



HAL
open science

Optimisation des procédés de mise en forme par les réseaux de neurones artificiels

Abdessalem Chamekh

► **To cite this version:**

Abdessalem Chamekh. Optimisation des procédés de mise en forme par les réseaux de neurones artificiels. Mécanique [physics.med-ph]. Université d'Angers, 2008. Français. NNT: . tel-00445341

HAL Id: tel-00445341

<https://theses.hal.science/tel-00445341>

Submitted on 8 Jan 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Laboratoire de Génie Mécanique,
Ecole Nationale d'Ingénieurs de
Monastir, Tunisie

Laboratoire en Sécurité de fonctionnement, Qualité
et Organisation Institut des Sciences et Techniques
de l'Ingénieur d'Angers, France

THÈSE

En cotutelle¹

Pour l'obtention du grade de
DOCTEUR

Spécialité :
GÉNIE MÉCANIQUE

Présentée par :
CHAMEKH ABDESSALEM

OPTIMISATION DES PROCÉDÉS DE MISE EN FORME PAR LES RESEAUX DE NEURONES ARTIFICIELS

Soutenue le 16-02-2007
Devant la Commission d'Examen

JURY :

L. ROMDHANE	Professeur des Universités- E.N.I.Sousse	Président
O.DEBORDES	Professeur des Universités- E.C.Marseille	Rapporteur
Z. TOURKI	Maître de conférence- E.N.I.Sousse	Rapporteur
F. GUERIN	Professeur des Universités- I.S.T.I.Angers	Examineur
H. BELHADJ SALAH	Maître de conférence -E.N.I. Monastir	Directeur
R. HAMBLI	Professeur des Universités- Polytech'-Orléans	Directeur

¹ Cette thèse s'est déroulée dans le cadre d'une cotutelle entre l'École Nationale d'Ingénieurs de Monastir (TUNISIE) et l'Institut des Sciences et Techniques de l'Ingénieur d'Angers (FRANCE).

OPTIMISATION DES PROCÉDES DE MISE EN FORME PAR LES RESEAUX DE NEURONES ARTIFICIELS

RESUME :

Ce travail concerne la modélisation et l'optimisation des procédés de mise en forme par les Réseaux de Neurones Artificiels. Etant donnée que les méthodes classiques sont très coûteuses en temps de calcul et divergentes pour des problèmes présentant beaucoup des paramètres à contrôler, notre contribution consiste à développer une méthode de modélisation et d'optimisation plus rapide et efficace. Nous avons fait appel à une étude paramétrique pour coupler un programme basé sur la technique des RNA avec un code de calcul par la méthode des éléments finis. Néanmoins pour être viable, la méthode de modélisation et d'optimisation développée a été appliquée avec succès à des procédés de mise en forme complexes et variés. Elle a été testée en premier lieu pour la modélisation et l'optimisation du procédé d'emboutissage d'un flan circulaire et dans un second lieu pour l'identification des paramètres du matériau à partir du procédé d'hydroformage. La méthode a été aussi comparée avec une méthode d'optimisation classique de plusieurs points de vue. Il a été constaté le long de cette étude que notre démarche présente une grande potentialité à modéliser des relations qui sont difficile à les décrire avec des modèles mathématiques simple. Elle est aussi, rapide et parallélisable. La qualité des résultats obtenus est convaincante. Ce travail mène à des perspectives plus prometteuses. Elles peuvent être vulgarisées et exploitées dans d'autres applications.

MOTS CLES : Réseaux de Neurones Artificiels, Eléments Finis, Procédés de Mise en Forme, Optimisation, Identification.

OPTIMISATION OF SHEET METAL FORMING BY ARTIFICIAL NEURAL NETWORKS

ABSTRACT:

Sheet metal forming continues to be a significant industrial activity. The numerical simulation and the optimization strategy of these processes are extremely useful tools for the control of such technology. But these tools are based on mathematical models. On one hand, these models must take in account different constraints during deformation and take account of all the process parameters. On the other hand they must be simple, in order to facilitate their identification and their implementation in computer codes.

