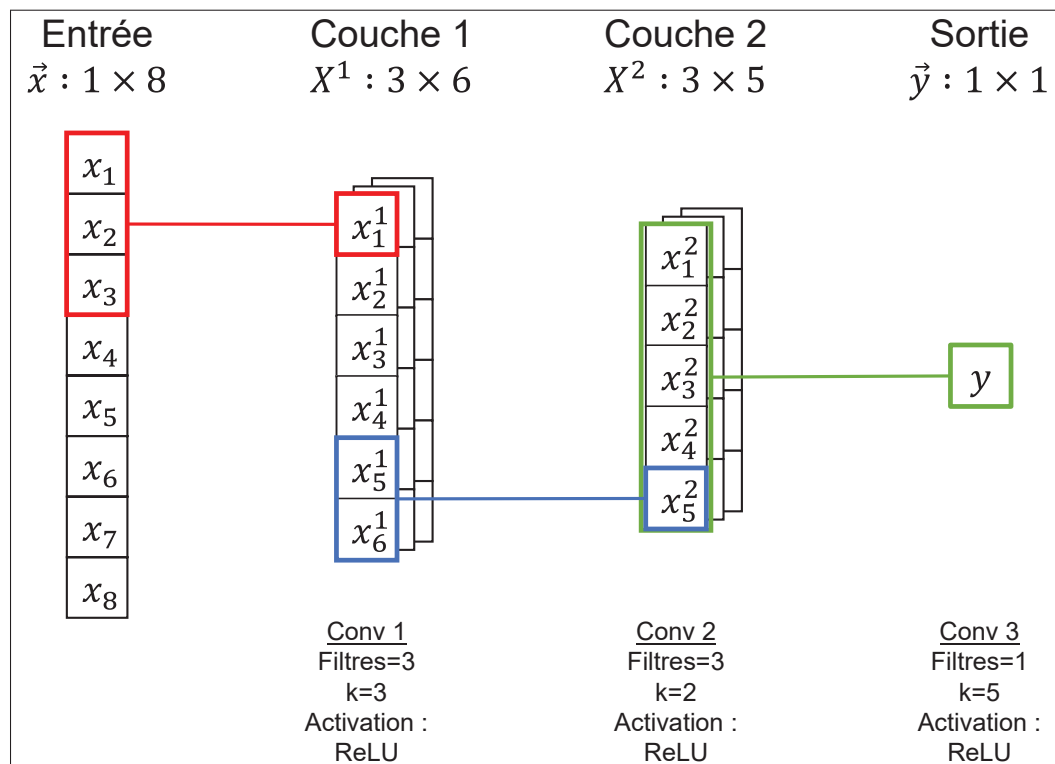


Figure 1.18 Exemple simple d'architecture d'un réseau de neurones convolutif.

Bien que la réduction de dimension soit intéressante dans des applications comme la classification, il est parfois préférable de conserver des vecteurs de même dimension. Cela peut être accompli grâce à l'opération de « padding », qui consiste à ajouter des zéros aux bords pour conserver la même dimension.

La dernière couche est en générale une couche d'aplanissement (flatten), où la matrice à deux ou trois dimensions issue des convolutions est transformée en un vecteur unidimensionnel. Ce vecteur est ensuite traité par une couche de neurones entièrement connectés, comme dans les réseaux de type FFN (Yuan, 2022).

Bien que l'architecture CNN présente des avantages, notamment en mettant en évidence les caractéristiques locales tout en étant plus performante en termes de calcul, elle a aussi ses limites. La taille des filtres impose une restriction sur la portée des caractéristiques observables : l'IA ne peut examiner qu'un voisinage local, d'une dimension finie. Pour capturer des informations globales ou intégrer des paramètres spécifiques répartis sur l'ensemble de l'image, il est souvent nécessaire d'empiler plusieurs couches convolutives. Cependant, cette superposition peut entraîner une augmentation significative de la complexité computationnelle, compromettant ainsi l'avantage initial des CNN (Gulati *et al.*, 2020).

#### **1.4.5 Le réseau de neurones Transformer (TNN)**

Permettre à l'ensemble des données de communiquer entre elles représente un coût computationnel important. En plus du FFN, le RNN a été conçu pour traiter des séquences, comme des chaînes de caractères, où le contexte global de la phrase est crucial pour la génération de texte (Hochreiter & Schmidhuber, 1997). Cependant, bien que des travaux aient considérablement amélioré l'efficacité de calcul de ces réseaux, ils restent limités par les contraintes du calcul séquentiel (Vaswani *et al.*, 2017).

Vaswani *et al.* (2017) ont proposé d'utiliser le mécanisme d'attention pour favoriser les relations à longue distance dans leur architecture Transformer. Le fonctionnement est le suivant.

Soit un modèle de dimension  $d_{model}$ . La fonction d'attention prend en entrée les matrices  $Q, K, V \in \mathbb{R}^{n \times d_{model}}$  nommées « Query », « Key » et « Value » respectivement, pour calculer la matrice « Attention » à l'aide de la fonction d'activation softmax :

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_{model}}}\right)V \quad (1.35)$$

avec T l'opérateur transposé.

Dans le cadre de la génération de texte ou le traitement de chaînes de caractère, les matrices  $Q$ ,  $K$  et  $V$  sont obtenues grâce à un encodeur et un décodeur qui extraient et modélisent les relations entre les séquences. Toutefois, pour des applications comme le traitement du signal, seul le décodeur est nécessaire (Pu *et al.*, 2022). Dans ce cas, les matrices  $Q$ ,  $K$  et  $V$  sont calculées à partir de la sortie de la couche précédente, formant ainsi l'opérateur de « Self-Attention ».

Pour améliorer la capacité des modèles à capturer plusieurs types de relations au sein des données, Vaswani *et al.* (2017) ont introduit le mécanisme de Multi-Head Attention. Les matrices  $Q$ ,  $K$  et  $V$  sont alors projetées linéairement  $h$  fois à l'aide de matrices de poids  $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_{model} \times d_k}$  pour donner  $h$  têtes de dimension  $d_k = \frac{d_{model}}{h}$ .

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \text{ avec } i = 1 \dots h \quad (1.36)$$

Les sorties des différentes têtes sont ensuite concaténées et projetées à nouveau à l'aide d'une matrice de poids  $W^O \in \mathbb{R}^{d_{model} \times d_{model}}$ .

$$MultiHead(Q, K, V) = concat(head_1, \dots, head_h)W^O \quad (1.37)$$

Ce mécanisme permet au modèle de focaliser son attention sur plusieurs aspects ou représentations de l'information simultanément. Par exemple, dans le traitement du signal ultrasonore, certaines

têtes pourraient se concentrer sur les caractéristiques des ondes transversales, tandis que d'autres se focaliseraient sur celles des ondes longitudinales. Cela améliore considérablement la capacité du modèle à capturer des informations globales et locales. La figure 1.19 illustre l'architecture de l'encodeur dans le modèle Transformer tel que décrit par Vaswani *et al.* (2017) et utilisé dans ce mémoire.

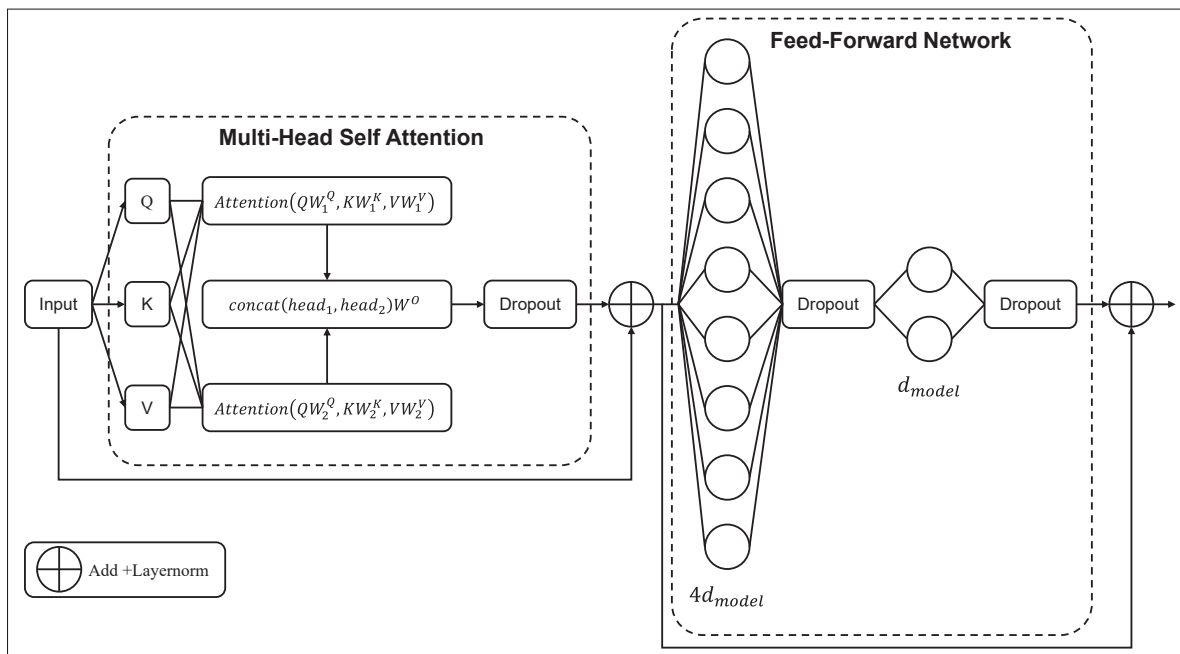


Figure 1.19 Schématisation de l’encodeur de l’architecture Transformer avec 2 têtes et  $d_{model} = 2$ .

### 1.4.6 L'optimisation des modèles

Pour atteindre la fonction souhaitée, l'IA ajuste ses poids en interprétant correctement les données. Une fonction de perte compare la sortie du modèle à la réalité, et les poids sont mis à jour par descente de gradient. Le choix de cette fonction dépend de l'objectif. Pour la déconvolution, où l'objectif est de capturer les grandes variations tout en réduisant l'impact des petites, la fonction de perte des moindres carrés (« Mean Square Error », MSE, ou L2-loss) est privilégiée (Wang, Ma, Zhao & Tian, 2022). Si  $\hat{y}$  est la prédiction du modèle et  $y$  la valeur réelle, la MSE est définie comme :

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1.38)$$

Suite au calcul de la fonction perte, les poids du réseau sont mis à jour grâce à l'algorithme Adam. Il combine les avantages des méthodes de descentes de gradient classiques et adaptatives pour ajuster les taux d'apprentissage, ce qui améliore la convergence et la stabilité de l'apprentissage, en particulier avec des données bruitées (Kingma & Ba, 2017).

Le dernier paramètre crucial à ajuster est le learning rate ( $\eta$ ), qui détermine l'amplitude des ajustements des poids à chaque itération. Un taux trop élevé provoque des oscillations, empêchant la convergence, tandis qu'un taux trop faible ralentit l'apprentissage et risque d'enfermer le modèle dans un minimum local. Le learning rate joue ainsi un rôle clé dans l'équilibre entre la vitesse et la stabilité de l'apprentissage. Pour résoudre ces problèmes, un taux décroissant est souvent utilisé : il débute à une valeur élevée et diminue progressivement au fil des époques (Montavon *et al.*, 2012), (Bjorck *et al.*, 2018).

## 1.5 Conclusion

Dans cette partie, nous avons analysé le parcours des ultrasons, depuis leur émission jusqu'à leur réception, en détaillant les phénomènes de propagation, de réflexion et de dispersion. Pour aborder la déconvolution des A-scans et mesurer l'épaisseur sur des géométries complexes, l'IA s'est imposée comme la solution la plus adaptée. Nous avons également comparé différentes architectures de réseaux de neurones afin de justifier le choix de notre approche pour développer une architecture optimale.



## CHAPITRE 2

### ON THE USE OF A TRANSFORMER NEURAL NETWORK TO DECONVOLVE ULTRASONIC SIGNALS

Thibault Sendra<sup>1</sup> , Pierre Bélanger<sup>1</sup>

<sup>1</sup> Département de Génie Mécanique, École de Technologie Supérieure,  
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article soumis à la revue « Ultrasonics » en janvier 2025.

#### 2.1 Abstract

Pulse-echo ultrasonic techniques play a crucial role in assessing wall thickness deterioration in safety-critical industries. Current approaches face limitations with low signal-to-noise ratios, weak echoes, or vague echo patterns typical of heavily corroded profiles. This study proposes a novel combination of Convolution Neural Networks (CNN) and Transformer Neural Networks (TNN) to improve thickness gauging accuracy for complex geometries and echo patterns. Recognizing the strength of TNN in language processing and speech recognition, the proposed network comprises three modules : 1. pre-processing CNN, 2. a Transformer model and 3. a post-processing CNN. Two datasets, one being simulation-generated, and the other, experimentally gathered from a corroded carbon steel staircase specimen, support the training and testing processes. Results indicate that the proposed model outperforms other AI architectures and traditional methods, providing a 5.45% improvement over CNN architectures from NDE literature, a 1.81% improvement over ResNet-50, and a 17.5% improvement compared to conventional thresholding techniques in accurately detecting depths with a precision of under  $0.5\lambda$ .

**Keywords :** Convolutional Neural Network, Corrosion, Deconvolution, Non-Destructive Evaluation, Time-of-Flight, Thickness Measurement, Transformer Neural Network, Ultrasonic Testing

## 2.2 Introduction

Pulse-echo ultrasonic approaches have been widely adopted for the monitoring of wall thickness deterioration due to aging and corrosion in various sectors, such as the oil and gas (Cegla & Allin, 2015; T. Eason, 2017), offshore (Baltzersen, Waag, Johnsen, Ahlen & Tveit, 2007), and energy industries (Norris, Naus & Graves, 1999). Accurate and reliable wall thickness measurements facilitate informed decisions for the safe operation of critical assets. For thickness evaluation, an emitted ultrasonic pulse is transmitted by an ultrasonic transducer, and propagates through the material. The wave reflects on the backwall or any other sudden acoustic impedance changes along the propagation path. The reflected waves are then captured by the transducer. Often referred to as an A-scan, the recorded signal informs users of pertinent features within the inspected material. Meanwhile, computing the time difference between emission and reception – commonly known as the Time-of-Flight (ToF) – allows to calculate the wall thickness (Krautkrämer *et al.*, 1977) if the speed of sound is known. Therefore, accurately determining the ToF in order to properly assess the safety of pipeline thickness, for instance, is crucial. However, corrosion profiles may produce complex backwall geometries, leading to variable echo amplitudes and patterns resulting from overlapping reflections coming from different depths. Constructive and destructive interferences add a layer of complication by obscuring true distances within unwanted information. These highly complex geometries thus introduce challenges, such as unknown echo patterns or overlapping echoes, as shown in Fig.2.1. Accurately identifying the correct ToF to calculate the minimal distance to the probe remains a critical unsolved problem in nondestructive evaluation (NDE).

Overlapping echoes have been extensively scrutinized in NDE. A first approach used to tackle them consists in increasing the frequency of the transducer (Liu *et al.*, 2018) to increase the resolution, but attenuation follows the same trend, leading to a reduced amplitude for a given sample thickness (Cheeke, 2002; Shull, 2002). Increasing the amplitude is hardly a solution because coherent noise will maintain the overall signal-to-noise ratio (SNR) (Shull, 2002).



Over the years, cross-correlation (Marioli *et al.*, 1992), threshold-crossing (Parrilla *et al.*, 1991) or sparse and blind deconvolution (Chang *et al.*, 2017; He chen, Li bing & Gao fei, 2021) algorithms, among others, have been proposed to improve the resolution of ultrasonic thickness measurements. While those methods are effective in most cases, they become limited with low SNR, attenuated echoes or unknown echo signatures.

Sparse signal representation methods, also known as dictionary-based methods, such as matching-pursuit (Mallat & Zhang, 1993), orthogonal matching-pursuit (Tropp & Gilbert, 2007), and most recently, support matching pursuit (Mor *et al.*, 2010), are other possible approaches for tackling the problem. Matching echo patterns to decompose the signals effectively separates the echoes and allows a global understanding of the A-scan's composition. However, as stated before, irregular corrosion profiles lead to highly variable echo patterns. There is therefore the need for a large dictionary of such patterns, which reduces efficiency and performance.

Advances in computational capabilities – specifically, boosts in GPU and CPU capacities and speeds – have rendered deep learning a feasible industrial solution for complicated tasks such as image classification (Rawat & Wang, 2017), face recognition (Jiang, 2020) and image-to-text generation (He & Deng, 2017). Consequently, deep learning, particularly through convolutional neural network (CNN), has successfully made its way into NDE for defect detection and characterization (Posilović *et al.*, 2019), (Pyle *et al.*, 2021). Indeed, CNN allows to reduce high-dimensional problems and capture local features with its filters using shared weights. Its ability to detect complex patterns have thus been proven through the years in ultrasonic NDE image classification (Uhlig, Alkhasli, Schubert, Tschöpe & Wolff, 2023).

The ability of CNN to perform inverse operations through its primary convolution blocks is at the root of recent advances in ultrasonic NDE, including signal deconvolution and thickness measurement. For instance, Chapon *et al.* (2021) used a simple CNN architecture consisting of two convolutive layers to effectively deconvolve two overlapping echoes, distinguishing flat-bottom holes up to a depth of  $0.5 \lambda$ . However, their database lacks diversity, and only allows the signal signature of a single probe to be recognized, with one frequency and a relatively high

SNR. Furthermore, an individual A-scan can be composed of an unknown number of superposed echoes, whereas their CNN was solely trained to identify two. Shpigler *et al.* (2022) refined the model by adding distinction layers and augmenting the database with highly variable signals, varying SNRs, and an unknown number of overlapping echoes. The architecture was evaluated and compared on two phantoms with matching pursuit algorithms and was proven to have a higher detection accuracy in high overlap conditions, where the layer thickness can reach  $0.25 \lambda$ . Yet, the tests were realized on flat geometries and the architecture was not proven to be effective on corroded profiles, where signal signatures can be modified from echo to echo within the same A-scan. As such, Cantero-Chinchilla *et al.* (2023) proposed an improved database with simulated A-scans on randomly generated corroded profiles. An optimal CNN model was developed to capture the minimal and mean wall thicknesses under the transducer, and was proven to provide a four-fold root mean square error improvement relative to the peak of the envelope technique. Nonetheless, the lack of real data hindered efficient and precise wall thickness determination on experimental data acquired on equally difficult geometries.

Due to the limited scope of kernels, CNNs require the superposition of multiple convolutional layers to capture a global context and physical meaning. Unfortunately, this multiplication hampers their initial computational advantage (Gulati *et al.*, 2020). To tackle this drawback, the Transformer Neural Network (TNN), based on the attention mechanism, was devised, initially for machine translation and language processing (Vaswani *et al.*, 2017). Subsequently, TNNs were broadened to cover signal processing with speech recognition (Gulati *et al.*, 2020; Karita *et al.*, 2019; Zhang *et al.*, 2020), electrocardiogram signal classification (Che, Zhang, Zhu, Qu & Jin, 2021; Guan, Wang, Feng, Wang & Wang, 2021) and noise reduction (Pu *et al.*, 2022; Wang, He & Zhu, 2021). This innovative approach outperforms previous architectures in the mentioned disciplines by offering affordable long-range dimensions and interactions to AI models.

More recent research has explored the combination of CNNs with TNNs, yielding significant performance improvements over traditional Transformer architectures. Wu *et al.* successfully enhanced vision Transformer models by embedding convolutional tokens before the Transformer blocks and incorporating convolutional projections within the Transformer framework (Wu *et al.*,

2021). Similarly, Wang et al. took inspiration from encoder-decoder frameworks to propose a CNN encoder-decoder architecture combined with a Transformer block. This approach achieved superior results for monaural speech enhancement in the time domain by effectively extracting both local and global contextual information (Wang *et al.*, 2021). Gulati *et al.* (2020) further demonstrated the effectiveness of integrating CNNs with TNNs by incorporating a CNN block within the TNN framework and adding a convolutional subsampling layer prior to the modified Transformer architecture. This approach achieved state-of-the-art accuracy in speech recognition tasks. These successes highlight the benefits of CNN-augmented Transformer architectures, where locality guidance and the added distance limitation to the self-attention mechanism effectively improve convergence and performance with smaller datasets (Li *et al.*, 2022).

In this paper, the integration of a TNN for wall thickness measurements was investigated. Inspired by previous research, it was hypothesized that bringing together global and local interactions through a hybrid CNN-TNN architecture will enable accurate and robust wall thickness estimation, regardless of the complexity of the underlying geometry. The proposed architecture was trained with a unique dataset comprising simulations and experimentally measured A-scans on a corroded carbon steel sample. This unique dataset was used to gain valuable insights on the performance of the proposed architecture.

The paper begins with an overview of the materials and methods, and provides details on the generation of the related datasets and architecture. The results are then presented and compared with other architectures and conventional methods available in the literature. Finally, the results are discussed and conclusions are drawn.

## **2.3 Materials and Methods**

### **2.3.1 Problem formulation**

The fundamental nature of the deconvolution predicament is generally outlined as follows. Given the received signal  $y(n)$  as an A-scan, it can be modelled as the mathematical product of  $h(n)$ , the

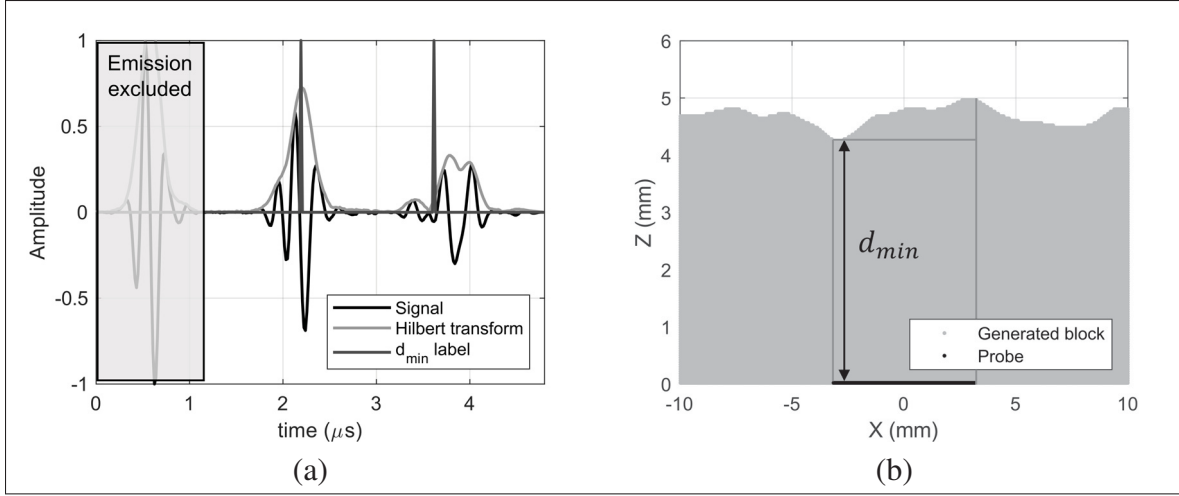


Figure 2.1 Example of the labeling methodology for minimal distance under the probe with : (a) the result coming from probe on the generated block (b).

impulse response, convoluted with  $x(n)$ , representing the ultrasonic waveform, and contaminated by a zero-mean additive band-limited Gaussian noise  $e(n)$  :  $y(n) = h(n) * x(n) + e(n)$ , where  $*$  denotes the linear convolution operation (Nandi *et al.*, 1997).