To cure at the long computing time that a finite element simulation or an optimization procedure can take and with the insufficiency of these mathematical models to take into account all the parameters in a sheet metal forming, alternatives are sought in other methods like responses surfaces, genetic algorithms, etc. In this work a method of modeling and optimization based on the ANN was proposed. The ANN method, judged very fast in computing times. It was exploited for modeling as well as for optimization. This procedure of modeling and optimization was tested in the first time for the modeling and the optimization of a deep drawing process of a cylindrical cup and in the second time for the identification of the materials parameters starting from the experimental bulge test. The quality of the results obtained is convincing. This work leads to more promising prospects. They can be popularized and exploited in other applications.

KEY WORDS: Artificiel Neural Networks, Finite element, Sheet metal forming, Optimisation, Identification

Remerciements

Ce travail a été effectué en collaboration entre le Laboratoire de Génie Mécanique à l'Ecole Nationale d'Ingénieurs de Monastir (LGM-ENIM) et le Laboratoire en Sécurité de fonctionnement, QUALité et Organisation à l'Institut des Sciences et Techniques de l'Ingénieur d'Angers (LASQUO-ISTIA). Je tiens à remercier Monsieur le professeur **A. Dogui** le chef du laboratoire LGM, Monsieur le professeur **A. Kobi** le directeur du laboratoire LASQUO et Monsieur professeur **B. Dumon** le précédent directeur du même laboratoire de m'avoir accueilli au sein de ces deux laboratoires et de m'avoir permis d'effectuer cette recherche dans les bonnes conditions.

Mes remerciements les plus chaleureux se dirigent à Monsieur **H. Bel Hadj Salah**, Maître de conférences à l'ENIM, pour avoir accepté de diriger ce travail avec beaucoup de patience et de présence. Il a assuré l'encadrement technique et scientifique de mes travaux de thèse. Qu'il soit salué pour sa grande disponibilité, son suivi continu, ses aides utiles, ses discussions fructueuses et ses conseils constructifs et précieux afin d'assurer l'encadrement de tous mes travaux allant de mon sujet de fin d'études d'ingénieurs. La qualité de son encadrement m'a permis d'atteindre la finalité de mes études. Qu'il veuille bien trouver ici mes vifs remerciements et mes profonds respects.

Je voudrais également remercier et exprimer mon profond respect à Monsieur **R. Hambli**, professeur à l'Ecole Polytechnique d'Orléans pour la confiance qu'il m'a accordé, en m'accueillant dans le laboratoire LASQUO et en me confiant un sujet de recherche ambitieux dont il fut l'instigateur. Il a mis à ma disposition les moyens matériels nécessaires et il a assuré l'encadrement technique et scientifique de mes travaux de thèse. Je tiens à le remercier très vivement pour sa participation constante, son aide permanente et sa grande contribution scientifique enrichissante dans l'étude effectuée.

Je remercie et j'exprime mon profond respects à Monsieur **O. Debordes**, professeur à l'Ecole Centrale de Marseille et Monsieur **Z. Tourki**, Maître de conférences à l'Ecole Nationale d'Ingénieurs de Sousse pour avoir accepté d'être rapporteurs et de m'accorder une partie de leur temps précieux pour bien critiquer ce travail.

Mes plus vifs remerciements s'adressent à Monsieur **L. Romdhane**, professeur à l'Ecole Nationale d'Ingénieurs de Sousse pour m'avoir fait l'honneur d'accepter de présider ce jury.

Egalement j'exprime ma gratitude à Monsieur **F. Guerin**, professeur à l'Institut des Sciences et Techniques de l'Ingénieur d'Angers qui a accepté d'examiner et de faire partie du jury de cette thèse.

Je me permets d'adresser mes remerciements et mes reconnaissances à tous les membres de deux laboratoires (LGM-ENIM et LASQUO-ISTIA) et à tous ceux qui ont contribué de façon directe ou indirecte à la réalisation de ce travail particulièrement à

Monsieur **R. Bahloul**, Maître assistant à l'Institut Supérieur des Sciences Appliquées et de Technologie de Sousse. Je souhaiterais dire à tous ceux dont je n'ai pas cité les noms et qui ont contribué, de près ou de loin, à l'aboutissement de ce travail, que je suis grandement sensible à leurs gestes et gentillesse.

Enfin, je m'incline envers ceux à qui je dois l'essentiel, en particulier à ma mère pour son soutien continu et ses sacrifices, à mes frères, à mes sœurs et à ma fiancée pour ses encouragements incessants, la confiance et l'intérêt qu'ils ont accordé à mes études et à ma réussite.

Table des matières

Introduction générale	8
1 Généralités sur l'optimisation des procédés de mise en forme	11
1.1 Introduction	11
1.2 Généralités et historique	12
1.3 Procédés de mise en forme	13
1.3.1 Emboutissage	14
1.3.2 Hydroformage	18
1.4 Problème d'optimisation	21
1.5 Méthodes d'optimisation	23
1.5.1 Méthodes par calcul de gradient	23
1.5.1.1 Méthode de Newton	23
1.5.1.2 Méthode de Quasi-Newton	24
1.5.1.3 Méthode du gradient conjugué	25
1.5.1.4 Méthode de Levenberg-Marquardt	25
1.5.1.5 Méthode GCMMA	26
1.5.1.6 Méthodes combinées	28
1.5.2 Méthodes Stochastiques	28
1.5.2.1 Algorithmes Evolutionnaires (EA)	28
1.5.2.2 Plan d'expériences	30
1.5.2.3 Surface de réponse	31
1.5.2.4 Réseaux de neurones artificiels	32
1.6 Conclusion	33
2 Les Réseaux de Neurones Artificiels	34
2.1 Introduction et historique	34
2.2 L'intérêt de l'utilisation des réseaux de neurones artificiels	35
2.3 Notions de base sur les réseaux de neurones artificiels	36
2.3.1 Du Neurone biologique au neurone artificiel	36
2.3.2 Les réseaux de neurones	37
2.3.2.1 Réseau de neurones non bouclé	38
2.3.2.2 Réseau de neurones bouclé (ou récurrent)	39
2.4 Apprentissage des réseaux de neurones	39
2.4.1 L'apprentissage supervisé	40
2.4.2 L'apprentissage non supervisé	40
2.4.3 Calcul d'erreur	41

2.4.3.1	Notation et définitions	41
2.4.3.2	Fonctionnement d'un réseau	42
2.4.3.3	L'algorithme de rétro-propagation des erreurs	43
2.4.3.4	L'algorithme du gradient total	46
2.4.3.5	L'algorithme du gradient stochastique LMS	47
2.5	Principaux travaux dans le domaine de la mécanique	47
2.6	Conclusion	53
3	Développement d'un algorithme d'optimisation basé sur les RNA	55
3.1	Introduction	55
3.2	Etude paramétrique	56
3.3	Programme RNA	56
3.3.1	Structure	57
3.3.2	Normalisation des données	58
3.3.3	Ajustement du modèle par apprentissage	59
3.3.4	Exécution	61
3.3.5	Critères d'arrêt et considérations pratiques	61
3.3.6	Validation	63
3.3.6.1	Fonctions analytiques	63
3.3.6.2	Prédiction par attribut	65
3.4	Influence des paramètres du modèle RNA	66
3.4.1	Influence du nombre d'epoch	67
3.4.2	Influence de nombre de neurones cachés	68
3.4.3	Influence de coefficient de moment	69
3.5	Algorithme	70
3.6	Exemple de validation	71
3.6.1	Mise en situation	71
3.6.2	Modélisation par éléments finis	72
3.6.3	Modèle RNA	74
3.6.4	Exécution de l'algorithme	75
3.6.5	Résultats et interprétations	75
3.7	Conclusion	76
4	Application des Réseaux de Neurones Artificiels à la modélisation et l'optimisation du procédé d'emboutissage	77
4.1	Introduction	77
4.2	Opération d'emboutissage : Modélisation numérique	78
4.3	Prédiction de l'épaisseur minimale dans l'embouti	80
4.3.1	Choix des paramètres opératoires	80
4.3.2	Modèle éléments finis	81
4.3.3	Description générale de l'algorithme de calcul	83
4.3.4	Modèle RNA	84
4.3.5	Résultats et interprétations	85
4.4	Identification du matériau par RNA	87
4.4.1	Identification globale de tous les paramètres opératoires	88