To grasp the form of the impulse response, one can represent it mathematically by a series of Dirac delta functions  $\delta(n)$  having diverse amplitudes  $h_i$  at distinct moments  $\tau_i$ , hence forming a “sparse spike train” (Kaaresen, 1997).

$$h(n) = \sum_{i=1}^{\infty} (h_i \cdot \delta(n - \tau_i)) \quad (2.1)$$

Thus, locating any pair of consecutive  $\tau_i$  values would allow to measure the ToF accurately. While effective on flat surfaces, it becomes nonviable on rough surfaces where thickness variations follow a continuous function.

In typical NDE inspections, the goal is to establish the minimum thickness under the transducer. To achieve this, the machine learning (ML) architecture will be designed to identify the ultrasonic waveform corresponding to this minimum thickness (Fig.2.1). This simplifies the problem to

locating the  $\tau_i$  pairs associated with the ultrasound reflections from the minimal thickness beneath the probe. The resulting subset of the deconvolved signal will be denoted as  $h'$ .

This paper's approach, therefore, focuses on approximating the function  $f(y) = h'$  by training a CNN-TNN architecture using a learning database of deconvolved A-scans on minimum thicknesses.

### 2.3.2 Database generation

At the core of machine learning lies the quality of the database. Having a qualitative and abundant dataset is paramount to successfully train the different AI models. In this paper, simulated and experimental datasets were generated.

#### 2.3.2.1 Simulation database

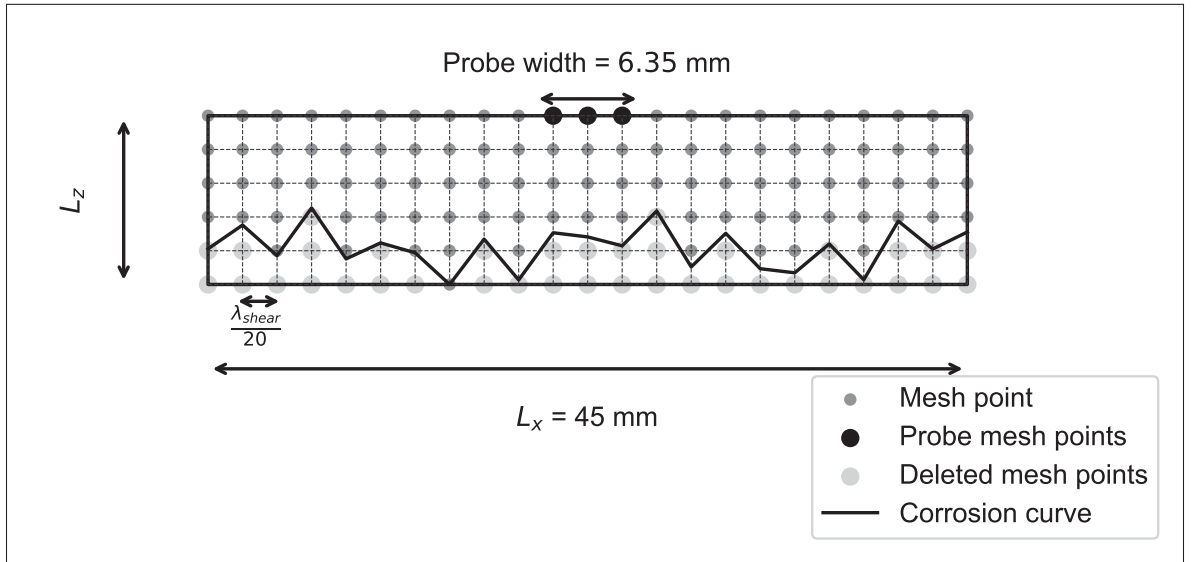


Figure 2.2 Schematic of the block generated for simulation with its mesh.

Finite element simulations using GPU-accelerated Pogo FEA (Huthwaite, 2014) were performed to obtain a large amount of training data. A 2D plane strain corroded steel block was modeled with a depth  $L_z \in [3; 20]$  mm and a width  $L_x = 45$  mm, using 20 linear square elements per the shortest wavelength  $\lambda_{shear}$ , as shown in Fig.2.2. The isotropic and homogeneous steel material

properties were  $E = 205$  GPa,  $\nu = 0.29$  and  $\rho = 7870$  kg/m<sup>3</sup>, implying a longitudinal velocity  $V_l = 5842.5$  m/s. To ensure time marching stability, a time step  $dt = \frac{\lambda_{shear}}{2 \cdot 20 \cdot V_l}$  was used.

For a given signal and block generated, Pogo FEA simulates the emission including the beam spread, the propagation, and the reflections of ultrasonic waves throughout the block. However, attenuation and dispersion were not considered, as the experiments were conducted on an isotropic steel bar of a relatively small thickness. The simulation of a probe was performed by selecting the nodes over a width of 6.35 mm. The same input signal was applied to all nodes, as illustrated in Fig.2.2.

To accurately simulate real-world applications, emitted waveforms must be representative of typical transducer emission patterns. Li, Chen, Zhang, Zhao & Liu (2023) showed that the Nakagami distribution model had the best proximity to echo patterns. While signal generation in the time domain is the standard approach, signal generation in the frequency domain was not discussed. The frequency spectrums of real transducer emissions were acquired on a steel sample, and to match real echo patterns, the spectrums were approximated with Laplace's distributions in the frequency domain. Then, an inverse Fourier Transform was performed to obtain the waveform. The advantages of this method are its flexibility and the ability to easily obtain variables from experimental data, such as  $\mu$ , the central frequency desired during the emission and  $\beta$ , the bandwidth :

$$W(f_n) = \frac{\exp\left(-\frac{|f_n - \mu|}{\beta}\right)}{2\beta} \quad (2.2)$$

Finally, a phase-shifted Hann's window was applied to add an asymmetry to the signal :

$$x(n) = \frac{1}{Len} \cdot \cos^2\left(\frac{\pi(n-d)}{Len}\right) \cdot IFFT(W(f_n)) \quad (2.3)$$

where  $Len = \lfloor \frac{(n_{cycles} \cdot F_{sampling})}{\mu} \rfloor$  is the signal's length – with  $\lfloor . \rfloor$  being the integer part operator – determined by the sample rate  $F_{sampling} = 62.5$  MHz, number of cycles  $n_{cycles}$  and frequency  $\mu$ , and  $d$  the phase-shift.

To conclude, four variables dictate the generated signal's variation and randomness : (1) the central frequency  $\mu \in [2; 2.5] \cup [4.75; 5.25]$  MHz; (2) the bandwidth  $\beta \in 1.7 \cdot [0.63; 1.37]$ ; (3) the number of cycles  $n_{cycles} \in \{3, 5\}$  and (4) the phase-shift  $d \in [\lfloor -\frac{Len}{3} \rfloor; \lfloor \frac{Len}{3} \rfloor]$ . The selected variables were adapted according to the Verasonics Vantage 64 LE and Evident single and multi-element transducer properties, which were used experimentally : V125-RM and 5L64-32X10-A32-P-2.5-OM.

Using this algorithm to approximate the signal emitted by the probes on flat surfaces, an average proximity of 94.86% was achieved using the Normalized Cross-Correlation (NCC) indicator, as described by Li *et al.* (2023).

To simulate the corrosion on Pogo FEA (Huthwaite, 2014), corrosion profiles were randomly generated with an algorithm adapted from Cantero-Chinchilla *et al.* (2023) and subtracted to the backwall of the generated 2D block, as illustrated Fig.2.2. Firstly, the corrosion profile of the corroded step block used experimentally and described in the next subsection was acquired from its laser scan as shown in Fig.2.3a. The properties of that profile, including the amplitudes  $\sigma_i$  at each length frequency  $\lambda_i$ , were calculated and used to generate similar profiles following the algorithm schematized in Fig.2.3c. In this paper, the maximum corrosion amplitude  $\sigma_{tot}$ , as specified Fig.2.3c, was taken within the set  $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$  mm, which is in accordance with the corrosion profiles obtained on the experimental sample (Fig.2.4).

The FE simulated results were filtered with a second-order bandpass between 2 and 10 MHz. The data were then normalized by the maximum amplitude to rescale the values within the interval  $[-1, 1]$  :

$$y'(n) = \frac{y(n)}{\max(|y(n)|)} \quad (2.4)$$

Then, a similarly filtered white Gaussian noise was added, with a SNR varying from 0 to 40 dB.

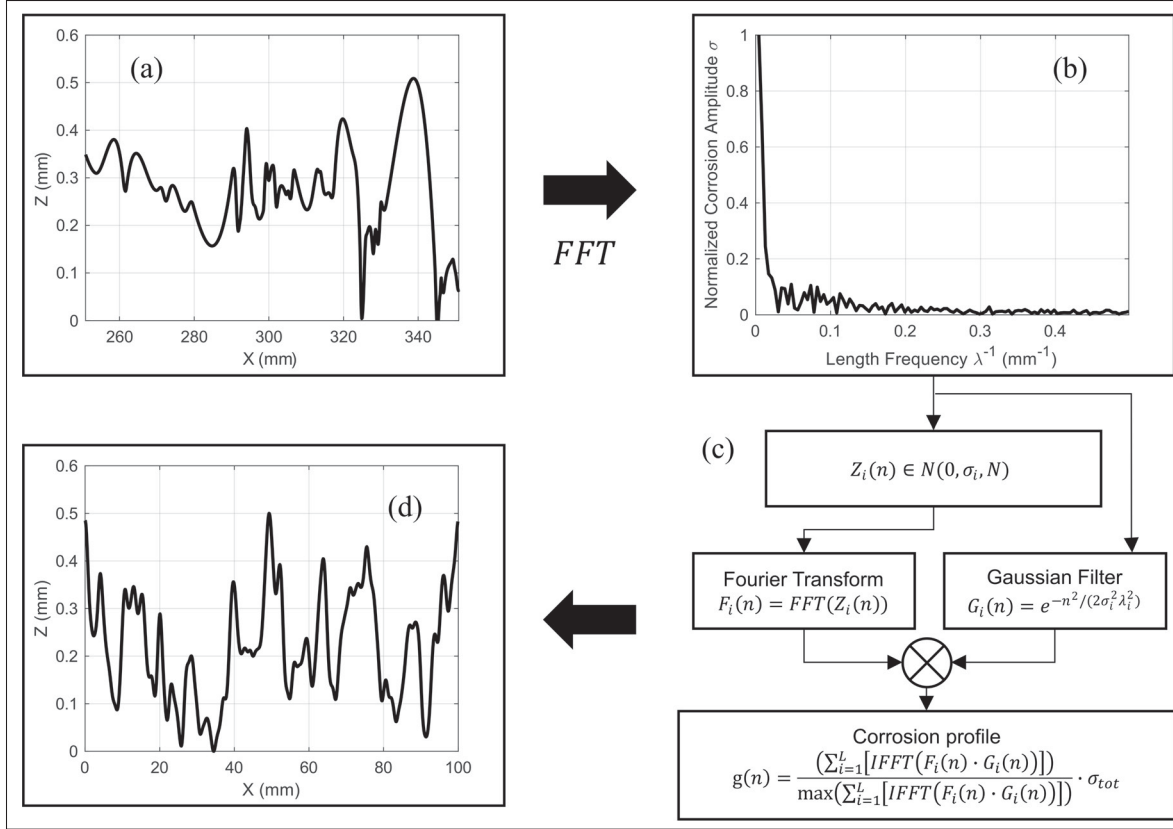


Figure 2.3 Schematic of the corrosion profile generation with : (a) a corrosion profile of the corroded block, (b) the FFT of (a), (c) the algorithm used to randomly generate a corrosion profile with similar properties to the experimental profile, and (d) a typical simulated profile.

Finally, a time window and a tapered cosine window, calculated based on the echo size and the maximum thickness beneath the probe, were applied to the simulated data to isolate the first two echoes (Fig.2.1). This approach excludes the emitted signal, which, in our case, was not experimentally accessible.

Labeling was performed using the minimum distance from the simulated corrosion profile directly beneath the probe (Fig.2.1). Due to experimental constraints, where the emitted signal was inaccessible, the ultrasonic waveform  $x(n)$  used as a reference was taken from the first echo. To ensure a consistent position of the first peak across the datasets used for ML training and to provide a clear visual output, the first peak was identified as the maximum of the Hilbert



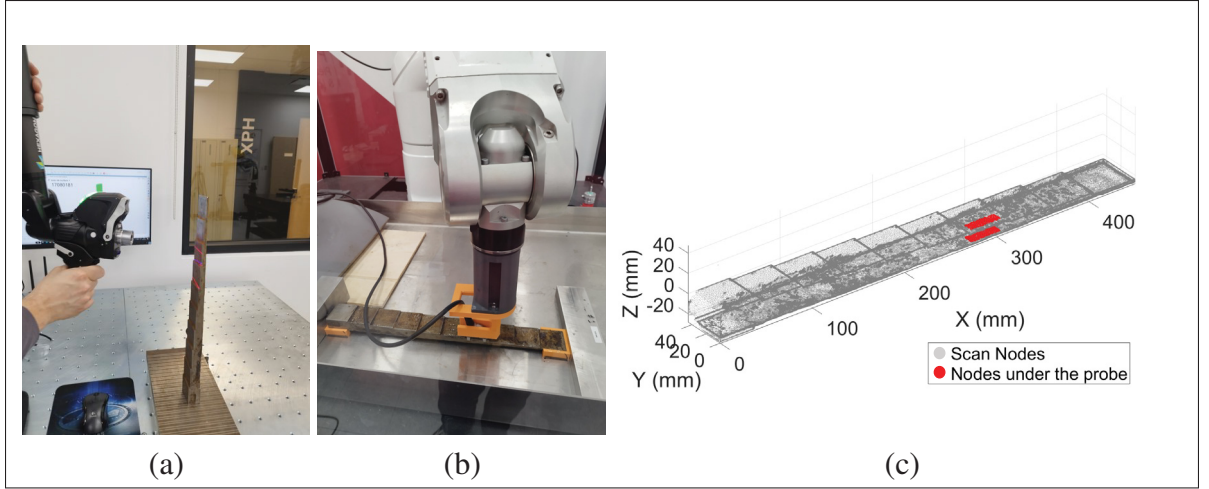


Figure 2.4 Pictures of the experimental setup : (a) laser scan of the corroded staircase specimen, (b) example of an acquisition with the multi-element probe mounted on the robot, and (c) the probe's absolute position on the specimen to determine the minimal distances beneath it.

transform of the first echo. The second peak, corresponding to  $x(n)$  crossing  $2 \cdot d_{min}$ , was then calculated based on the first peak using the longitudinal wave velocity  $V_l$  in the material and the sampling frequency  $F_{sampling}$  (eq.2.6).

From these calculations, the labeled deconvolution signal, representing  $h'$  as described in section 2.3.1, was obtained, with an offset introduced by the position of the first reference point.

$$h'(n) = \begin{cases} 1 & \text{if } n = peak_1 \text{ or } n = peak_2 \\ 0 & \text{else} \end{cases} \quad (2.5)$$

with :

$$\begin{cases} peak_1 = index\{\max[|Hilbert\{y(n)\}|]\} \\ peak_2 = peak_1 + \lfloor \frac{2d_{min} \cdot F_{sampling}}{V_l} \rfloor \end{cases} \quad (2.6)$$

Thus, the proposed TNN and CNN models in this paper will have as input the processed signal  $y(n)$ , and will compare their prediction to the labeled deconvolution  $h'(n)$  for training.

### 2.3.2.2 Experimental database

In order to test the proposed architecture with experimental data and to verify the need to use experimental data in the training, an experimental database was acquired. An AISI 1018 carbon steel stair block was machined, and its longitudinal speed of sound was measured at  $V_l = 5932.1$  m/s. To induce corrosion, the flat face of the block was immersed in saline water to accelerate the corrosion process with electrolysis. A laser scan was subsequently performed, capturing the surface profile with a resolution of 40  $\mu\text{m}$  (Fig.2.4a), providing the corrosion parameters required for the simulations (Fig.2.3a).

A and B-scans were acquired throughout the block using a 2.25 MHz single element longitudinal probe (Olympus V125-RM) and a 5 MHz 64-element longitudinal probe (Olympus 5L64-32X10-A32-P-2.5-OM). Data acquisition was done using a high-frequency Verasonics Vantage 64 LE system with a sampling rate of 62.5 MHz.

A robotic arm was used to position the mono and multi-element probes on various predefined targets on the staircase sample, with a positioning accuracy of approximately 1 mm thanks to RoboDK software (Fig.2.4b). As a result, 162 A-scans and 27 B-scans composed of 64 A-scans each were acquired, uniformly distributed across 9 steps with depths ranging from 5 mm to 25 mm, and a step size increment of 2.5 mm, as illustrated in Fig.2.4.

Owning the absolute position of each A and B-scan, alongside the scan of the staircase sample, minimum distances were extracted for labeling purposes (Fig.2.4c). labeling adhered to the same criteria as the simulation (eqs.2.5 and 2.6). Altogether, a total of 1890 experimental labeled A-scans were acquired (162 from the single element probe and 1728 from the 64-element probe).

### 2.3.2.3 Training datasets

To train the architecture in sections 2.4.1.1 to 2.4.1.3, two databases were used. The first consisted of 2000 simulated A-scans, with depths randomly distributed between 2 mm and 20 mm thanks to the corrosion algorithm (Fig.2.3). This range allows for varying thicknesses to address cases

Tableau 2.1 Tabular summary of the datasets distribution.

Datasets	2.25 MHz A-scans	5 MHz A-scans	Total A-scans
Exp. total	162	0 1728	1890
Exp. train	103	1097	1200
Exp. validation	59	631	690
Sim. total	1377	1623	3000
Sim. train	931	1069	2000
Sim. testing	446	554	1000

where traditional methods struggle to detect the minimal thickness. The second one was made out of 1200 A-scans randomly picked from the pool of 1890 experimental tagged A-scans. The 690 unused experimental A-scans were used as validation while testing was done separately using the methodology outlined in section 2.3.6 with 1000 simulations. The datasets distribution for training, validation and testing are described in Table 2.1.

Finally, the dataset composition is analyzed in section 2.4.1.4 and the optimal configuration is employed in section 2.4.2.

### 2.3.3 Deep Neural Network architecture

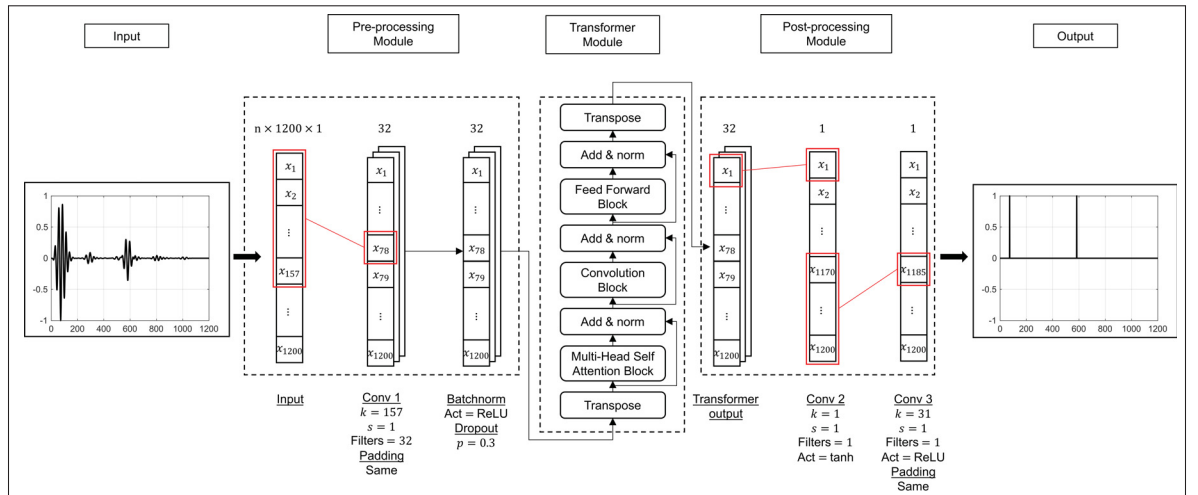


Figure 2.5 Proposed CNN-TNN architecture with reference sizes.

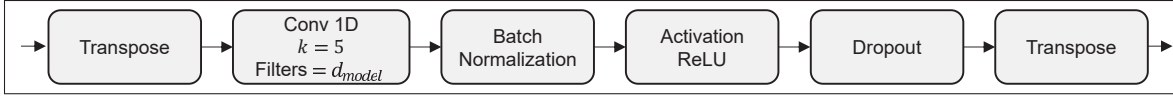


Figure 2.6 Convolution Block details.



Figure 2.7 Feed Forward Block details.

The proposed network architecture  $f^*(y) : \mathbb{R}^T \rightarrow \mathbb{R}^T$ , illustrated in Fig.2.5, consists of three primary components : a pre-processing module, a transformer module, and a post-processing module. This network takes an A-scan  $y(t)$  of length  $T$  as input and produces a deconvoluted signal  $h'(t)$  of equal duration  $T$ . In this part, details of each of these modules are given.

### 2.3.3.1 Pre-processing module

Inspired by the architectures of Chapon *et al.* (2021) and Shpigler *et al.* (2022), the pre-processing module follows a similar approach, initially employing a convolutional layer with 32 filters of kernel size  $k = 157$ . This configuration covers the Region of Interest (ROI) given by the product of the number of cycles  $n_{cycles}$  and the sampling frequency  $F_{sampling}$ , all divided by the signal frequency  $F_{signal}$ . Keeping in mind the variability in signal frequency and the number of cycles, the maximal ROI was set to be 156.25 :

$$ROI = \frac{n_{cycle} \times F_{sampling}}{F_{signal}} \quad (2.7)$$

This kernel size is fixed for sections 2.4.1.2 to 2.4.1.4. However, the significance of the ROI in determining the first kernel size will be examined in section 2.4.1.1.

Multiple filters gather different types of information and interpret various echo signatures, effectively increasing the model's dimensionality. In contrast to the original architectures,

the model was enhanced by introducing batch normalization after the convolutional layer to accelerate training and improve generalization (Ioffe & Szegedy, 2015; Bjorck *et al.*, 2018).

To introduce non-linearity to the model, the batch normalization is followed by a Rectified Linear Unit (ReLU) activation function (Agarap, 2018). In addition, to avoid overfitting, a dropout layer is applied (Srivastava *et al.*, 2014). Through testing, an optimal drop rate of  $p = 0.3$  was found and used for optimization and comparative analysis. Since the drop rate is a subject that has already been covered in other articles (Garbin, Zhu & Marques, 2020), it will not be studied in depth in the present paper.

### 2.3.3.2 Transformer encoding module

The transformer module receives processed local features from the pre-processing module and passes them through the Multi-Head Self-Attention (MHSA) block. Shpigler *et al.* (2022) proposed to use 3 convolutional layers of kernel size 5, stride 1 and dilatation 2 to confront each filter’s data on the proposed detected echoes. While this method may work locally, the limitation of the kernel size prevents it from effectively contrasting the final deconvolution peaks, making it unable to reliably select the two correct ones (Han *et al.*, 2020). Thus, this paper proposes a transformer architecture similar to the one introduced by Vaswani *et al.* (2017). This setup encourages relating long-range dependencies without resorting to Recurrent Neural Networks (RNN) or serial convolutional layers, conserving computational effort and enhancing global understanding. Since a convolutional layer containing the positional information precedes the transformer, position encoding is not required (Yuan, Li & Wu, 2022; Wu *et al.*, 2021).

Inspired by the architectures of Pu *et al.* (2022), Gulati *et al.* (2020) and Vaswani *et al.* (2017), the input goes through the MHSA block, comprised of multiple self-attention units operating together. Each self-attention block projects queries  $Q_i$ , keys  $K_i$  and values  $V_i$  using learned matrix weights  $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{1200 \times d_k}$ . In self-attention, the matrices Q, K and V correspond

to the output of the previous layer :

$$\begin{aligned} Head_i &= SelfAttention(Q_i, K_i, V_i) \\ &= softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \end{aligned} \quad (2.8)$$

where  $Q_i, K_i, V_i \in \mathbb{R}^{d_{model} \times d_k}$  and  $d_k = \frac{d_{model}}{n_{head}}$  using  $d_{model} = 32$  from the reference Fig.2.5.

$Head_i$  represents the self-attention computed outputs, which are ultimately concatenated and passed through another projection matrix  $W^O \in \mathbb{R}^{d_{model} \times 1200}$  to yield the final MHSA output :

$$MHSA(Q, K, V) = concat(Head_1, \dots, Head_{n_{head}}) W^O \quad (2.9)$$

Using multiple heads allows to focus on details across various subspace and positions. To capture distinct information within the model, using multiple heads may be useful (Vaswani *et al.*, 2017). For instance, having two heads could help in distinguishing longitudinal waves from transversal ones within an A-scan. Since deconvolution simplicity does not warrant multiple heads, and only longitudinal waves are addressed, a single head  $n_{head} = 1$  is deployed to minimize the complexity of the architecture. Finally, a dropout layer with a drop rate of  $p = 0.3$  accompanies the MHSA. The same drop rate is used over the transformer module.

Inspired by Gulati *et al.* (2020), a convolution-augmented transformer consolidates global information locally, connecting the global and local information processing stages. Wrapping up the transformer module, a Feed Forward Network (FFN) lets parameters within each A-scan interact both locally and globally. The Convolution Block and Feed Forward Block are detailed in Figs.2.6 and 2.7, respectively.

### 2.3.3.3 Post-processing module

To conclude, the final module consists of two convolutional layers aimed at deconvolving the signal, condensing all earlier information into a vector. The first layer ends with a hyperbolic tangent (tanh) activation used to capture both positive and negative data, avoiding sigmoids and thereby hastening convergence (Lau & Hann Lim, 2018). Since only positive information is to be seized, the last layer captures the final data to convolve it into a positive vector with a ReLU activation.

Unlike the architectures of Chapon *et al.* (2021) and Shpigler *et al.* (2022), which conclude with a convolutional layer of kernel size  $k = 1$  and a ReLU activation, this model introduces an innovative approach. The output is first encoded using the tanh activation, transforming all values into the interval  $[-1, 1]$ . Subsequently, a final convolutional layer with a kernel size  $k = 31$  and ReLU activation refines the information. This larger kernel size enhances the model's capacity to consolidate features over a broader context and accurately construct the final deconvolution peaks.

Finally, the  $L_2 - loss$  function compares the output with the expected result, emphasizing errors between estimated and real deconvolution, quickening convergence and lifting the model accuracy (Wang *et al.*, 2022). To prevent the occurrence of vanishing gradients and dead neurons linked to the terminal ReLU activation, labeled data got amplified by 1000, yielding two deconvolution peaks of magnitude 1000.

## 2.3.4 Training

### 2.3.4.1 Training and batching

Using TensorFlow and Keras (Abadi *et al.*, 2016), the architectures were generated and trained for 80 epochs with a training batch size of 32 and a validation batch size of 690, utilizing the datasets from Table 2.1. Checkpoint callbacks were employed to save each trained model at the point of its highest validation loss, based on the metric defined in Eqs. 2.13 and 2.14, throughout

the 80 epochs. This approach ensures that, for each trained architecture, the best-performing model is selected for comparison.

#### 2.3.4.2 Optimizer

The models were trained using the Adam optimizer (Kingma & Ba, 2017), with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$  and  $\epsilon = 10^{-7}$ . The learning rate varied from epoch to epoch according to the formula :

$$\begin{cases} l_{start} + l_{step} \cdot epoch & \text{if } epoch < warmup \\ \frac{l_{start} + l_{step} \cdot warmup}{1 + 0.1 \cdot (epoch - warmup)} & \text{if } epoch \geq warmup \end{cases} \quad (2.10)$$

The learning rate rose linearly to hit a ceiling before descending inversely proportional to the epoch number. For this article, the values used were  $l_{start} = 0.0001$ ,  $l_{step} = 0.0001$  and  $warmup = 10$ .

#### 2.3.5 Comparison to other architectures and methods

The proposed architecture was compared with those outlined in NDE-focused publications by Chapon *et al.* (2021), Shpigler *et al.* (2022) and Cantero-Chinchilla *et al.* (2023), ensuring a fair comparison by employing identical databases and conditions. Each architecture was evaluated using the parameters and optimized setups outlined in their respective studies.

To further evaluate the capabilities of the proposed TNN architecture against traditional CNN models, popular architectures such as ResNet-50 (He, Zhang, Ren & Sun, 2016) and EfficientNet-B0 and B4 (Tan & Le, 2019) were also included in the comparison. Input signals were resized to 224x224 images using bilinear interpolation, then converted from grayscale to RGB to match the standard input dimensions of 224x224x3. The models were implemented as described in their respective papers, with 1200 output classes and a linear activation function to perform the deconvolution, effectively identifying two classes among the 1200 possible ones. Training was



conducted using the L2-loss function over 100 epochs with an initial learning rate of 0.001, which was halved if no improvement in loss was observed after 10 epochs.

Additionally, the industry-standard threshold algorithm was used for comparisons with AI techniques, as done by Cantero-Chinchilla *et al.* (2023). Using the Hilbert transform as the envelop of the signal, the two rising points crossing the threshold were taken to calculate the ToF (Barshan, 2000). Adaptive thresholds were used according to the signal's frequency for better results :

1. 19% for 2.25 MHz and 38% for 5 MHz for experimental testing
2. 58% for 2.25 MHz and 87% for 5 MHz for simulation testing

### 2.3.6 Evaluation methodology

The trained CNN and TNN models, and the threshold algorithm were tested on both experimental and simulation datasets to ensure their adaptability and performance in broad case scenarios. The datasets were separated into 690 unused labeled A-scans from the experimental dataset and 1000 unseen simulated A-scans.

The distance was calculated with the time-of-flight (ToF) metric, provided by the top two AI-derived deconvolution peaks separated by a minimum of 3 mm,  $peak_1$  and  $peak_2$ . Knowing the longitudinal speed  $V_l$  and sampling frequency of the machine  $F_{sampling}$ , we got :

$$d_{predicted} = \frac{|peak_1 - peak_2|}{2 \cdot F_{sampling}} \cdot V_l \quad (2.11)$$

Given the true distance  $d_{exp}$  for each A-scan,  $d_{predicted}$  was compared to  $d_{exp}$ . A success was recorded if the difference was lower than a predetermined precision  $p$ . Success percentages were acquired based on the cumulative success count across the experimental or simulated datasets.

$$|d_{predicted} - d_{exp}| \leq p \quad (2.12)$$

The success criterion allows to calculate the success percentage (SP) of each trained model ( $AI_i$ ) as follows :

$$SP(AI_i) = \frac{\sum_{k=1}^N \begin{cases} 1 & \text{if } |d_{predicted_k} - d_{exp_k}| \leq 0.5\lambda \\ 0 & \text{else} \end{cases}}{N} \quad (2.13)$$

To comprehensively evaluate every solution, a precision criterion of  $p = 0.5\lambda$  was imposed (1.32 mm for 2.25 MHz and 0.59 mm for 5 MHz), as per the minimal axial resolution described by Shastri *et al.* (2020), serving as a benchmark for assessing the performance in both simulations and real-world scenarios. For each configuration or architecture tested, five AI models were trained under the same conditions. The success rate for each configuration was then computed as the average of the success percentages of the five trained models :

$$\text{Success Rate} = \frac{\sum_{i=1}^5 (SP(AI_i))}{5} \quad (2.14)$$

To provide a more robust representation of the results, error bars were added to each success rate, representing the Margin of Error of the success percentages across the five evaluated models, with a confidence level of 95% :

$$\text{Margin of Error} = 1.96 \cdot \frac{\sigma(\{SP(AI_i)\}_{i \in [1,5]})}{\sqrt{5}} \quad (2.15)$$

## 2.4 Results

### 2.4.1 Transformer study

An efficient utilization and maximum exploitation of the Transformer architecture require a thorough investigation of its key parameters such as the kernel size, the overall size, and layer implementations, on unseen simulation and real-world datasets. This strategic approach seeks

to underline the correlations between the model's inner workings and external performances. Parallel to this study, a comprehensive survey of the database size and composition will be presented. This critical appraisal provides essential insights into the data requirements for training a proficient deconvolution CNN-TNN model, ensuring an efficient training with available resources.

#### 2.4.1.1 Convolution kernel size

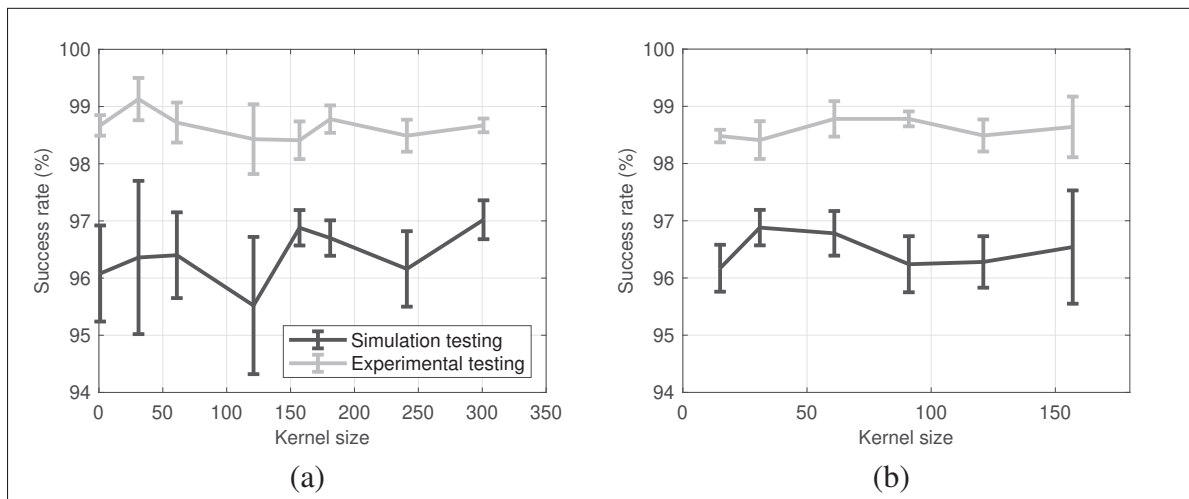


Figure 2.8 Success rate according to the kernel size of : (a) the conv 1 and (b) the conv 3 layer on the simulation and the experimental test datasets.

To investigate the significance of the pre-processing module in capturing the local context through echo signatures, the kernel size of the conv 1 layer from Fig.2.5 was studied, as shown in Fig.2.8a. Its influence on the overall structure is examined by varying it from 1 to 301.

Similarly, the conv 3 layer in the post-processing module being the core of the deconvolution process, its understanding is necessary for a complete survey of the architecture. By keeping the architecture shown in Fig.2.5, a kernel size scan from 15 to 157 was performed, with results presented in Fig.2.8b.

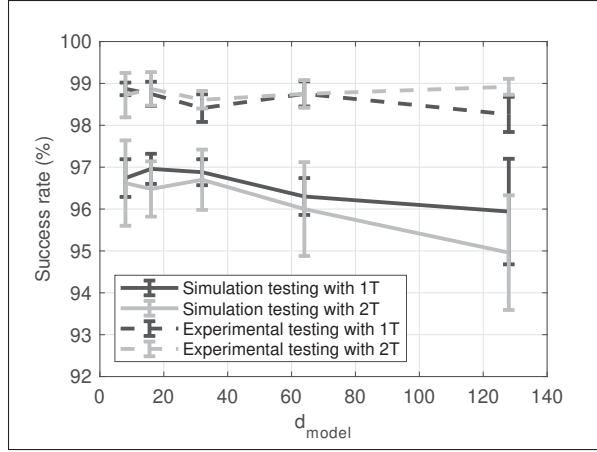


Figure 2.9 Success rate with respect to the dimension of the model  $d_{model}$  with one or two transformers in series on the simulation and the experimental test datasets.

#### 2.4.1.2 Architecture size

For computational efficiency, an optimized model size is desirable. To fully understand the connection between the architecture scale and task performance, the impact of the parameter count on the overall AI results was studied and illustrated in Fig.2.9. Accordingly, the number of filters in the pre-processing module, corresponding to  $d_{model}$ , was adjusted to values ranging from 8 to 128. This trial was run twice : once with a solitary Transformer, and again with two Transformers in series.

#### 2.4.1.3 Ablation studies

To study the internal dynamics of the proposed architecture and isolate the effects of each module — namely the Transformer, Pre-processing, and Post-processing modules — selected components were strategically disabled, and the results are presented in Fig.2.10. This approach allows for a focused evaluation of each module's contribution, helping to gauge the significance and impact of each constituent element on the overall performance of the architecture.

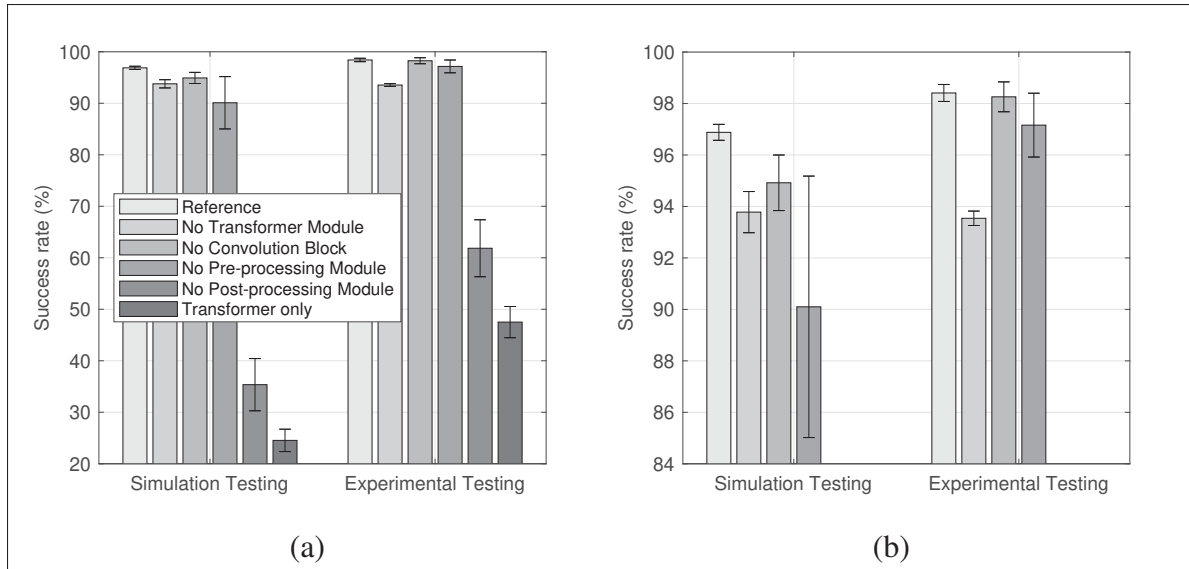


Figure 2.10 Success rate depending on the removed blocks or modules on the simulation and the experimental test datasets with : (a) unzoned and (b) zoomed view.

The entire Transformer module was first disabled to highlight its pivotal role within the overall structure. Next, the Transformer module was modified by omitting the local interaction achieved through the convolution block, granting insights on its effects. The impact of the echo detection and projection modules were then independently evaluated by analyzing the individual performance of the Pre-processing and Post-processing modules in conjunction with the Transformer module. Finally, the Transformer was assessed in isolation, without the Pre/Post-processing modules or the convolution block, to evaluate its standalone performance.

#### 2.4.1.4 Database composition

Since running simulations is more convenient and budget-friendly than acquiring experimental data, training an AI model exclusively using simulation data with satisfactory real-world performance is desirable. Thus, the impact of the composition of the training dataset was investigated, as shown in Fig. 2.11. Additionally, a TNN model trained solely on experimental data was tested to : (1) evaluate the realism of the simulations and (2) establish a benchmark comparison for the simulation-supplemented database.

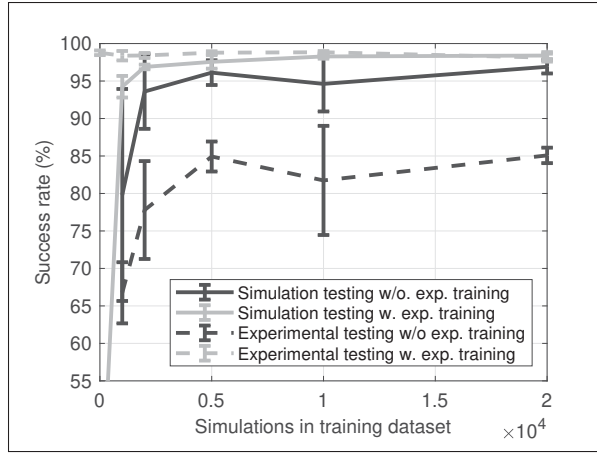


Figure 2.11 Success rate following the quantity of simulations within the training dataset on the simulation and the experimental test datasets.

#### 2.4.2 Comparative results

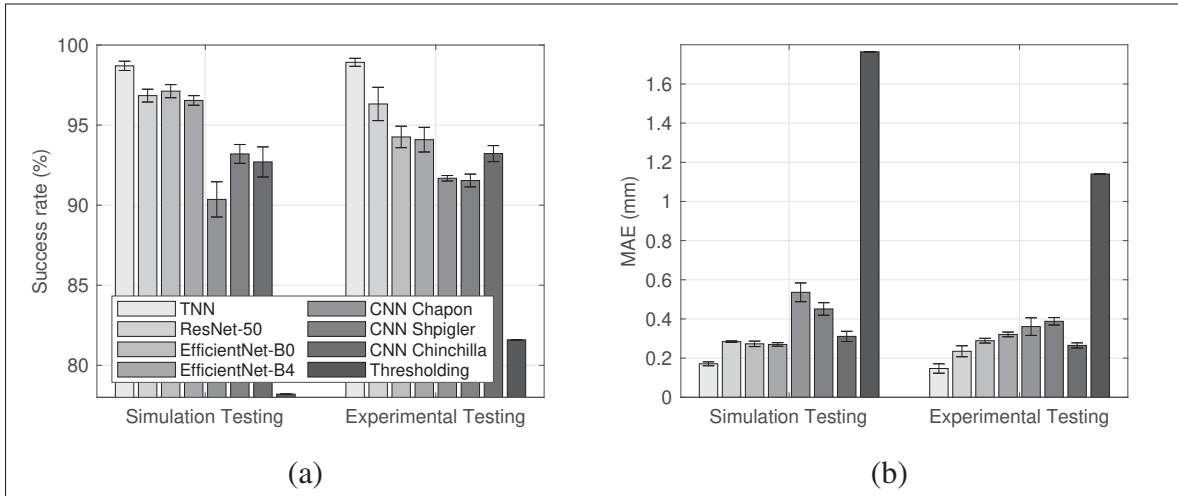


Figure 2.12 Comparative bar chart of (a) the success rates and (b) the MAE of different AI architectures and industrial algorithm on the simulation and the experimental test datasets.

The fine-tuned Transformer architecture was compared with those presented by Chapon *et al.* (2021), Shpigler *et al.* (2022), Cantero-Chinchilla *et al.* (2023), He *et al.* (2016), Tan & Le (2019) and with the thresholding method, with the results summarized in Fig.2.12 and Table 2.2. The refined TNN employs 16 filters with a conv 1 layer kernel size of 31 and a conv 3

layer kernel size of 31. All dropout rates were standardized to 0.3, while the single Transformer remained untouched. Lastly, to ensure a balanced training, 10,000 simulations were augmented with the experimental dataset comprising 1,200 measurements to compose the training dataset.

To further analyze and compare models with similar success rates, another perspective using the MAE is presented in Fig.2.12b. Furthermore, using keras-flops to measure the number of parameters and calculations (FLOPS) performed by each architecture, additional visualizations of the models' performances are shown in Fig.2.13.

Tableau 2.2 Number of parameters, number of FLOPS, Success Rate (SR) and MAE between actual and predicted distances shown in Fig.2.12 of the different methods on the simulation and the experimental test datasets.

Architecture	#Parameters	#FLOPS	SR exp (%)	SR sim (%)	MAE exp (mm)	MAE sim (mm)
TNN (this article)	11,613,233	373.8M	$98.92 \pm 0.25$	$98.70 \pm 0.29$	$0.147 \pm 0.024$	$0.171 \pm 0.010$
ResNet-50	26,046,512	7756M	$96.32 \pm 1.04$	$96.84 \pm 0.40$	$0.235 \pm 0.028$	$0.284 \pm 0.004$
EfficientNet-B0	5,586,771	804M	$94.26 \pm 0.67$	$97.12 \pm 0.41$	$0.289 \pm 0.012$	$0.273 \pm 0.014$
EfficientNet-B4	19,825,423	3089M	$94.09 \pm 0.77$	$96.54 \pm 0.30$	$0.321 \pm 0.012$	$0.270 \pm 0.009$
CNN Chapon	30,913	74.1M	$91.68 \pm 0.17$	$90.36 \pm 1.10$	$0.361 \pm 0.045$	$0.536 \pm 0.048$
CNN Shpigler	15,451	36.9M	$91.54 \pm 0.40$	$93.20 \pm 0.59$	$0.388 \pm 0.019$	$0.451 \pm 0.032$
CNN Cantero-Chinchilla	650,497	1203M	$93.22 \pm 0.50$	$92.70 \pm 0.94$	$0.265 \pm 0.013$	$0.311 \pm 0.026$
Threshold	//	//	81.58	78.20	1.14	1.76

MAE = Mean Absolute Error, SR = Success Rate

## 2.5 Discussion

### 2.5.1 Convolution kernel size

The initial assumption regarding the role of the Conv 1 layer in capturing potential echoes is challenged by the results presented in Fig.2.8a. Surprisingly, using a kernel size of 1 does not negatively impact the architecture, yielding satisfactory results. Moreover, the best performance is achieved when the kernel size of the Conv 1 layer matches that of the Conv 3 layer. While the Conv 1 layer is involved in echo detection, as demonstrated in Shpigler's work (Shpigler *et al.*, 2022) and in Fig.2.8a (with the average line), its primary function is to encode the A-scans for better interpretation by the Transformer (Wu *et al.*, 2021). The Conv 3 layer, responsible for decoding the encoded signal to perform deconvolution, is more effective when its kernel

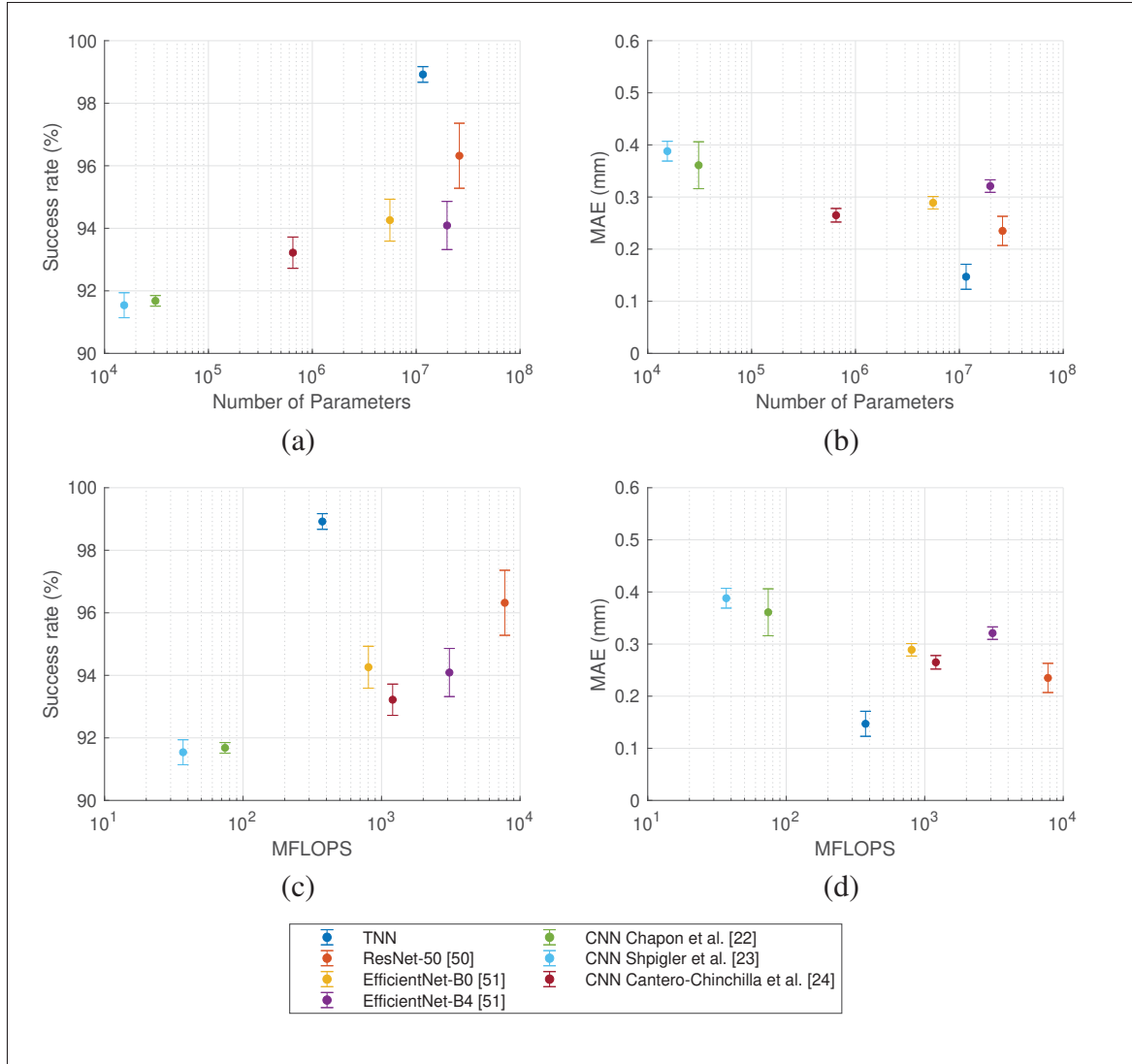


Figure 2.13 Comparative analysis of various AI architectures on the experimental test dataset : (a) success rates versus the number of parameters, (b) MAE versus the number of parameters, (c) success rates versus the number of FLOPS, and (d) MAE versus the number of FLOPS.

size matches that of the encoding layer, as illustrated in Fig.2.8a with  $k = 31$  and Fig.2.8b with  $k = 157$ . Integrating a TNN between a CNN encoder and decoder with matching kernel sizes has been explored previously for speech enhancement (Wang *et al.*, 2021). In this study, the best feature extraction was achieved with a kernel size of 31. Further investigation would be required



to fine-tune the ideal kernel sizes for both convolution layers, but this lies beyond the scope of the present work.

### 2.5.2 Size

Increasing  $d_{model}$  causes the model to grow exponentially. However, this does not lead to a proportional improvement in performance. As seen in Fig.2.9, the proposed architecture's success rate reaches an upper bound in both simulation and experimental testing, and even declines when  $d_{model} \geq 64$  during simulation testing. This trend may be attributed to the growing complexity of the Transformer Module as  $d_{model}$  increases, combined with the low-size training dataset, as demonstrated by Dosovitskiy *et al.* (2021). Indeed, as the model size grows, so does the demand for a larger training dataset to fully exploit the model's capacity. Without sufficient data, the model struggles to generalize, leading to diminished performance despite its increased complexity. Therefore, increasing the size of the architecture beyond a certain point offers diminishing returns, resulting in a decelerated processing speed and increased resource consumption without noticeable improvements. Adapting the architecture's dimensions to match the task's intricacy and training dataset size thus ensures a balance between the success rate and computational efficiency.

### 2.5.3 Ablation

Considering the best obtainable success rate with a 95% confidence level, removing the Transformer Module results in a 4.92% reduction in success rate on experimental data and a 2.61% reduction on simulation data, as shown in Fig.2.10. This highlights the importance of the global context provided by the Transformer in improving the overall understanding of the A-scan for its final deconvolution. Interestingly, removing the Convolution Block within the Transformer Module slightly increases the success rate by 0.10% on experimental data but decreases it by 1.19% on simulation data. However, the local interactions provided by the convolution block contribute to increased model stability, reducing the margin of error between each trained model.

This can be attributed to the better generalization capabilities of CNNs when trained with smaller dataset sizes (Dosovitskiy *et al.*, 2021) (Li *et al.*, 2022).

Similarly, while removing the pre-processing module has little effect on experimental data, it leads to a significant drop in success rate on simulation data and is associated with high variability across the five trained models ( $\pm 5.08\%$ ). These results suggest that the pre-processing module plays a key role in position encoding (Wu *et al.*, 2021) and facilitates learning, particularly with small datasets, enhancing the model's comprehension, learnability, and stability. However, its primary role in echo detection is questioned, given the relatively high success rates achieved without it.

In contrast, removing the post-processing module drastically lowers the success rate, dropping it below 70% on experimental data and 30% on simulation data. The Transformer Module alone does not provide adequate signal processing capabilities, leading to poor signal regression and deconvolution performances. This is further emphasized by the low success rates observed when the Transformer Module is used without other components (50.54% on experimental data and 26.71% on simulation data). This decline can be attributed to the absence of key elements such as adequate number of training samples, position encoding, and the ability to capture local features, as described by Wu *et al.* (2021).

Thus, while the Transformer is less effective for regression tasks, it excels at capturing the global context of the A-scans. This contextual information supplements the CNN modules, providing key insights that are essential for the model's overall performance, especially in tasks like signal deconvolution and echo detection. By combining the strengths of both the Transformer and CNN, the architecture achieves improved precision and stability.

#### **2.5.4 Database composition**

The Transformer's poor generalization capabilities in the context of small datasets have been well-documented in the AI community (Wu *et al.*, 2021; Dosovitskiy *et al.*, 2021; Li & Zhang, 2024), and this limitation is evident in Fig.2.11 for dataset sizes under 10k. While Vision

Transformers (ViT) and other Transformer architectures require datasets in the millions to outperform traditional CNN architectures, such as ResNet (He *et al.*, 2016) or EfficientNet (Tan & Le, 2019), in tasks like image classification, the Transformer architecture proposed in this paper achieves high performance with much smaller datasets as shown in Fig.2.11. Specifically, the performance of the CNN-TNN model converges after 10k training samples, with minimal improvement when the dataset size is doubled. This phenomenon can be attributed to two key factors.

Firstly, the task addressed in this study is simpler compared to those tackled by larger models. Here, the model deconvolve a 1D signal of size 1200x1, whereas image classification tasks typically involve processing input of size 224x224x3 across 1k, 18k or 21k classes (Dosovitskiy *et al.*, 2021). Supporting this observation, Pu *et al.* successfully trained a Transformer model to denoise EEG signals of size 512x1 with a training dataset size of 13,512 (approximately 45k after augmentation), surpassing CNN performance in their application (Pu *et al.*, 2022). While their model predicted 512 values, the model in this study needs to identify 2 values within the 1,200 possible classes, significantly reducing the task complexity and the required dataset size.

Secondly, the proposed architecture incorporates convolutional layers for encoding and decoding. These layers effectively introduce local information, facilitating learning during backpropagation (Gulati *et al.*, 2020; Li *et al.*, 2022). This hybrid design leverages the strengths of both CNNs and Transformers, enabling efficient training even on smaller datasets. Additionally, the design could be further enhanced by incorporating the convolutional block in parallel with the MHSA layer, as demonstrated by Wu, Liu, Lin, Lin & Han (2020).

With limited access to experimental data but abundant simulation samples, finding the optimal balance between these resources is critical for effective learning. At a 95% confidence level, when no experimental data are included, increasing the size of the simulation-based training dataset from 1,000 to 10,000 data improves the success rate on both experimental and simulation testing datasets by 18.19% and 4.36% respectively. However, the Transformer struggles to generalize to experimental data when trained exclusively on simulation samples. Adding 1,200

experimental samples to the 10,000 simulation samples boosts the success rate on experimental data from 89.02% to 99%. This result underscores the model’s limited generalization ability and highlights the necessity of including experimental data in the training process.

Incorporating experimental data also enhances the success rate on the simulation testing set and improves the model’s learning stability. This improvement is likely due to the increased overall dataset size, as adding experimental data augments the total number of training samples.

However, training the model exclusively on experimental data leads to overfitting, as demonstrated by the success rate of under 55% on the simulation testing when no simulation data is included (Fig.2.11). Therefore, adding simulation data with high variability helps the model generalize and prevents overfitting.

For this application, the optimal ratio of training data was found to be approximately 90% simulation and 10% experimental data (10,000 simulations for 1,200 experimental samples), offering the most effective balance between training efficiency and generalizability across different datasets. This ratio ensures that the model can leverage the abundance of simulation data while still incorporating enough experimental data to generalize well in real-world scenarios.

### 2.5.5 Comparative results

At a 95% confidence level, following the training of each architecture as described in Section 2.5, the top-performing TNN demonstrated a 5.20% higher success rate compared to the best CNN result in simulation testing (CNN Shpigler) and a 5.45% improvement in experimental testing (CNN Cantero-Chinchilla). These CNN architectures, described in NDE-focused studies (Chapon *et al.*, 2021; Shpigler *et al.*, 2022; Cantero-Chinchilla *et al.*, 2023), were specifically designed for tasks in this domain. For deeper and more general models like ResNet and EfficientNet, the TNN exhibited a 1.46% improvement in simulation testing and a 1.81% improvement in experimental testing, as shown in Fig.2.12a and summarized in Table 2.2.

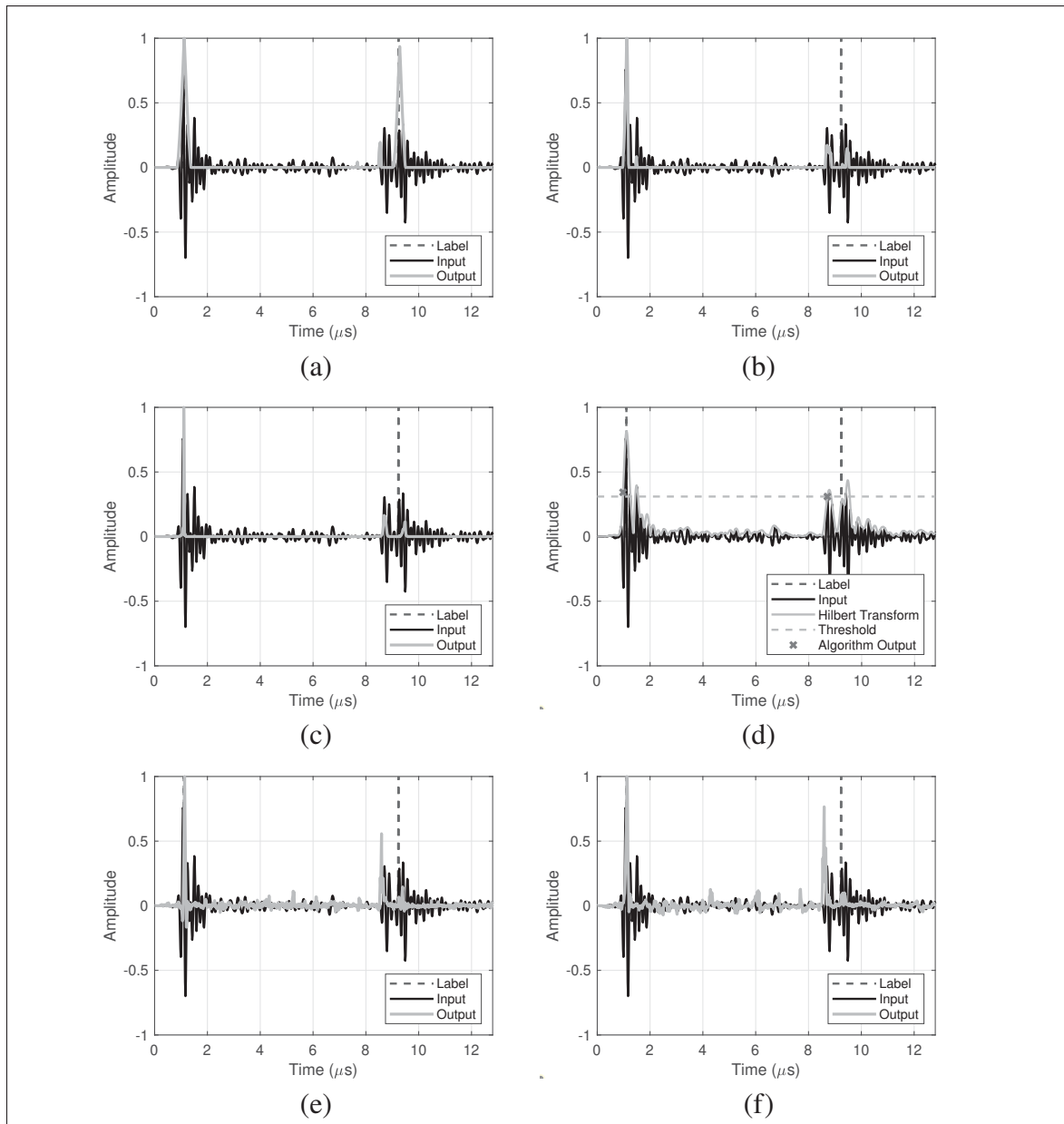


Figure 2.14 Distance and deconvolution prediction of the experimental sample 106 with :  
 (a) TNN, (b) Chapon's CNN, (c) Shpigler's CNN, (d) Thresholding technique, (e)  
 ResNet-50, and (f) EfficientNet-B0.

Additionally, according to Fig.2.12b and Table 2.2, the Transformer architecture emerged as the most accurate option for predicting distances based on 2.25 MHz and 5 MHz experimental data. Specifically, the TNN was 1.7 times more precise in terms of total MAE compared to the best CNN alternative (ResNet-50). Similarly, the best CNN model was 5.5 times more precise than

traditional thresholding methods, while the TNN variant achieved an even greater improvement, reducing MAE by a factor of 9.3. This highlights the superiority of AI-based methods over conventional techniques.

Finally, despite its relatively high parameter count, the optimized TNN model remains competitive in terms of FLOPS. It is 3.2 times less computationally demanding than the best CNN architecture described in NDE papers (CNN Cantero-Chinchilla), and approximately 21 times less demanding than the top overall CNN model (ResNet-50). Coupled with this efficiency, the TNN consistently outperforms all CNN alternatives for deconvolution tasks in experimental testing, achieving notable improvements in both MAE and success rate, as shown in Fig.2.13.

To better illustrate individual model behaviour on complex experimental samples, Fig.2.14 presents a typical indistinct A-scan for CNN models. Limited by their regional focus, CNNs decomposed each local echo, thereby misinterpreting the dominant one. A secondary echo originating from lateral reflections in the corroded sample was identified by all CNN architectures, introducing interpretive complications. Thanks to the addition of long-range dependencies with the MHSA module, the TNN model successfully identified the correct echo to deconvolve within the main echo packet (Fig.2.14a). Finally, although the thresholding method struggles to distinguish the minimum separation within echo packets, it grasps the average distance reasonably well if the threshold is appropriately selected (Fig.2.14d), as outlined by Cantero-Chinchilla *et al.* (2023). In this specific example, however, it first selects the wrong echo, resulting in failure.

### 2.5.6 Error analysis

Using the 5 optimal TNNs referenced in Table 2.2, 12 unique experimental A-scans were identified as failure cases, with four examples shown in Fig.2.15 for analysis.

Firstly, while the maximum of the Hilbert transform generally provides a reliable reference, it can mislabel data and mislead the ML during training, as observed in Figs.2.15a and 2.15c. In these cases, the deconvolutions failed because the first peak corresponded to an edge reflection rather than the intended reference echo.

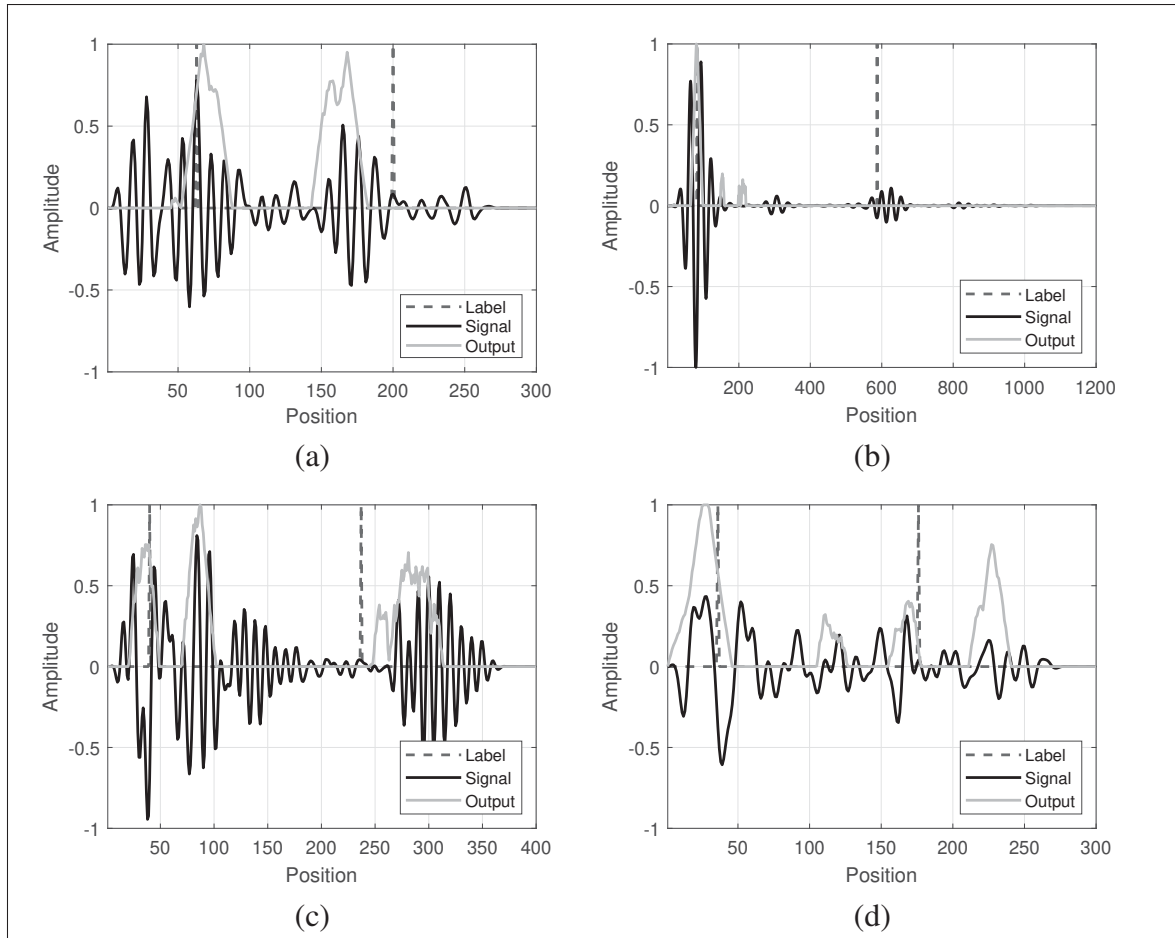


Figure 2.15 Failure cases of the TNN's deconvolution predictions on experimental sample : (a) 622, (b) 6, (c) 489, and (d) 600

Secondly, destructive interference and high surface roughness caused low-amplitude echoes to be reflected back to the probe, as seen in Fig.2.15b. The lack of sufficient 2.25MHz data with low echo amplitudes in the training set caused the model to misinterpret these signals as artifacts rather than valid echo packets.

Finally, mode conversion effects were observed in several A-scans due to angled reflections, producing both longitudinal and transverse waves. These waves, originating from the backwall or edges of the block, can lead to misleading peak amplitudes. As seen in Figs.2.15a and 2.15c., the largest peak did not always correspond to the backwall echo. Additionally, Fig.2.15d highlights

how an unwanted reflection can mimic the first echo waveform, causing the model to incorrectly identify it as the second peak.

## 2.6 Conclusion

In this paper, a novel CNN-TNN architecture was developed to partially deconvolve A-scans of severely corroded profiles on the minimal thickness and was compared against multiple CNN architectures sourced from the literature. Utilizing a simulation-based dataset – developed with Cantero-Chinchilla *et al.* (2023) as a foundation – alongside experimental data acquired from a corroded carbon steel step specimen, the study demonstrated TNN’s superior deconvolution capabilities. The CNN-TNN architecture achieved a notable 1.81% success rate improvement over the best-performing CNN and a 17.5% advantage over established thresholding methods in detecting minimal depths on experimental data, with a resolution under  $0.5 \lambda$  (1.32 mm for 2.25 MHz and 0.59 mm for 5 MHz).

Additionally, the TNN demonstrated exceptional precision and accuracy, achieving an overall MAE that was 1.7 times lower than the widely recognized ResNet-50. Its ability to enhance signal decomposition alongside CNN models further underscores its superior performance in complex signal processing tasks.

Although TNN emerged successful in this study, it is important to bear in mind that demonstrated learning capabilities shown in this paper are confined to specified emission frequencies (around 2.25 MHz and 5 MHz) and sampling rates (62.5 MHz). Moreover, the proposed method focuses on two echoes to identify minimal distances within fixed-length A-scans. This approach requires a pre-processing step to isolate echo pairs and padding to standardize the input length.

Additionally, the limitations of the simulation were highlighted and could be improved through enhanced signal generation or the use of GAN-based synthetic data generation.

Finally, the deconvolved output allows the model to be adapted for various applications. Beyond distance measurement, it can precisely estimate echo arrival times for multimodal TFM, unlocking



new possibilities in advanced imaging and diagnostics. Given its focus on signal processing, the model could also be extended for signal denoising or reconstructing FMC amplitudes from binary FMC data.

### **Acknowledgment and fundings**

This research was funded by Natural Sciences and Engineering Research Council of Canada (NSERC) grant number ALLRP 569922-21, Nucleom and Eddyfi Technologies.

### **Declaration of generative AI and AI-assisted technologies in the writing process**

During the preparation of this work the authors used ChatGPT in order to improve the readability and language of the manuscript. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.



## **CHAPITRE 3**

### **AMÉLIORATION DU MODÈLE**

Les problèmes identifiés dans le chapitre 2 sont les suivants :

1. Surapprentissage (overfitting) dû à un nombre trop élevé de paramètres à entraîner.
2. Performances limitées sur les données expérimentales après un entraînement réalisé principalement avec des données simulées.
3. Déséquilibre entre les données expérimentales et simulées, où la prédominance des données simulées entraîne un surapprentissage sur ces dernières.
4. Stratégie d'entraînement non optimisée en fonction de la répartition des ensembles de données.

Pour remédier à ces limitations, nous allons dans un premier temps examiner en détail l'architecture CNN-TNN proposée au chapitre 2 afin d'identifier les sources du surapprentissage et de concevoir une alternative plus performante.

Ensuite, nous chercherons à améliorer le modèle de simulation, avec pour objectif d'aider le réseau de neurones à interpréter les données expérimentales.

Enfin, nous analyserons la méthodologie d'apprentissage en adaptant notre modèle aux besoins identifiés et en proposant une stratégie d'augmentation de données pour enrichir les ensembles d'entraînement.

### **3.1 Overfit et généralisation du Transformer**

#### **3.1.1 Utilisation d'une séquence aléatoire fixe (seed)**

Pour garantir la répétabilité de l'apprentissage des réseaux de neurones et isoler l'entraînement des impacts liés aux modifications des ensembles de données ou des architectures, l'utilisation d'une seed est essentielle. Fixer une seed impose un séquençement contrôlé des éléments aléatoires, tels que la génération initiale des poids du réseau de neurones ou l'ordre d'entraînement sur les

batches de données. Cela garantit que le GPU exécute les mêmes calculs avec une parallélisation identique à chaque exécution.

En fixant la seed pour les différentes bibliothèques utilisées, notamment Python, NumPy, Keras, TensorFlow, Random, et l'OS, on obtient une courbe d'apprentissage et une évolution de la loss identiques pour un même entraînement, et ce, quelle que soit la session.

De plus, en sélectionnant une seed favorable, il est possible de démarrer l'entraînement avec une répartition initiale des poids qui optimise l'application du réseau de neurones à la tâche cible, offrant ainsi un avantage dès les premières itérations.

Dans la suite de ce mémoire, la même seed sera systématiquement utilisée afin de garantir la comparabilité des résultats entre les différentes expériences.

### **3.1.2 Identification du problème**

Afin d'identifier la source de l'overfitting, une étude d'ablation est menée sur le modèle Transformer. Cette analyse consiste à tracer la courbe du taux de succès sur les 690 données de test expérimentales en fonction des époques, lorsque le modèle est entraîné avec 5000 simulations sur 80 époques et un batch size de 250. Les datasets utilisés sont identiques à ceux présentés au chapitre 2.

Pour isoler les sources potentielles de surapprentissage, le modèle a été entraîné selon plusieurs configurations comparatives : avec et sans Transformer, avec un FFN modifié, puis sans le module MHSA. Les résultats obtenus sont présentés à la Figure 3.1.

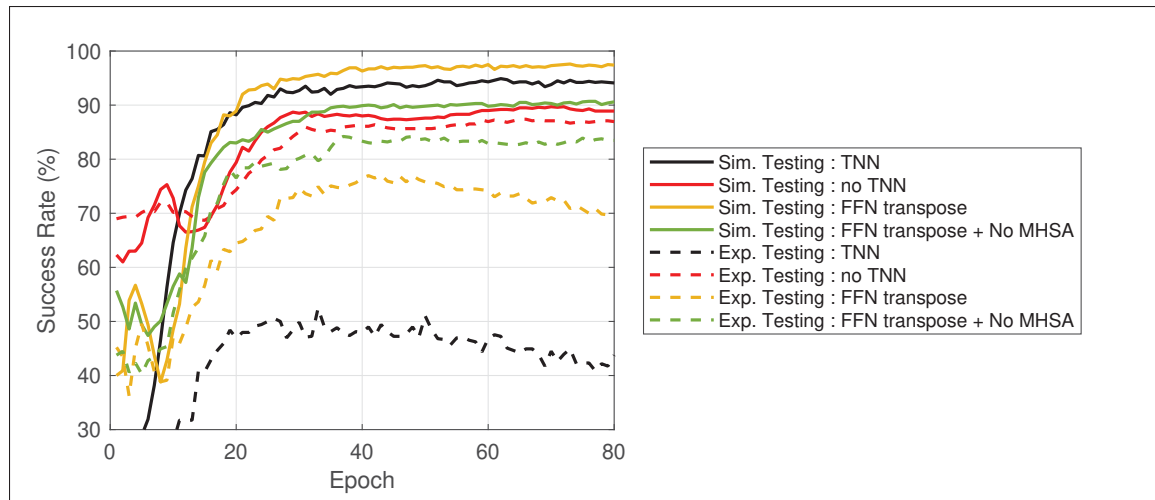


Figure 3.1 Graphique comparatif du taux de succès en fonction des époques avec modifications du modèle CNN-TNN.

Le graphique met en évidence que l'overfitting est principalement causé par la présence du TNN. Une première cause provient du FFN, qui contient 11,520,000 paramètres en augmentant la dimension de (16, 1200) à (16, 4800) avant de revenir à (16, 1200). Pour explorer des alternatives, nous avons proposé de transposer l'entrée du FFN afin de lier les filtres entre eux. Dans cette configuration, le FFN modifie la dimension de (1200, 16) à (1200, 64) avant de revenir à (1200, 16), réduisant ainsi le nombre total de paramètres à 2,048.

Cette modification a permis d'améliorer significativement les performances du modèle sur les datasets de simulation et expérimental, comme le montre le graphique. Cependant, l'overfitting persiste en partie. En maintenant le FFN transposé, nous avons ensuite retiré le module MHSA pour étudier son impact. Comme l'indique la courbe verte, cette suppression améliore considérablement la capacité du réseau à généraliser, atteignant des résultats proches de ceux obtenus sans TNN. Toutefois, cette modification a également un impact négatif important sur les performances du modèle sur le dataset de simulation, sur lequel il est entraîné.

Ces résultats suggèrent que le MHSA joue un rôle clé sur le succès global du réseau, mais aussi sur son manque de généralisation. Par conséquent, le supprimer n'est pas une solution viable.

En conclusion, cette étude d'ablation a mis en évidence que le FFN non transposé et le MHSA sont les principales causes du surapprentissage du modèle. Ces éléments devront être optimisés ou adaptés pour résoudre pleinement le problème d'overfitting.

### 3.1.3 R-drop

Comme mentionné en section 2.5.4, le surapprentissage est un problème récurrent dans les architectures Transformer. Le Vision Transformer (ViT) a fait l'objet de nombreuses recherches sur ce sujet. Pour y remédier, des méthodes avancées de dropout, telles que le R-Drop, ont été développées, atteignant des résultats de pointe dans la classification avec le ViT (liang *et al.*, 2021; Salehin & Kang, 2023).

En s'inspirant des succès obtenus avec le ViT, le R-Drop est ici utilisé et évalué afin d'améliorer la généralisation de notre modèle, en renforçant la cohérence entre les prédictions.

Cette méthode, illustrée à la Figure 3.2, repose sur le calcul de l'erreur quadratique moyenne (MSE) de prédiction deux fois pour une même entrée  $x$ , mais sous deux configurations différentes de dropout, générant ainsi deux sorties,  $y_1$  et  $y_2$ . Ces deux prédictions sont comparées à l'aide de la divergence de Kullback-Leibler ( $D_{KL}$ ), laquelle est intégrée à la perte totale avec un facteur de pondération  $\alpha = 0.5$ .

$$\text{Loss}(x) = \text{MSE}(y_1) + \text{MSE}(y_2) + \alpha \cdot D_{KL}(y_1||y_2) \quad (3.1)$$

avec

$$D_{KL}(y_1||y_2) = \sum_{i=1}^N \left( y_1(i) \cdot \log \left( \frac{y_1(i)}{y_2(i)} \right) \right) \quad (3.2)$$

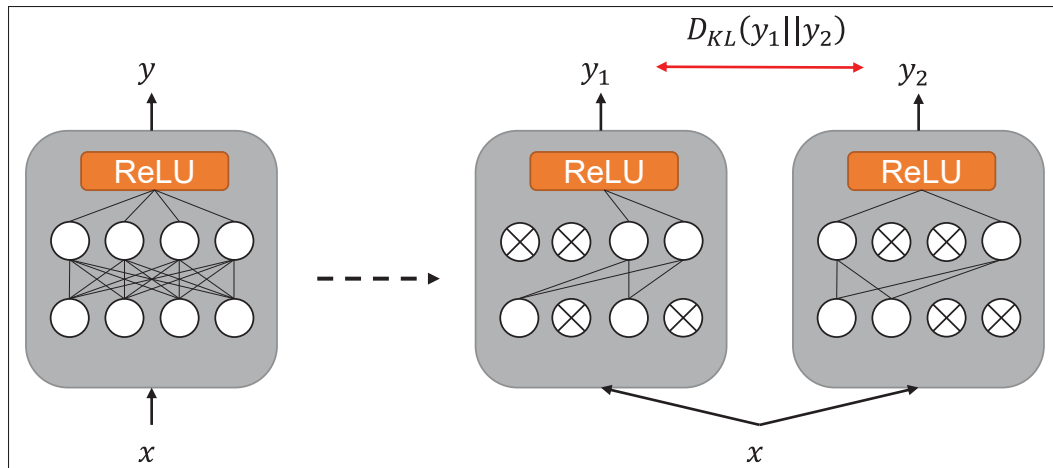


Figure 3.2 Schéma illustrant le principe du R-Drop : l'entrée  $x$  traverse deux fois le modèle représenté à gauche, générant deux sorties  $y_1$  et  $y_2$  qui diffèrent en raison du sous-modèle induit par un dropout avec un taux  $p = 0.5$  illustré sur l'image de droite.

Pour évaluer l'impact du R-Drop sur l'apprentissage du modèle, la méthodologie décrite en section 3.1.2 est appliquée, et les résultats sont présentés à la Figure 3.3.

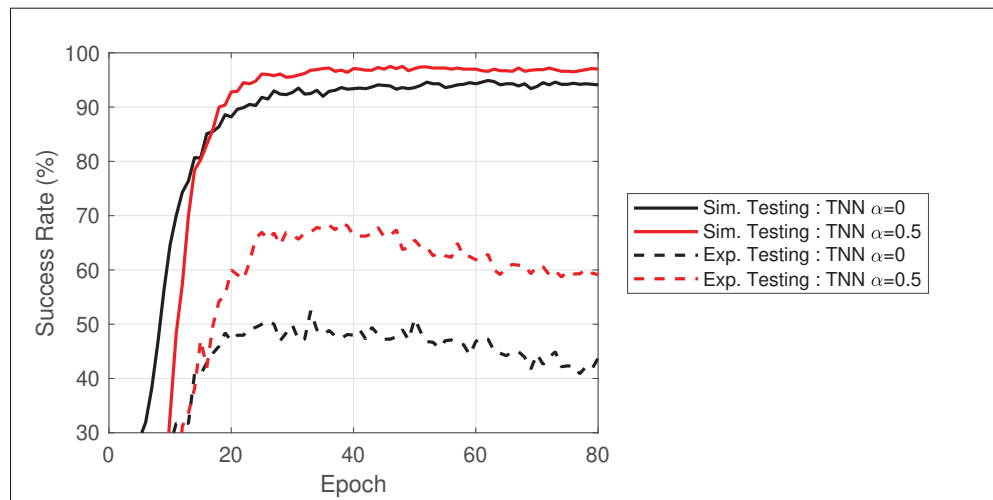


Figure 3.3 Graphique comparatif du taux de succès en fonction des époques avec ( $\alpha = 0.5$ ) et sans ( $\alpha = 0$ ) R-drop.

Bien que l'entraînement soit plus lent, le modèle CNN-TNN parvient à obtenir de meilleurs taux de succès, avec une augmentation de 2,6% sur le dataset de simulation test et de 16,2% sur le dataset expérimental. Ces résultats montrent que le R-Drop est aussi efficace sur notre architecture qu'il l'est sur le ViT. Par conséquent, il sera adopté dans la suite de ce travail pour améliorer l'apprentissage et la généralisation du modèle.

Cependant, le R-Drop à lui seul ne suffit pas pour atteindre les performances souhaitées. En effet, le modèle présente encore un taux de succès inférieur de 18,7% sur le dataset expérimental par rapport à la version sans TNN (Figure 3.1).

#### 3.1.4 Transformer Lite

Pour atténuer le problème d'overfitting associé au MHSA, nous proposons une stratégie qui combine l'efficacité du MHSA sur le dataset d'apprentissage avec la capacité de généralisation des couches convolutives. En nous inspirant des travaux de Wu *et al.* (2020) et de leur architecture Lite Transformer, l'entrée du MHSA est divisée en deux parties.

Une première partie est transmise à la couche MHSA, tandis que l'autre passe par une couche convolutive suivie d'une normalisation par batch (batch normalization), d'une activation ReLU, et enfin d'un dropout. Les résultats globaux issus du MHSA sont ensuite concaténés avec les résultats locaux produits par la couche convolutive. Cette approche permet ainsi au modèle de bénéficier des avantages des deux types de traitement lors de l'apprentissage.

Lors de la propagation inverse, la couche MHSA profite d'un apprentissage guidé par les paramètres locaux obtenus via la concaténation avec la couche convolutive. Cela favorise effectivement l'apprentissage du modèle, en particulier sur des datasets de petite taille, comme l'ont démontré les travaux de Li *et al.* (2022).

Le modèle modifié résultant de cette méthodologie, cumulé à la transposition avant le FFN évoqué section 3.1.2, est présenté à la Figure 3.4 et les résultats sont présentés Figure 3.5.



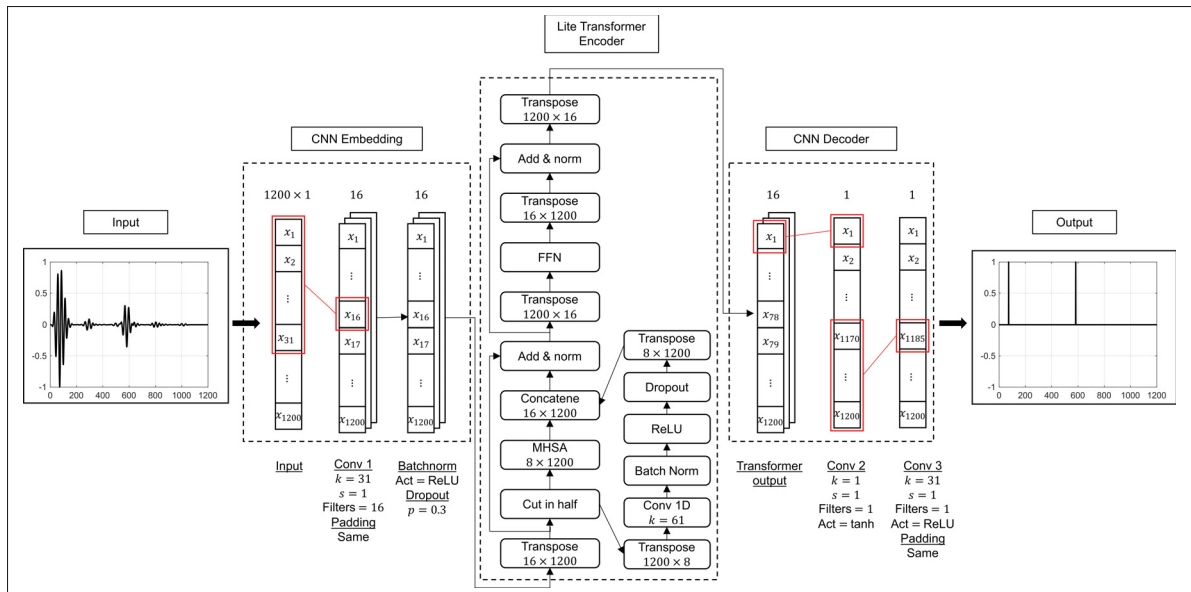


Figure 3.4 Représentation de l'architecture Transformer Lite proposé, optimisée pour la déconvolution.

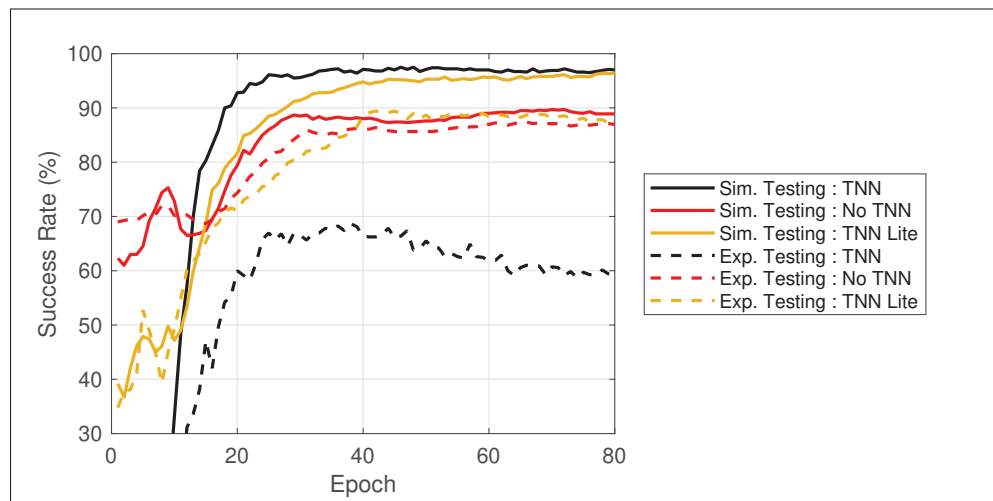


Figure 3.5 Graphique comparatif du taux de succès en fonction des époques des différentes architectures avec un taux de R-drop  $\alpha = 0.5$ .

Bien que le temps d'apprentissage soit plus long pour le TNN Lite, le problème de généralisation est non seulement résolu, mais également surpassé, comme le montre le graphique ci-dessus. Malgré une légère perte de 1,1% du taux de succès sur le dataset de test simulé, le TNN Lite

réalise un gain significatif de 20,7% par rapport au modèle original avec R-drop sur les données expérimentales et de 2,0% par rapport au modèle convolutif sans Transformer. Cela démontre que le TNN Lite généralise mieux qu'un modèle purement convolutif, tout en maintenant un taux de succès élevé sur les données sur lesquelles il est spécialisé.

De plus, le TNN Lite simplifie considérablement le modèle en réduisant le nombre de paramètres : il passe de 11,616,233 paramètres pour le TNN original à seulement 51,121. Cette optimisation réduit également le nombre de FLOPS de 373,8 MFLOPS à 16,4 MFLOPS, ce qui en fait une solution beaucoup plus légère et efficace pour les applications nécessitant à la fois généralisation et performance.

### 3.1.5 Résultats

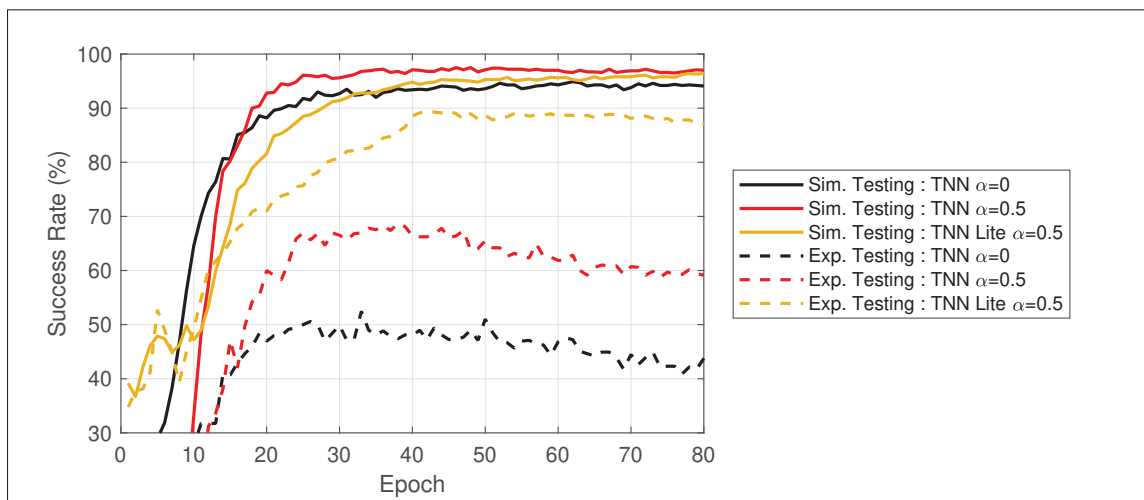


Figure 3.6 Graphique comparatif du taux de succès en fonction des époques regroupant les méthodes appliquées pour répondre au problème de surapprentissage.

Dans cette section, nous avons démontré comment améliorer l'architecture Transformer pour résoudre le problème de généralisation, tout en augmentant ses performances globales et en réduisant sa taille. Le taux de succès sur le dataset expérimental a ainsi progressé de 52,5% à 68,7% grâce à l'intégration du R-drop, puis de 68,7% à 89,4% grâce à la modification de

l'architecture, notamment par l'ajout d'une couche convolutive en parallèle à la couche MHSA et la transposition de l'entrée du FFN.

Concernant les données simulées, le taux de succès est passé de 94,9% à 97,5% avec l'ajout du R-drop, avant de se stabiliser à 96,4% avec l'adoption du TNN Lite. Ces résultats confirment que l'approche proposée permet de maintenir des performances élevées tout en améliorant la généralisation du modèle.

La Figure 3.6 illustre l'évolution des performances à chaque étape d'amélioration, mettant en évidence les bénéfices des différentes optimisations apportées au modèle.

### **3.2 Amélioration de la simulation**

Malgré les améliorations apportées au modèle pour renforcer sa capacité de généralisation entre datasets, une différence de 7,0% subsiste entre le taux de succès sur le dataset simulé et celui sur le dataset expérimental. Cet écart s'explique par le manque de réalisme du modèle de simulation développé au Chapitre 2.

Afin de rapprocher les simulations des données expérimentales et ainsi favoriser l'apprentissage du modèle lors de l'intégration des données expérimentales dans l'entraînement, plusieurs leviers peuvent être ajustés :

1. L'algorithme de génération des signaux
2. L'algorithme de génération des profils de corrosion
3. La taille de la sonde ultrasonore

Les autres paramètres, tels que le SNR, l'amplitude des zones de corrosion ou encore la taille de la pièce, ont déjà été optimisés dans le cadre du Chapitre 2 et ne nécessitent pas de modifications supplémentaires.

### 3.2.1 Signal émis

Le modèle de génération de signal reproduit dans le modèle fréquentiel la bande passante des signaux réels capturés expérimentalement, atteignant une proximité moyenne de 94.86% avec les premiers échos expérimentaux.

Néanmoins, d'autres modèles de génération du signal dans le domaine temporel existent et ont été utilisés avec succès pour entraîner des architectures neuronales (Li *et al.*, 2023).

En s'inspirant de ces travaux, les modèles Gaussien double et Nakagami ont été étudiés pour la génération des signaux dans notre modèle de simulation.

Le modèle Gaussien double est défini par l'équation suivante :

$$s(\Theta, t) = A \left( e^{-\left(\frac{t-\tau-a}{T}\right)^2} + e^{-\left(\frac{t-\tau-b}{T}\right)^2} \right) \sin(2\pi f_c (t - \tau) + \phi) \text{ avec } \Theta = [A, a, b, T, \tau, f_c, \phi] \quad (3.3)$$

Le modèle de Nakagami est défini par :

$$s(\Theta, t) = K \frac{2m^m (t - \tau)^{2m-1}}{\Gamma(m) \Omega^{2m}} e^{-\frac{m}{\Omega^2}(t-\tau)^2} \cos(2\pi f_c (t - \tau) + \phi) \text{ avec } \Theta = [K, m, \Omega, \tau, f_c, \phi] \quad (3.4)$$

Les paramètres  $\Theta$  utilisés sont ceux adaptés des articles de Wang, Wang, LI, Chen & Li (2019) et Lian *et al.* (2019) pour nos fréquences et nombre de cycles  $n_{cycles}$  :

1. Pour le modèle Gaussien double :

$$\Theta = [A = 1, a = 0.9e-6 \cdot \gamma, b = 1.7e-6 \cdot \gamma, T = 0.8e-6 \cdot \gamma, \tau = \frac{n_{cycles}}{f_c}, f_c = 5MHz, \phi = 0] \\ \text{avec } \gamma = \frac{n_{cycles}}{7} \cdot \frac{2.51e6}{f_c}.$$

2. Pour le modèle Nakagami :

$$\Theta = [K = 1, m = 1.9, \Omega = 0.34e-6 \cdot \gamma, \tau = 0.08e-6 \cdot \gamma, f_c = 5MHz, \phi = 0] \text{ avec } \gamma = \frac{n_{cycles}}{5} \cdot \frac{10e6}{f_c}.$$

Enfin, une dernière méthode, proposée par Chapon *et al.* (2021), consiste à utiliser directement les échos expérimentaux reçus comme émetteurs dans la simulation. Pour ce faire, 64 formes d'écho ont été récupérées à l'aide d'une sonde multi-éléments 5 MHz sur le bloc corrodé, ainsi que 18 échos à l'aide d'une sonde mono-élément 2.25 MHz.

La Figure 3.7 illustre les formes d'écho obtenues pour  $n_{cycles} = 3$ . Pour les modèles Gaussien double et Nakagami, les signaux sont générés avec  $n_{cycles} \in \{3, 4, 5\}$ . Lorsque les échos réels sont utilisés, une forme d'écho est sélectionnée aléatoirement parmi les 82 disponibles.

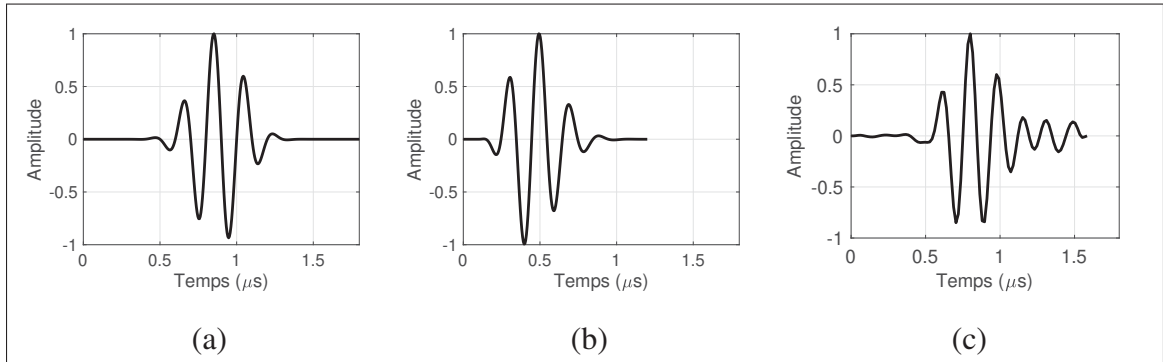


Figure 3.7 Graphique exemple des signaux d'émission (a) Gaussien Double, (b) Nakagami et (c) du 64<sup>ème</sup> élément de la sonde multi-élément.

Pour comparer les différentes méthodes de génération de signal, le modèle présenté en Figure 3.4 est entraîné sur 5000 données simulées, puis validé sur 690 données expérimentales sur une durée de 80 époques avec 100 batchs par époques. L'époque présentant le meilleur taux de succès sur le dataset de validation est retenue comme point de comparaison. Une même seed est utilisée pour garantir la répétabilité des résultats.

Ensuite, les 5000 simulations sont mélangées à 1200 données expérimentales afin d'évaluer l'impact des données simulées sur les performances expérimentales.

Les résultats obtenus sont synthétisés dans la Figure 3.8.

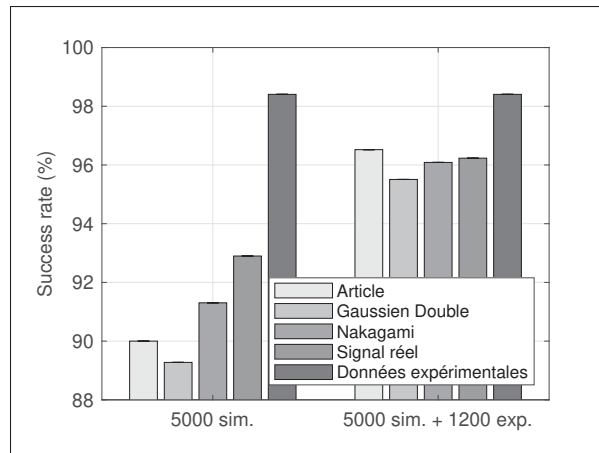


Figure 3.8 Taux de succès en fonction de la méthodologie de génération de signal.

En superposant le signal généré avec le signal réel, le Normalized Cross-Correlation (NCC) maximal atteint avec l'algorithme Gaussien Double est de 90.18%, contre 95.35% pour l'algorithme de Nakagami. Cette différence se reflète dans les résultats présentés à la Figure 3.8, où l'algorithme de génération de signal utilisé au Chapitre 2 (avec un NCC moyen maximal de 94.86%) surpasse celui basé sur le Gaussien Double, mais reste inférieur à l'algorithme de Nakagami.

Similairement, l'algorithme de Nakagami est surpassé par l'utilisation du signal réel dans les simulations. Ces observations mettent en évidence une tendance : plus la forme du signal simulé se rapproche du signal réel, plus la simulation contribue à l'apprentissage et à la généralisation du réseau sur le dataset expérimental.

Ces conclusions restent valides même lorsque des données expérimentales sont ajoutées au dataset d'entraînement. En effet, un dataset de simulation plus réaliste permet à l'IA d'obtenir de meilleurs taux de succès que ce soit avec ou sans données expérimentales. L'exception notable concerne le dataset présenté dans l'article Chapitre 2, qui dépasse les performances obtenues avec le signal réel. Cette anomalie pourrait s'expliquer par une différence dans la taille de la

sonde utilisée lors des simulations : 6.35 mm contre 10 mm. Cette hypothèse sera approfondie dans la prochaine section.

Enfin, un écart de 5.5% persiste entre le taux de succès de l'IA entraînée uniquement sur des simulations et celui de l'IA spécialisée sur le dataset expérimental. De plus, l'ajout des 5000 simulations aux 1200 données expérimentales entraîne une diminution de 1.9% du taux de succès sur le dataset expérimental de test.

Ces résultats confirment l'impact négatif des simulations sur la performance du modèle et mettent en évidence leur manque de réalisme.

### 3.2.2 Corrosion et taille de sonde

Pour se rapprocher davantage du dataset expérimental, l'utilisation de profils de corrosion réels peut être envisagée. À partir du scan laser réalisé sur la pièce corrodée (Figure 2.4), une interpolation bi-dimensionnelle quadratique est effectuée avec un pas de  $40 \mu\text{m}$ , correspondant à la limite de précision du laser, afin d'obtenir une distribution régulière des points du profil de corrosion.

Cette approche permet de générer environ 85,000 profils de corrosion réels uniques d'une longueur de  $L_x = 45 \text{ mm}$ . Pour les intégrer à la simulation, ces profils sont ensuite interpolés de manière à respecter un pas  $dx = \frac{\lambda_{\text{shear}}}{20}$ . La figure 3.9 illustre des extraits de profils de corrosion extrudés à partir du scan laser.

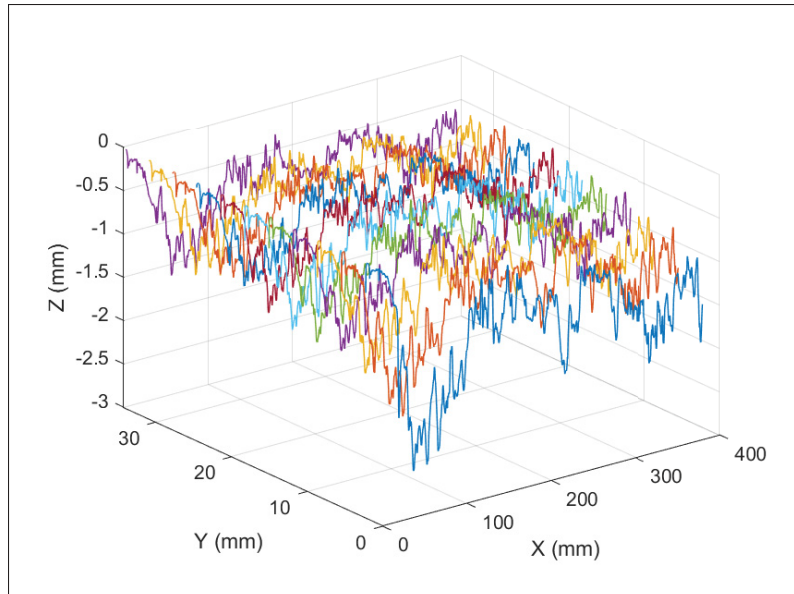


Figure 3.9 Profils de corrosion réel

Le dernier point à considérer est la taille de la sonde. En effet, la sonde mono-élément est une sonde circulaire de diamètre 10 mm, tandis que les éléments de la sonde multi-éléments sont rectangulaires, mesurant  $0.42 \text{ mm} \times 10 \text{ mm}$ .

L'objectif est donc, dans un premier temps, d'évaluer l'effet de la corrosion réelle en utilisant des éléments de 10 mm pour les sondes mono-élément et multi-éléments. Ensuite, les simulations à 5 MHz seront ajustées pour refléter la taille réelle des éléments (0.42 mm), tandis que la taille restera 10 mm pour les simulations à 2.25 MHz.

Deux méthodologies de labellisation peuvent alors être envisagées :

1. Méthode directe : la distance est récupérée directement sous la sonde, sans prendre en compte l'ouverture du faisceau (Beam Opening).
2. Méthode élargie : la labellisation intègre une partie du Beam Opening (BO), permettant ainsi de définir une zone étendue plutôt que les uniques points sous la sonde. Afin de correspondre aux conditions expérimentales, une largeur fixe de 10 mm sera retenue pour cette labellisation, indépendamment de l'épaisseur de la pièce.



Les différentes approches sont illustrées dans la Figure 3.10.

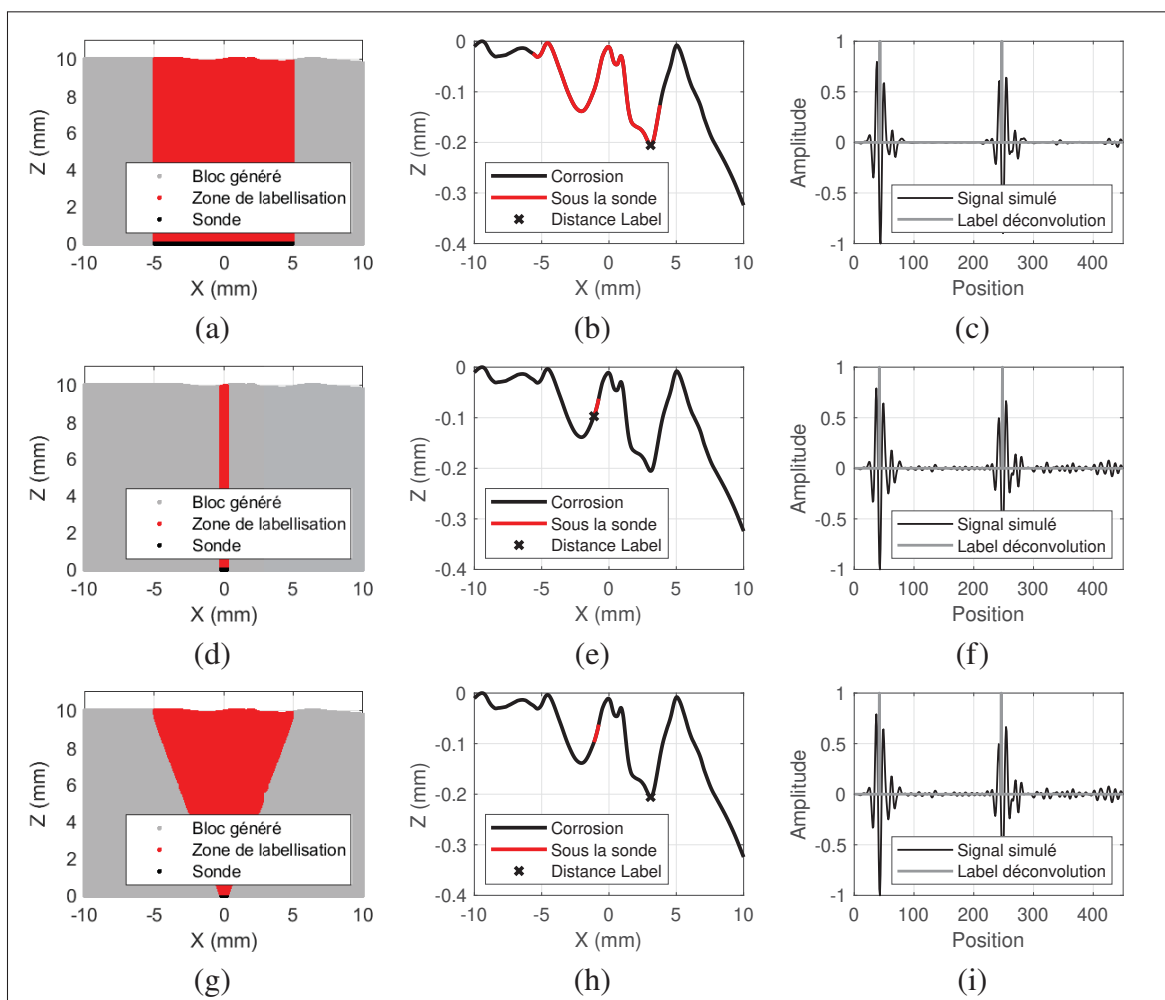


Figure 3.10 Schématisation des considérations de labellisation : corrosion réelle (a-c), avec largeur adaptée (width) (d-f), et avec ouverture de faisceau (BO) (g-i). (a), (d) et (g) représentent le bloc simulé avec la zone de labellisation. (b), (e) et (h) montrent le profil de corrosion utilisé et la position de la distance minimale pour la labellisation. (c), (f) et (i) illustrent le A-scan simulé avec la labellisation associée à la distance minimale.

L'évaluation de ces différentes méthodes est réalisée en appliquant la même approche que celle décrite en section 3.2.1, avec un signal généré à partir du signal réel. Les résultats obtenus sont présentés dans la Figure 3.11.

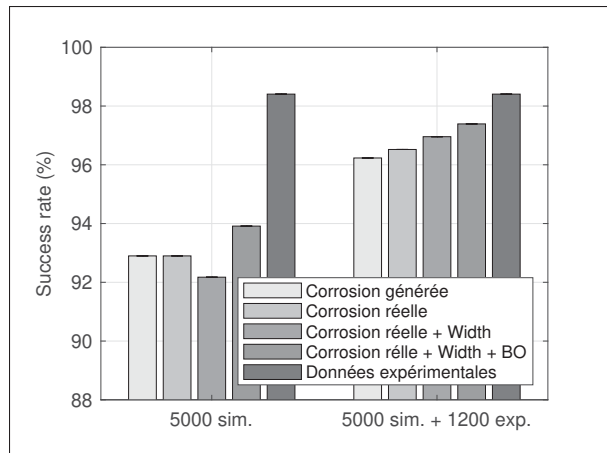


Figure 3.11 Taux de succès en fonction de la méthodologie de corrosion, de la taille de la sonde et de la méthode de labellisation.

L'utilisation de profils de corrosion réels ne modifie pas le taux de succès sur le dataset expérimental lorsque l'IA est entraînée uniquement sur des simulations. En revanche, elle améliore l'apprentissage lorsque des données expérimentales sont ajoutées, augmentant ainsi le taux de succès de 0.3%.

La réduction de la taille de la sonde de 10 mm à 0.42 mm sans prendre en compte le Beam Opening (BO) pour la labellisation dégrade les performances lorsque l'entraînement repose uniquement sur des simulations. Cependant, cette approche devient bénéfique avec l'ajout de données expérimentales, entraînant une hausse supplémentaire de 0.45%.

Enfin, l'intégration du BO dans la labellisation des données, combinée à l'adaptation de la taille de la sonde, donne les meilleurs résultats en entraînement sur simulations, avec un taux de succès atteignant 93.91%, contre 92.90% initialement (corrosion réelle). Cet avantage se confirme avec l'ajout de données expérimentales, améliorant encore le taux de succès de 0.45%.

### 3.2.3 Résultats

L'amélioration du dataset de simulation a un impact significatif sur l'apprentissage de l'IA sur les données expérimentales, aussi bien en l'absence qu'en présence de données expérimentales lors de l'entraînement.

En se basant uniquement sur des données simulées, le modèle CNN-TNN Lite a vu son taux de succès sur les données expérimentales passer de 90.00% à 93.91%. Avec l'ajout de données expérimentales, ce taux est passé de 96.52% à 97.39%, contre 98.40% lorsque l'IA est exclusivement entraînée sur des données expérimentales (voir Fig. 3.12).

Ces résultats confirment toutefois l'écart persistant entre les données simulées et expérimentales, soulignant l'importance des données réelles dans l'apprentissage. Ils mettent également en évidence les limites de la méthodologie actuelle, qui n'est pas entièrement optimisée pour l'entraînement de l'IA sur des datasets hétérogènes.

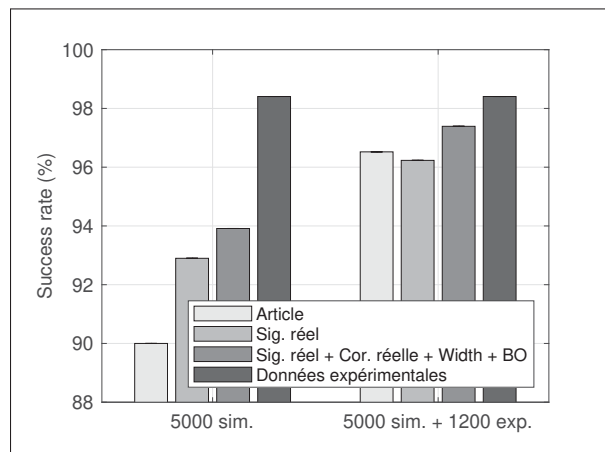


Figure 3.12 Taux de succès en fonction de la méthodologie de simulation.

### 3.3 Méthodologie d'apprentissage

Dans cette dernière partie, une méthodologie d'entraînement optimisée pour le Lite CNN-TNN sera développée afin d'améliorer ses performances sur des datasets hétérogènes. L'objectif est de concevoir un modèle capable de généraliser à différents contextes, incluant des variations de sondes, matériaux, épaisseurs, fréquences et fréquences d'échantillonnage.

Dans un premier temps, l'efficacité de l'apprentissage par transfert sera comparée à la concaténation des datasets, méthode utilisée jusqu'à présent.

Enfin, une approche d'augmentation de données sera explorée afin d'améliorer les performances d'entraînement et d'accroître la capacité de généralisation du modèle.

#### 3.3.1 Concaténation contre apprentissage par transfert (Transfer Learning)

L'apprentissage par transfert (Transfer Learning, TL) est largement utilisé dans le domaine de l'apprentissage automatique (Machine Learning, ML) pour accélérer l'entraînement des modèles complexes et améliorer leurs performances sur une application spécifique (Zhuang *et al.*, 2021).

Dans un premier temps, le modèle est entraîné sur un dataset de référence similaire à celui d'intérêt, mais où les données labellisées sont plus accessibles. Ici, les données simulées sont plus facilement disponibles que les données expérimentales, rendant leur exploitation plus simple et moins coûteuse. Entraîner le modèle sur un grand volume de données simulées permet ainsi d'obtenir un point de départ plus proche de l'objectif final.

Une fois pré-entraîné, le modèle possède des poids optimisés, facilitant son adaptation aux données expérimentales lors d'un second entraînement. Cette approche accélère la convergence et améliore la précision du modèle, car les poids initiaux sont déjà adaptés aux caractéristiques générales du problème.

Différentes stratégies de TL sont décrites dans Zhuang *et al.* (2021). Certaines impliquent de geler des couches pour préserver leurs poids, mais la méthode adoptée ici consiste à réentraîner entièrement le modèle sur le nouveau dataset afin d’affiner ses poids pour l’objectif visé.

Le modèle est d’abord pré-entraîné pendant 80 époques sur un dataset de 600, 1200, 2400, 4800, puis 10 800 données simulées, en utilisant la meilleure méthode de simulation identifiée en section 3.2. Chaque version est validée sur un dataset de 690 données expérimentales, et le meilleur taux de succès atteint sur les 80 époques est retenu pour le TL. Ensuite, le TL est réalisé avec 100 puis 1200 données expérimentales et validé sur le même dataset de 690 données expérimentales. Le taux de succès optimal atteint en 80 époques est enregistré.

Le batch size est ajusté pour obtenir 100 batchs par entraînement. De plus, le learning rate utilisé pour le TL est fixé à 1/5 de celui employé lors du pré-entraînement, afin de limiter le sur-apprentissage sur les données expérimentales et de préserver la capacité de généralisation du modèle. Cette généralisation est vérifiée à l’aide du dataset de test composé des 1000 simulations inconnues de l’IA.

Enfin, la méthode TL est comparée à la concaténation des datasets simulés et expérimentaux, utilisée jusqu’à présent, afin d’évaluer l’impact des deux approches sur l’apprentissage du modèle. Les résultats sont regroupés Figure 3.13.

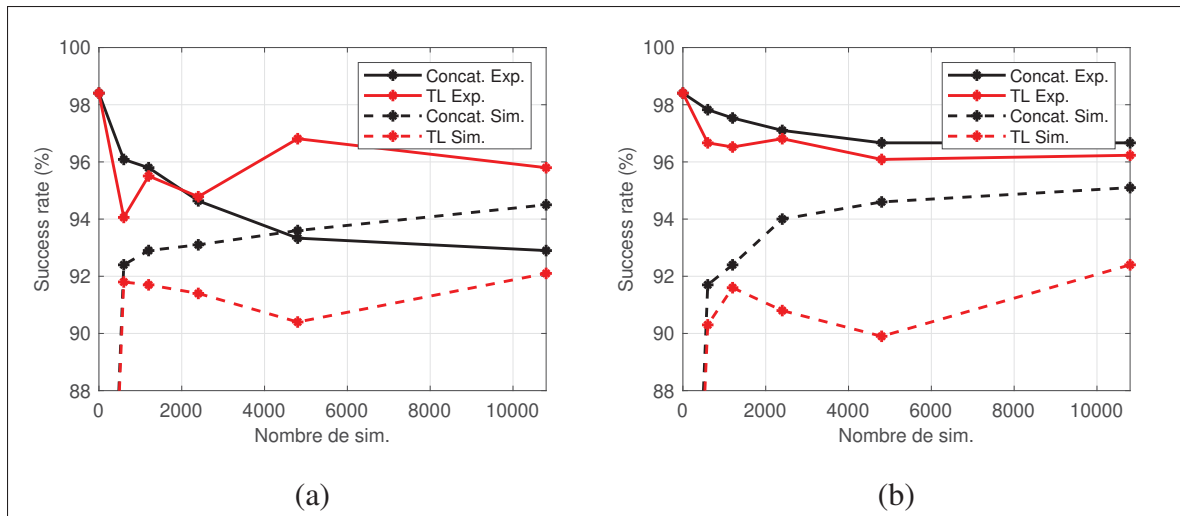


Figure 3.13 Taux de succès en fonction du nombre de simulations et de la méthodologie d'apprentissage avec (a) 100 données expérimentales et (b) 1200 données expérimentales.

Lorsque le dataset d'entraînement expérimental est limité (100 données), le TL permet d'améliorer les performances sur les données expérimentales en spécialisant l'IA. Toutefois, cette approche entraîne une baisse du taux de succès sur les données simulées.

En revanche, lorsque le dataset expérimental est plus conséquent (1200 données), la concaténation des datasets s'avère plus efficace. Cette méthode garantit de meilleurs résultats, quel que soit le nombre de données simulées, aussi bien sur le dataset de validation expérimental que sur le dataset de test simulé.

Ainsi, le TL est particulièrement adapté lorsque le nombre de données labellisées disponibles est limité, mais au détriment de la capacité du modèle à généraliser sur plusieurs datasets ou contextes.

Par ailleurs, la répartition et le volume des données d'entraînement jouent un rôle clé dans l'efficacité du modèle. Une proportion trop élevée de données simulées par rapport aux données expérimentales favorise l'apprentissage sur les simulations, mais se traduit par une baisse de performance sur le dataset expérimental.

L'objectif est d'atteindre un taux de succès optimal à la fois sur les données expérimentales (98,4%) et sur les simulations (95,1%). Pour y parvenir, il est proposé d'augmenter le nombre de données expérimentales afin d'obtenir un meilleur équilibre entre les deux ensembles de données, en exploitant des techniques d'augmentation de données.

### 3.3.2 Augmentation de données

Uhlig *et al.* (2023) ont recensé dans leur revue les principales méthodes d'augmentation de données (Data Augmentation, DA) en une ou deux dimensions pour le domaine des ultrasons. Quatre approches seront étudiées afin d'identifier la plus efficace :

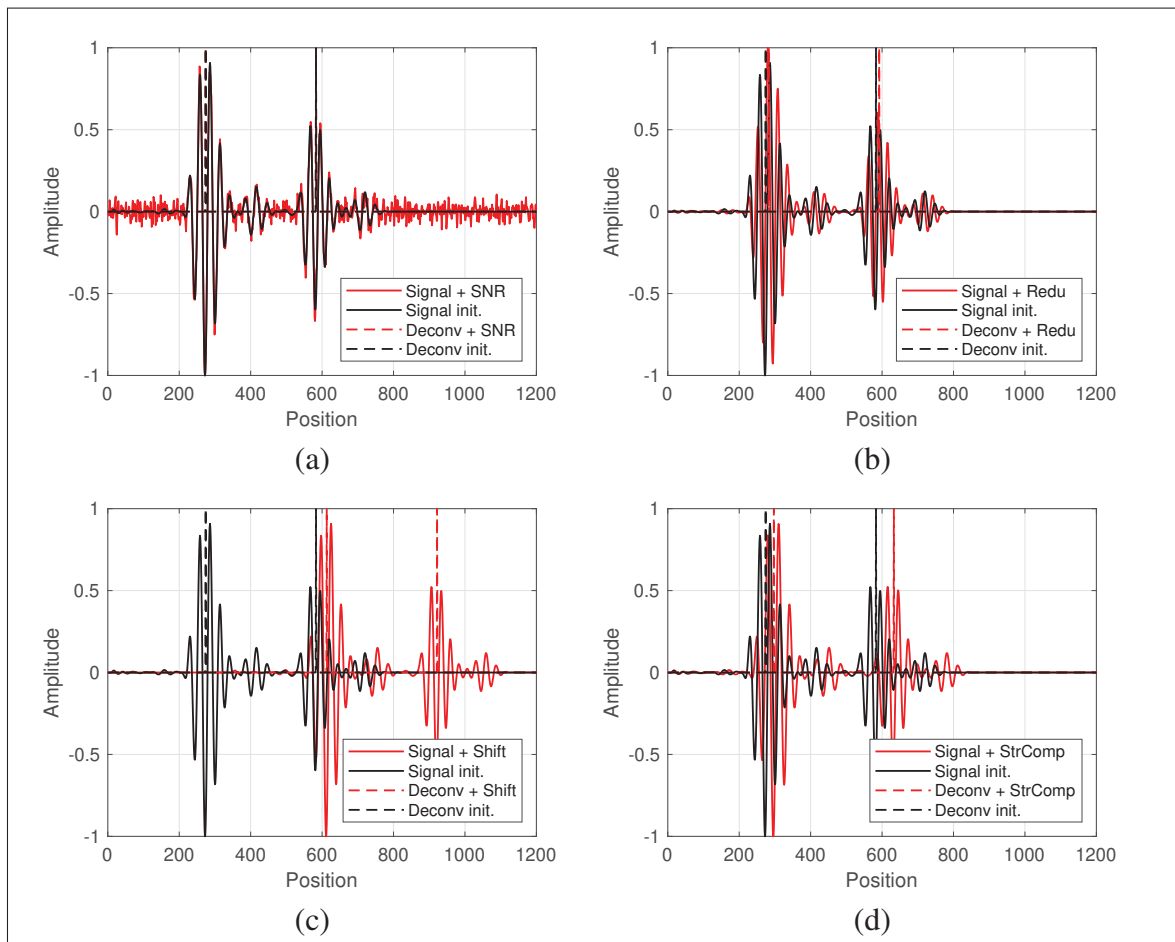


Figure 3.14 Méthodes d'augmentation de données : (a) ajout de bruit (SNR), (b) modification de l'ordre du filtre passe-bande (Redu), (c) décalage temporel (Shift) et (d) compression ou étirement du signal (StrComp)

1. Ajout de bruit (SNR) : Un bruit blanc est généré aléatoirement avec un SNR compris entre 5 et 40 dB, en fonction de l'amplitude maximale du premier écho. Ce bruit est ensuite filtré à l'aide du même filtre que le signal avant d'être ajouté au A-scan initial.
2. Réduction de bruit (Redu) : L'ordre du filtre passe-bande Butterworth (2-10 MHz) appliqué au signal est modifié de 1 à 10 par incrément de 1.
3. Décalage temporel (Shift) : Le A-scan est décalé temporellement de manière aléatoire entre le début du premier écho et la fin du second, garantissant ainsi la conservation des informations essentielles.
4. Étirement et compression (StrComp) : Le signal est interpolé à l'aide d'une méthode polynomiale pour obtenir une fréquence d'échantillonnage variant aléatoirement entre 57,5 MHz et 67,5 MHz, entraînant ainsi un étirement ou une compression du signal en fonction de la fréquence choisie.

Les datasets augmentés sont ensuite labellisés conformément au protocole décrit en section 2.3.2.1, puis ajoutés au dataset d'origine. Les différentes transformations appliquées sont illustrées en Figure 3.14.

Le modèle Lite CNN-TNN est entraîné sur les datasets augmentés en multipliant par 10 le volume des données expérimentales (soit 1200 données originales + 10 800 générées par augmentation), et le nombre de simulations est adapté en conséquence, pour un total de 12 000 données expérimentales et 12 000 simulations. L'entraînement est réalisé sur 80 époques, et le meilleur taux de succès est enregistré sur un dataset de validation de 690 données expérimentales, puis testé sur 1000 simulations inconnues.

Enfin, les méthodes sont testées individuellement, combinées ou concaténées pour explorer l'ensemble des possibilités. Par exemple, la méthode SNR est combinée à la méthode Redu en appliquant un filtre d'ordre plus élevé à la fois au bruit et au A-scan, générant ainsi une plus grande diversité de signaux. De même, SNR peut être concaténée avec Redu, augmentant le dataset expérimental à 22 800 données (1200 originales + 10800 SNR + 10800 Redu). Dans ce



cas, le dataset de simulations est ajusté pour contenir également 22 800 simulations, portant ainsi le volume total des données à 45 600.

Le tableau 3.1 répertorie l'ensemble des combinaisons testées avec leur numéro d'identification, et leurs performances sont présentées en Figure 3.15.

Tableau 3.1 Méthodes combinées ou concaténées  
selon le numéro du dataset.

Numéro	Méthode 1	Méthode 2	Méthode 3	Méthode 4
1	Référence			
2	SNR			
3	Redu			
4	Shift			
5	StrComp			
6	SNR	Redu		
7	SNR	Shift		
8	SNR	StrComp		
9	Redu	Shift		
10	Redu	StrComp		
11	Shift	StrComp		
12	SNR	Redu	Shift	
13	SNR	Redu	StrComp	
14	SNR	Shift	StrComp	
15	Redu	Shift	StrComp	
16	SNR	Redu	Shift	StrComp

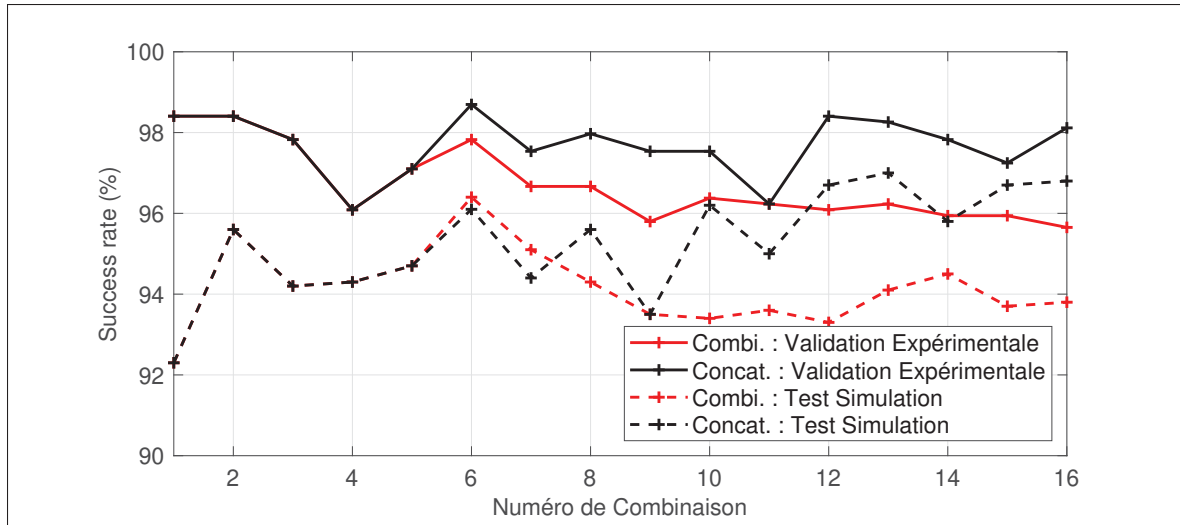


Figure 3.15 Comparaison du taux de succès en fonction du numéro de combinaison décrit tableau 3.1 entre la combinaison des méthodes et leur concaténation.

En rendant les signaux trop complexes par combinaison, l'augmentation de données risque d'altérer la signature physique des A-scans et d'introduire un bruit nuisible, réduisant ainsi le taux de succès. Les méthodes Shift et StrComp ne sont donc pas adaptées.

La méthode Shift, en décalant le signal, ajoute du padding entre le début du signal et le premier écho, perturbant ainsi la cohérence physique du A-scan liée à la propagation des ultrasons. Quant à StrComp, elle modifie la fréquence d'échantillonnage, ce qui revient à altérer la fréquence du signal. En s'éloignant des caractéristiques spectrales des signaux expérimentaux, le modèle perd en précision sur les signaux d'intérêt, entraînant une diminution du taux de succès de 2,3% et 1,3% respectivement sur le dataset expérimental. Ces résultats soulignent l'importance de la cohérence physique, de la fréquence d'échantillonnage et de la fréquence d'émission des signaux d'entraînement pour garantir la précision du modèle.

À l'inverse, les méthodes SNR et Redu préservent l'intégrité physique du A-scan, en modifiant uniquement la forme du signal sans en altérer la fréquence. Cela explique leur effet bénéfique sur l'apprentissage du modèle. Toutefois, les combiner peut complexifier excessivement les signaux et ne pas nécessairement améliorer l'entraînement du Lite CNN-TNN, comme en témoigne une

baisse du taux de succès de 0,6% sur le dataset expérimental. En revanche, leur concaténation permet d'accroître efficacement la taille du dataset, améliorant ainsi les performances globales du modèle : +0,2% sur le dataset de validation expérimental et +3,8% sur le dataset test simulé.

### 3.3.3 Résultats

Pour entraîner efficacement le modèle Lite CNN-TNN sur des datasets hétérogènes, il est essentiel de disposer d'une quantité suffisante de données dans chaque ensemble. Lorsque les données labellisées sont limitées dans un dataset mais abondantes dans un autre, la méthode de Transfer Learning est à privilégier. En revanche, si un volume suffisant de données est disponible, la concaténation des datasets s'avère plus efficace.

De plus, il a été démontré que l'augmentation des données par ajout de bruit (SNR) ou par modification de l'ordre du filtre (Redu) améliore significativement l'apprentissage du modèle.

Ainsi, la méthode retenue consiste à concaténer le dataset de simulation avec le dataset expérimental, augmenté d'un facteur 10x grâce aux méthodes SNR et Redu. Cela constitue un ensemble d'entraînement de 45 600 données, réparties en 22 800 données simulées et 22 800 données expérimentales (dont 1 200 données de base, 10 800 issues de SNR et 10 800 issues de Redu).

Avec ce dataset, le modèle Lite CNN-TNN atteint un taux de succès de 98,70% sur le dataset de validation expérimental et 96,1% sur le dataset de test simulé.

## 3.4 Conclusion

Le modèle CNN-TNN a été optimisé en un Lite CNN-TNN, intégrant le R-Drop, ce qui a renforcé ses capacités de généralisation tout en réduisant sa complexité et son nombre de paramètres.

Parallèlement, le modèle de simulation a été amélioré en modifiant le signal émis à partir d'échos réels récupérés sur le bloc corrodé, en ajustant la taille de la sonde et en affinant la méthodologie de labellisation, prenant désormais en compte le Beam Opening. Ces améliorations ont permis

au modèle d'apprendre plus efficacement et de mieux généraliser sur les données expérimentales, facilitant ainsi un entraînement plus robuste lors de la concaténation des datasets.

Enfin, la méthodologie d'entraînement a été analysée et optimisée afin d'améliorer les performances sur l'ensemble des datasets. Des méthodes d'augmentation de données ont été identifiées et exploitées pour accroître la précision du modèle sur les données expérimentales et simulées.

Avec un dataset enrichi de 45 600 données, l'architecture peut être optimisée en augmentant la dimension du modèle à  $d_{model} = 1200$  (L) au lieu de 8 (S), améliorant ainsi ses performances, que ce soit pour l'architecture présentée dans l'article ou pour le Lite CNN-TNN. Par ailleurs, bien que l'architecture de l'article utilisé Figure 3.16 reprenne les paramètres initiaux du papier de référence, la taille du filtre en série avec le MHSA a été portée à 61 contre 5 initialement, suite à des optimisations menées en aval.

Tableau 3.2 Nombre de paramètres, complexité de calcul (FLOPS), taux de succès (SR) et erreur absolue moyenne (MAE) des différentes architectures sur les datasets de validation expérimental et de test simulé.

Architecture	#Parameters	#FLOPS	SR exp (%)	SR sim (%)	MAE exp (mm)	MAE sim (mm)
Lite CNN-TNN (S)	51,121	16.4M	98.70	96.10	0.114	0.251
Lite CNN-TNN (L)	5,775,817	50.8M	98.84	97.40	0.109	0.163
Article CNN-TNN (S)	11,627,569	373.8M	98.99	98.20	0.075	0.146
Article CNN-TNN (L)	17,314,321	408.2M	99.28	98.80	0.070	0.109
CNN Chapon	30,913	74.1M	88.84	86.90	0.386	0.668
CNN Shpigler	15,451	36.9M	92.03	90.00	0.357	0.382
CNN Cantero-Chinchilla	650,497	1203M	98.70	95.70	0.166	0.204
EfficientNet-B0	5,586,771	804M	98.55	94.00	0.148	0.334
EfficientNet-B4	19,825,423	3089M	98.99	95.60	0.111	0.250
ResNet-50	26,046,512	7756M	99.28	91.90	0.095	0.601

SR = Success Rate, MAE = Mean Absolute Error

Pour conclure, les différentes architectures ont été comparées à travers un entraînement sur ce nouveau dataset augmenté, suivi d'une évaluation sur un dataset de validation composé de 690 données expérimentales et un dataset de test de 1000 données simulées. Les résultats obtenus, présentés en Figure 3.16 et synthétisés dans le Tableau 3.2, montrent que les modèles Lite CNN-TNN, bien que moins performants que ceux développés dans le Chapitre 2 (Article

CNN-TNN), surpassent les modèles CNN de la littérature, tout en bénéficiant d'une efficacité computationnelle nettement supérieure. En particulier, le Lite CNN-TNN (L) est 24 fois plus rapide que le modèle de Cantero-Chinchilla *et al.* (2023), tout en étant plus précis sur les deux datasets de validation et de test. Les mêmes observations s'appliquent aux modèles CNN issus de la littérature en IA, où EfficientNet-B4 est 60 fois plus lent, affichant une précision équivalente sur le dataset expérimental, mais inférieure sur le dataset de test simulé.

Entraînée sur un dataset encore plus large, l'architecture de l'article se révèle être la plus performante sur les deux datasets, offrant des résultats inégalés en termes de taux de succès et d'erreur absolue moyenne (MAE). Ces résultats confirment ainsi la supériorité des architectures Transformers, tant en précision qu'en efficacité computationnelle.

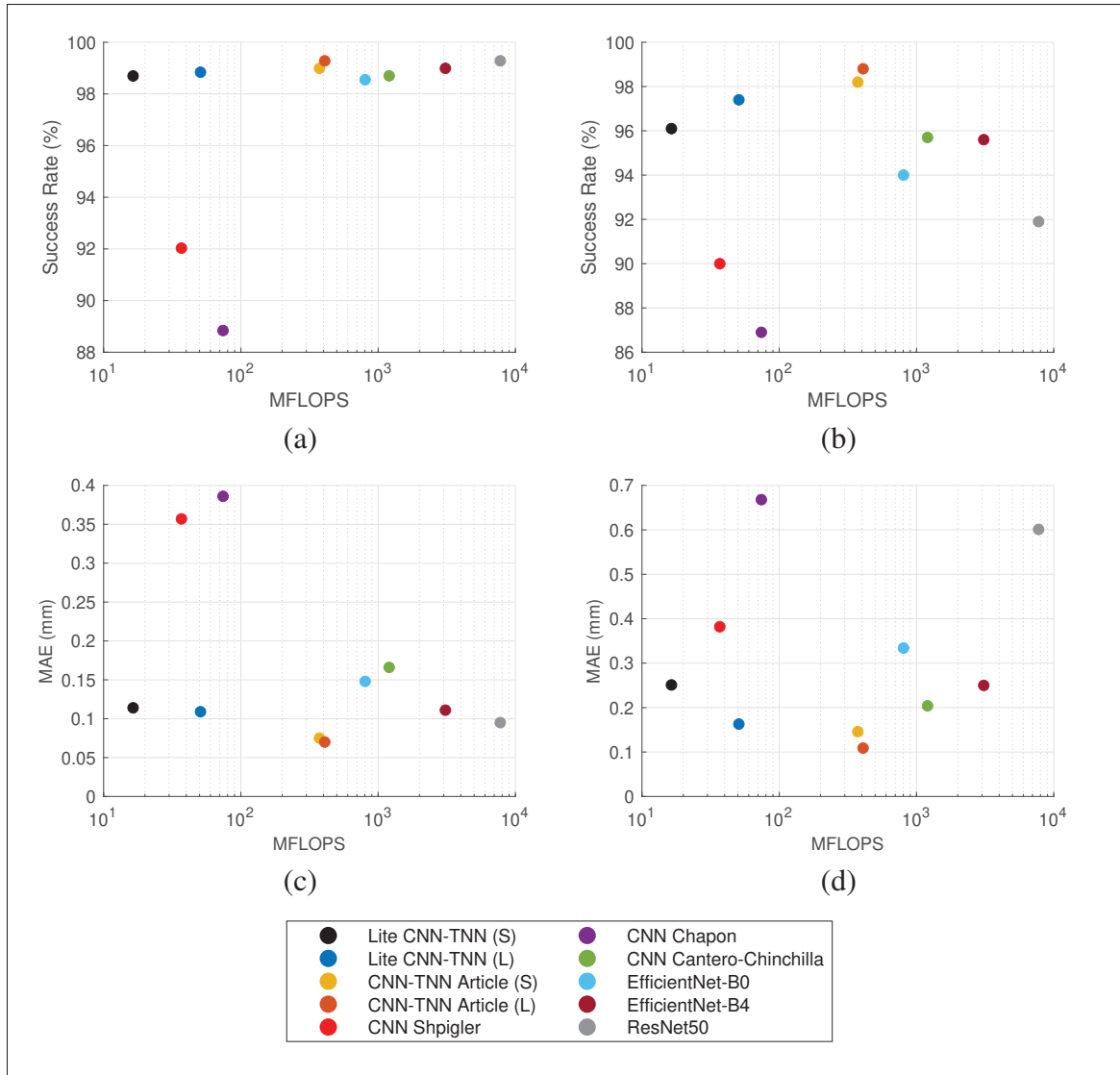


Figure 3.16 Taux de succès (a,b) et MAE (c,d) des différentes architectures sur les datasets (a,c) de validation expérimental et (b,d) de test simulé en fonction de la complexité de calcul (MFLOPS).

## CONCLUSION ET RECOMMANDATIONS

Ce travail de recherche visait à développer un modèle d'intelligence artificielle pour l'évaluation de l'épaisseur minimale des parois par ultrasons, particulièrement dans des configurations complexes où la géométrie de la pièce est inconnue. L'objectif était de démontrer que l'intégration de l'architecture Transformer, combinée à un modèle CNN, pouvait surpasser les approches existantes pour déterminer l'épaisseur minimale sous la sonde.

Un modèle hybride CNN-TNN a d'abord été développé, où la composante Transformer permettait d'introduire un contexte global dans l'apprentissage, améliorant ainsi la gestion des signaux complexes et des échos superposés. Ce modèle a montré des résultats supérieurs par rapport aux CNN classiques, aussi bien sur des données expérimentales que simulées. Cependant, des problèmes d'overfitting ont été identifiés, ce qui a conduit à l'introduction du R-Drop et du Lite Transformer pour améliorer la généralisation du modèle en réduisant l'overfitting.

L'optimisation de l'architecture par l'ajout du Lite CNN-TNN intègre une couche convolutive parallèlement au mécanisme de MHSA. Cette approche combine les avantages des CNN pour les caractéristiques locales et des Transformers pour les interactions globales, améliorant la stabilité de l'apprentissage tout en réduisant la complexité du modèle. En parallèle, le modèle de simulation a été enrichi avec des signaux réels et des ajustements de paramètres, améliorant ainsi la correspondance entre les données simulées et expérimentales.

Enfin, des techniques d'augmentation de données ont été appliquées pour renforcer la robustesse du modèle, tout en conservant la fidélité des signaux. Grâce à ces améliorations, le modèle Lite CNN-TNN (L) atteint un taux de succès de 98,84% sur le dataset expérimental et 97,40% sur le dataset simulé, surpassant ainsi le meilleur modèle CNN de la littérature en contrôle non destructif (Cantero-Chinchilla *et al.*, 2023), qui obtient respectivement 98,70% et 95,70%, tout en étant 24 fois plus rapide. Cette tendance se confirme également dans l'estimation de la distance minimale, où les modèles hybrides CNN-TNN affichent un MAE inférieur à celui des

modèles CNN, traduisant une meilleure précision. Ces résultats démontrent que l'intégration du Transformer améliore significativement la précision et la robustesse des modèles, tout en réduisant leur complexité. Ce travail ouvre ainsi la voie à une adoption plus large des architectures hybrides pour les applications industrielles de contrôle non destructif.

Cette étude repose sur l'hypothèse simplificatrice selon laquelle la corrosion se manifeste uniquement par une irrégularité géométrique de surface, sans prise en compte explicite des couches d'oxyde ou des dépôts de corrosion tels que la rouille. Or, ces éléments peuvent modifier localement les propriétés acoustiques du matériau, notamment par l'introduction d'impédances supplémentaires, d'effets de diffusion complexe ou d'absorption accrue. Ces aspects n'ont pas pu être reproduits expérimentalement dans le cadre de ce travail, mais leur intégration dans de futurs travaux permettrait de renforcer la robustesse et la généralisation de la méthode développée.

Des perspectives d'amélioration existent encore, notamment en explorant d'autres méthodes d'augmentation de données, en affinant l'architecture et en testant le modèle sur un plus large panel de configurations réelles. Parmi les ajustements possibles, la suppression de la fenêtre temporelle, peu adaptée aux inspections sur le terrain, pourrait être envisagée. De plus, le modèle pourrait être adapté pour la reconstruction d'images multimodales par la méthode de Total Focusing (TFM). En déterminant directement les temps d'arrivée des échos longitudinaux et transversaux à partir des signaux de la Full Matrix Capture (FMC), il deviendrait possible de reconstruire l'image TFM non pas à partir de l'enveloppe des signaux, mais à partir d'impulsions idéalisées de type Dirac, conduisant ainsi à une amélioration significative de la précision et de la résolution de l'image reconstruite.



## BIBLIOGRAPHIE

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. & Zheng, X. (2016). TensorFlow : A System for Large-Scale Machine Learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283. Repéré à <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>.
- Agarap, A. F. [arXiv :1803.08375 [cs, stat] version : 1]. (2018). Deep Learning using Rectified Linear Units (ReLU). arXiv. Repéré à <http://arxiv.org/abs/1803.08375>.
- Audoin, B. & Roux, J. (1996). An innovative application of the Hilbert transform to time delay estimation of overlapped ultrasonic echoes. *Ultrasonics*, 34(1), 25–33. doi : 10.1016/0041-624X(95)00088-K.
- Ba, J. L., Kiros, J. R. & Hinton, G. E. [arXiv :1607.06450 [cs, stat]]. (2016). Layer Normalization. arXiv. Repéré à <http://arxiv.org/abs/1607.06450>.
- Baltzersen, Ø., Waag, T. I., Johnsen, R., Ahlen, C. H. & Tveit, E. (2007). Wall Thickness Monitoring Of New And Existing Subsea Pipelines Using Ultrasound.
- Barshan, B. (2000). Fast processing techniques for accurate ultrasonic range measurements. *Measurement Science and Technology*, 11(1), 45. doi : 10.1088/0957-0233/11/1/307.
- Biraud, Y. (1976). Les méthodes de déconvolution et leurs limitations fondamentales. *Revue de Physique Appliquée*, 11(2), 203–214. doi : 10.1051/rphysap:01976001102020300.
- Bjorck, N., Gomes, C. P., Selman, B. & Weinberger, K. Q. (2018). Understanding Batch Normalization. *Advances in Neural Information Processing Systems*, 31. Repéré à <https://proceedings.neurips.cc/paper/2018/hash/36072923bfc3cf47745d704feb489480-Abstract.html>.
- Cantero-Chinchilla, S., Simpson, C. A., Ballisat, A., Croxford, A. J. & Wilcox, P. D. (2023). Convolutional neural networks for ultrasound corrosion profile time series regression. *NDT & E International*, 133, 102756. doi : 10.1016/j.ndteint.2022.102756.
- Carpenter, R. N. & Stepanishen, P. R. (1984). An improvement in the range resolution of ultrasonic pulse echo systems by deconvolution. *The Journal of the Acoustical Society of America*, 75(4), 1084–1091. doi : 10.1121/1.390782.
- Cegla, F. & Allin, J. (2015). Ultrasonic Monitoring of Pipeline Wall Thickness with Autonomous, Wireless Sensor Networks. Dans *Oil and Gas Pipelines* (pp. 571–578). John Wiley & Sons, Ltd. doi : 10.1002/9781119019213.ch39.

- Chang, Y., Zi, Y., Zhao, J., Yang, Z., He, W. & Sun, H. (2017). An adaptive sparse deconvolution method for distinguishing the overlapping echoes of ultrasonic guided waves for pipeline crack inspection. *Measurement Science and Technology*, 28(3), 035002. doi : 10.1088/1361-6501/aa52ae. Publisher : IOP Publishing.
- Chapon, A., Pereira, D., Toews, M. & Belanger, P. (2021). Deconvolution of ultrasonic signals using a convolutional neural network. *Ultrasonics*, 111, 106312. doi : 10.1016/j.ultras.2020.106312.
- Che, C., Zhang, P., Zhu, M., Qu, Y. & Jin, B. (2021). Constrained transformer network for ECG signal processing and arrhythmia classification. *BMC Medical Informatics and Decision Making*, 21(1), 184. doi : 10.1186/s12911-021-01546-2.
- Cheeke, J. D. N. (2002). *Fundamentals and applications of ultrasonic waves*. Boca Raton : CRC Press. Repéré à <http://catdir.loc.gov/catdir/toc/fy032/2002018807.html>.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. & Houlsby, N. [arXiv :2010.11929 [cs]]. (2021). An Image is Worth 16x16 Words : Transformers for Image Recognition at Scale. arXiv. Repéré le 2024-09-23 à <http://arxiv.org/abs/2010.11929>.
- Dubey, A. K. & Jain, V. (2019). Comparative Study of Convolution Neural Network's Relu and Leaky-Relu Activation Functions. *Applications of Computing, Automation and Wireless Systems in Electrical Engineering*, pp. 873–880. doi : 10.1007/978-981-13-6772-4\_76.
- Egaña, A., Seco, F. & Ceres, R. (2008). Processing of ultrasonic echo envelopes for object location with nearby receivers. *IEEE Transactions on Instrumentation and Measurement*, 57(12), 2751–2755. doi : 10.1109/TIM.2008.926408.
- Ferrara, K., Algazi, V. & Liu, J. (1992). The effect of frequency dependent scattering and attenuation on the estimation of blood velocity using ultrasound. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 39(6), 754–767. doi : 10.1109/58.165561. Conference Name : IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control.
- Fox, J., Khuri-Yakub, B. & Kino, G. (1983). High-Frequency Acoustic Wave Measurements in Air. *1983 Ultrasonics Symposium*, pp. 581–584. doi : 10.1109/ULTSYM.1983.198123.
- Frederiksen, T. & Howard, W. (1974). A single-chip monolithic sonar system. *IEEE Journal of Solid-State Circuits*, 9(6), 394–403. doi : 10.1109/JSSC.1974.1050533. Conference Name : IEEE Journal of Solid-State Circuits.

- Garbin, C., Zhu, X. & Marques, O. (2020). Dropout vs. batch normalization : an empirical study of their impact to deep learning. *Multimedia Tools and Applications*, 79(19), 12777–12815. doi : 10.1007/s11042-019-08453-9.
- Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*. MIT Press. Repéré à <https://www.deeplearningbook.org/>.
- Guan, J., Wang, W., Feng, P., Wang, X. & Wang, W. (2021). Low-Dimensional Denoising Embedding Transformer for ECG Classification. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1285–1289. doi : 10.1109/ICASSP39728.2021.9413766.
- Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y. & Pang, R. (2020). Conformer : Convolution-augmented transformer for speech recognition. 2020-October, 5036–5040. doi : 10.21437/Interspeech.2020-3015.
- Hahnloser, R. H. R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J. & Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789), 947–951. doi : 10.1038/35016072. Publisher : Nature Publishing Group.
- Han, W., Zhang, Z., Zhang, Y., Yu, J., Chiu, C.-C., Qin, J., Gulati, A., Pang, R. & Wu, Y. [arXiv :2005.03191 [cs, eess]]. (2020). ContextNet : Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context. arXiv. Repéré le 2023-11-02 à <http://arxiv.org/abs/2005.03191>.
- He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.
- He, X. & Deng, L. (2017). Deep Learning for Image-to-Text Generation : A Technical Overview. *IEEE Signal Processing Magazine*, 34(6), 109–116. doi : 10.1109/MSP.2017.2741510. Conference Name : IEEE Signal Processing Magazine.
- He chen, Li bing & Gao fei. (2021). An Overlapping Echo Separation Method Using Blind Deconvolution in Ultrasonic Testing. *Journal of Xi'an Jiaotong University*, 129–37. doi : 10.7652/xjtuxb202112015. Place : China Publisher : Editorial Board of Journal of Xi'an Jiaotong University.
- Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. doi : 10.1162/neco.1997.9.8.1735. Conference Name : Neural Computation.

- Hoseini, M. R., Wang, X. & Zuo, M. J. (2012). Estimating ultrasonic time of flight using envelope and quasi maximum likelihood method for damage detection and assessment. *Measurement*, 45(8), 2072–2080. doi : 10.1016/j.measurement.2012.05.008.
- Huthwaite, P. (2014). Accelerated finite element elastodynamic simulations using the GPU. *Journal of Computational Physics*, 257, 687–707. doi : 10.1016/j.jcp.2013.10.017.
- Ioffe, S. & Szegedy, C. (2015). Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift. *Proceedings of the 32nd International Conference on Machine Learning*, pp. 448–456. Repéré à <https://proceedings.mlr.press/v37/loff15.html>.
- Jiang, E. (2020). A review of the comparative studies on traditional and intelligent face recognition methods. *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, pp. 11–15. doi : 10.1109/CVIDL51233.2020.00010.
- Kaaresen, K. (1997). Deconvolution of sparse spike trains by iterated window maximization. *IEEE Transactions on Signal Processing*, 45(5), 1173–1183. doi : 10.1109/78.575692. Conference Name : IEEE Transactions on Signal Processing.
- Karita, S., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., Someki, M., Soplin, N. E. Y., Yamamoto, R., Wang, X., Watanabe, S., Yoshimura, T. & Zhang, W. (2019). A Comparative Study on Transformer vs RNN in Speech Applications. *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 449–456. doi : 10.1109/ASRU46091.2019.9003750.
- Kingma, D. P. & Ba, J. [arXiv :1412.6980 [cs]]. (2017). Adam : A Method for Stochastic Optimization. arXiv. Repéré à <http://arxiv.org/abs/1412.6980>.
- Krautkrämer, J., Krautkrämer, H. & Grabendörfer, W. (1977). *Ultrasonic testing of materials* (éd. Second edition). Berlin : Springer. Repéré à <http://dx.doi.org/10.1007/978-3-662-02296-2>.
- Lau, M. M. & Hann Lim, K. (2018). Review of Adaptive Activation Function in Deep Neural Network. *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pp. 686-690. doi : 10.1109/IECBES.2018.8626714.
- LeCun, Y. & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995. Repéré à <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e26cc4a1c717653f323715d751c8dea7461aa105>. Publisher : Citeseer.
- Li, C. & Zhang, C. (2024). Toward a deeper understanding : RetNet viewed through Convolution. *Pattern Recognition*, 155, 110625. doi : 10.1016/j.patcog.2024.110625.

- Li, D., Chen, Z., Zhang, Y., Zhao, S. & Liu, W. (2023). Separation of multi-echo overlapping ultrasonic signals for increasing the axial resolution using a neural network. *Measurement Science and Technology*, 34(12). doi : 10.1088/1361-6501/acefee.
- Li, K., Yu, R., Wang, Z., Yuan, L., Song, G. & Chen, J. (2022). Locality Guidance for Improving Vision Transformers on Tiny Datasets. *Computer Vision – ECCV 2022*, pp. 110–127. doi : 10.1007/978-3-031-20053-3\_7.
- Lian, M., Liu, H., Bo, Q., Zhang, T., Li, T. & Wang, Y. (2019). An improved matching pursuit method for overlapping echo separation in ultrasonic thickness measurement. *Measurement Science and Technology*, 30(6), 065001. doi : 10.1088/1361-6501/ab03ba. Publisher : IOP Publishing.
- liang, x., Wu, L., Li, J., Wang, Y., Meng, Q., Qin, T., Chen, W., Zhang, M. & Liu, T.-Y. (2021). R-Drop : Regularized Dropout for Neural Networks. *Advances in Neural Information Processing Systems*, 34, 10890–10905. Repéré à <https://proceedings.neurips.cc/paper/2021/hash/5a66b9200f29ac3fa0ae244cc2a51b39-Abstract.html>.
- Liu, H., Zhang, L., Liu, H. F., Chen, S., Wang, S., Wong, Z. Z. & Yao, K. (2018). High-frequency ultrasonic methods for determining corrosion layer thickness of hollow metallic components. *Ultrasonics*, 89, 166–172. doi : 10.1016/j.ultras.2018.05.006.
- Mallat, S. & Zhang, Z. (1993). Matching Pursuits With Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing*, 41(12), 3397–3415. doi : 10.1109/78.258082.
- Marioli, D., Narduzzi, C., Offelli, C., Petri, D., Sardini, E. & Taroni, A. (1992). Digital time-of-flight measurement for ultrasonic sensors. *IEEE Transactions on Instrumentation and Measurement*, 41(1), 93–97. doi : 10.1109/19.126639. Conference Name : IEEE Transactions on Instrumentation and Measurement.
- Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill. Repéré à <https://thuvienso.hoasen.edu.vn/handle/123456789/9610>.
- Montavon, G., Orr, G. & Müller, K.-R. (2012). *Neural Networks : Tricks of the Trade*. Springer.
- Mor, E., Azoulay, A. & Aladjem, M. (2010). A matching pursuit method for approximating overlapping ultrasonic echoes. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 57(9), 1996–2004. doi : 10.1109/TUFFC.2010.1647.
- Nandi, A., Mampel, D. & Roscher, B. (1997). Blind deconvolution of ultrasonic signals in nondestructive testing applications. *IEEE Transactions on Signal Processing*, 45(5), 1382–1390. doi : 10.1109/78.575716. Conference Name : IEEE Transactions on Signal Processing.

- Norris, W. E., Naus, D. J. & Graves, H. L. (1999). Inspection of nuclear power plant containment structures. *Nuclear Engineering and Design*, 192(2), 303–329. doi : 10.1016/S0029-5493(99)00125-9.
- Parrilla, M., Anaya, J. & Fritsch, C. (1991). Digital signal processing techniques for high accuracy ultrasonic range measurements. *IEEE Transactions on Instrumentation and Measurement*, 40(4), 759–763. doi : 10.1109/19.85348. Conference Name : IEEE Transactions on Instrumentation and Measurement.
- Posilović, L., Medak, D., Subašić, M., Petković, T., Budimir, M. & Lončarić, S. (2019). Flaw Detection from Ultrasonic Images using YOLO and SSD. *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, pp. 163–168. doi : 10.1109/ISPA.2019.8868929.
- Pu, X., Yi, P., Chen, K., Ma, Z., Zhao, D. & Ren, Y. (2022). EEGDnet : Fusing non-local and local self-similarity for EEG signal denoising with transformer. *Computers in Biology and Medicine*, 151, 106248. doi : 10.1016/j.combiomed.2022.106248.
- Pyle, R. J., Bevan, R. L. T., Hughes, R. R., Rachev, R. K., Ali, A. A. S. & Wilcox, P. D. (2021). Deep Learning for Ultrasonic Crack Characterization in NDE. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 68(5), 1854–1865. doi : 10.1109/TUFFC.2020.3045847. Conference Name : IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control.
- Queiros, R., Correa Alegria, F., Silva Girao, P. & Cruz Serra, A. C. (2010). Cross-Correlation and Sine-Fitting Techniques for High-Resolution Ultrasonic Ranging. *IEEE Transactions on Instrumentation and Measurement*, 59(12), 3227–3236. doi : 10.1109/TIM.2010.2047305. Conference Name : IEEE Transactions on Instrumentation and Measurement.
- Rawat, W. & Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification : A Comprehensive Review. *Neural Computation*, 29(9), 2352–2449. doi : 10.1162/neco\_a\_00990.
- Rosenblatt, F. (1958). The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386–408. doi : 10.1037/h0042519. Place : US Publisher : American Psychological Association.
- Salehin, I. & Kang, D.-K. (2023). A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain. *Electronics*, 12(14), 3106. doi : 10.3390/electronics12143106. Number : 14 Publisher : Multidisciplinary Digital Publishing Institute.



- Shastri, S. K., Rudresh, S., Anand, R., Nagesh, S., Seelamantula, C. S. & Thittai, A. K. (2020). Axial super-resolution in ultrasound imaging with application to non-destructive evaluation. *Ultrasonics*, 108, 106183. doi : 10.1016/j.ultras.2020.106183.
- Shpigler, A., Mor, E. & Bar-Hillel, A. (2022). Detection of overlapping ultrasonic echoes with deep neural networks. *Ultrasonics*, 119, 106598. doi : 10.1016/j.ultras.2021.106598.
- Shull, P. J. (Éd.). (2002). *Nondestructive evaluation : theory, techniques, and applications*. New York Basel : Dekker.
- Sin, S.-K. & Chen, C.-H. (1992). A comparison of deconvolution techniques for the ultrasonic nondestructive evaluation of materials. *IEEE Transactions on Image Processing*, 1(1), 3–10. doi : 10.1109/83.128026. Conference Name : IEEE Transactions on Image Processing.
- Soussen, C., Idier, J., Carcreff, E., Simon, L. & Potel, C. (2012). Ultrasonic non destructive testing based on sparse deconvolution. *Journal of Physics : Conference Series*, 353(1), 012018. doi : 10.1088/1742-6596/353/1/012018.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout : A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958.
- T. Eason, L. J. B. (2017). *Ultrasonic thickness structural health monitoring of steel pipe for internal corrosion*. (Doctor of Philosophy, Iowa State University, Digital Repository, Ames). Repéré à Pages : 11055505.
- Tan, M. & Le, Q. (2019). EfficientNet : Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*, pp. 6105–6114. Repéré à <https://proceedings.mlr.press/v97/tan19a.html>.
- Thakkar, V., Tewary, S. & Chakraborty, C. (2018). Batch Normalization in Convolutional Neural Networks — A comparative study with CIFAR-10 data. *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, pp. 1–5. doi : 10.1109/EAIT.2018.8470438.
- Tropp, J. & Gilbert, A. (2007). Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12), 4655–4666. doi : 10.1109/TIT.2007.909108.
- Uhlig, S., Alkhasli, I., Schubert, F., Tschöpe, C. & Wolff, M. (2023). A review of synthetic and augmented training data for machine learning in ultrasonic non-destructive evaluation. *Ultrasonics*, 134, 107041. doi : 10.1016/j.ultras.2023.107041.

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, 30. Repéré à [https://proceedings.neurips.cc/paper\\_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html](https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html).
- Wang, D., Wang, Z., LI, P., Chen, Y. & Li, H. (2019). Dual Gaussian attenuation model of ultrasonic echo and its parameter estimation. *AIP Advances*, 9(5), 055213. doi : 10.1063/1.5095994.
- Wang, K., He, B. & Zhu, W.-P. (2021). TSTNN : Two-Stage Transformer Based Neural Network for Speech Enhancement in the Time Domain. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7098–7102. doi : 10.1109/ICASSP39728.2021.9413740.
- Wang, Q., Ma, Y., Zhao, K. & Tian, Y. (2022). A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*, 9(2), 187–212. doi : 10.1007/s40745-020-00253-5.
- Wanin, M. (2001). Évaluation non destructive de la qualité des matériaux (Partie 1). Repéré à <https://www.techniques-ingenieur.fr/base-documentaire/mesures-analyses-th1/cnd-applications-sectorielles-42587210/evaluation-non-destructive-de-la-qualite-des-materiaux-partie-1-m4130/>.
- Wei, L., Huang, Z.-y. & Que, P.-w. (2009). Sparse deconvolution method for improving the time-resolution of ultrasonic NDE signals. *NDT & E International*, 42(5), 430–434. doi : 10.1016/j.ndteint.2009.01.009.
- Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L. & Zhang, L. (2021, October). CvT : Introducing Convolutions to Vision Transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22-31.
- Wu, Z., Liu, Z., Lin, J., Lin, Y. & Han, S. [arXiv :2004.11886 [cs]]. (2020). Lite Transformer with Long-Short Range Attention. arXiv. Repéré le 2023-11-02 à <http://arxiv.org/abs/2004.11886>.
- Yuan, S., Li, P. & Wu, B. (2022). Towards Single-Component and Dual-Component Radar Emitter Signal Intra-Pulse Modulation Classification Based on Convolutional Neural Network and Transformer. *Remote Sensing*, 14(15), 3690. doi : 10.3390/rs14153690. Number : 15 Publisher : Multidisciplinary Digital Publishing Institute.
- Yuan, Y. (2022). *Deep Learning and Computer Vision for Industrial Applications : Cellular Microscopic Image Analysis and Ultrasound Nondestructive Testing*. (Ph.D., Illinois Institute of Technology, United States – Illinois). Repéré à ISBN : 9798438784081.



- Zhang, G.-M., Zhang, C.-Z. & Harvey, D. M. (2012). Sparse signal representation and its applications in ultrasonic NDE. *Ultrasonics*, 52(3), 351–363. doi : 10.1016/j.ultras.2011.10.001.
- Zhang, Q., Lu, H., Sak, H., Tripathi, A., McDermott, E., Koo, S. & Kumar, S. (2020). Transformer Transducer : A Streamable Speech Recognition Model with Transformer Encoders and RNN-T Loss. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7829–7833. doi : 10.1109/ICASSP40776.2020.9053896.
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H. & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1), 43–76. doi : 10.1109/JPROC.2020.3004555. Conference Name : Proceedings of the IEEE.