4.4.2	Identification des coefficients d'anisotropie	89
4.4.3	Identification de la fonction d'écrouissage	91
4.5	Optimisation du procédé d'emboutissage cylindrique	92
4.5.1	Introduction	92
4.5.2	Problématique liée au phénomène du retour élastique	92
4.5.3	Modélisation numérique	92
4.5.4	Plan d'expériences	94
4.5.5	Modélisation neuronale	94
4.5.6	Influence du rayon de la matrice	96
4.5.7	Influence du rayon du poinçon	96
4.5.8	Influence du jeu	97
4.5.9	Optimisation du retour élastique	98
4.6	Conclusion	101
5	Identification des paramètres du matériau AISI 304 par RNA à partir du procédé d'hydroformage	102
5.1	Introduction	102
5.2	Essai d'hydroformage	103
5.2.1	Principe	103
5.2.2	Préparation des éprouvettes	104
5.2.3	Résultats expérimentaux	105
5.3	Analyse par éléments finis de l'essai	108
5.3.1	Modélisation et simulation	108
5.3.2	Analyse numérique de l'essai	112
5.4	Identification de la courbe d'écrouissage par une interprétation bi-axiale . . .	115
5.4.1	Hypothèses	115
5.4.2	Détermination de la courbe d'écrouissage	115
5.5	Identification par méthode inverse	118
5.5.1	Méthode basée sur les procédures d'optimisation classique	118
5.5.1.1	Stratégie d'identification	118
5.5.1.2	Résultats et interprétation	119
5.5.2	Méthode basée sur les RNA	121
5.5.2.1	Procédure d'identification	121
5.5.2.2	Choix des paramètres opératoires du procédé	123
5.5.2.3	Résultats et discussions	124
5.6	Synthèse et comparaison des deux méthodes	126
5.7	Conclusion	128
	Conclusions et Perspectives	129
	Références	132
	Liste de publications et de communications	139

Liste des tableaux

3.1	Dureté HRC en fonction des paramètres milieu et température de trempe . . .	66
4.1	Les valeurs considérées des paramètres	81
4.2	Les valeurs utilisées	82
4.3	Paramètres utilisés	93
4.4	Plan d'expériences du retour élastique en ($^{\circ}$) pour différents rayons de matrice et de poinçon ($J=0\text{mm}$)	94
4.5	Plan d'expériences du retour élastique en ($^{\circ}$) pour différents jeu et rayons de matrice $R_p=8\text{mm}$	94
4.6	Optimisation du retour élastique pour $J=0\text{mm}$	99
4.7	Optimisation du retour élastique pour $R_p=8\text{mm}$	100
5.1	Profil de la pression	109
5.2	Coefficients de Lankford de l'acier inox	109
5.3	Comparaison entre les valeurs identifiées et expérimentales	120
5.4	Comparaison des coefficients de Lankford identifiés et expérimentaux	120
5.5	Niveaux et valeurs des paramètres liés à l'écrouissage	123
5.6	Niveaux et valeurs des paramètres liés à l'anisotropie	123
5.7	Comparaison entre valeurs identifiées et valeurs expérimentales	124
5.8	Comparaison des coefficients de Lankford's	126
5.9	Comparaison des valeurs identifiées avec les valeurs expérimentales	127

Table des figures

1.1	Exemple d'un problème d'optimisation [Kle2003]	12
1.2	Procédé d'emboutissage	15
1.3	Problèmes rencontrés dans les procédés de mise en forme particulièrement par emboutissage	16
1.4	Optimisation de la forme initiale du flan	18
1.5	Exemples des pièces obtenues par hydroformage	18
1.6	Principe du procédé d'hydroformage des tubes	19
1.7	Principe d'hydroformage des flans	19
1.8	Principaux modes de défaillances (Flambage, Plissement et Rupture)	20
1.9	Problème direct et problème inverse	22
1.10	Calcul de l'écart entre la courbe numérique et la courbe expérimentale	23
1.11	Approximation convexe utilisant le GCMMA	27
1.12	GCMMA avec méthode à deux points	27
1.13	Organigramme de l'Algorithme Evolutionnaire	29
1.14	Plan Factoriel complet	31
2.1	Un neurone avec son arborisation dendritique	36
2.2	Mise en correspondance neurone biologique / neurone artificiel	37
2.3	a-Fonction de Heaviside, b-Fonction signe	38
2.4	a-Fonction linéaire à seuil, b-Fonction sigmoïde	38
2.5	Structure d'un réseau de neurones non bouclé.	39
2.6	Un réseau de neurones bouclé à deux entrées.	39
2.7	Le neurone artificiel de Mc Culloch et Pitts	43
2.8	Réseau filtre des lois de comportement [Per1999]	48
2.9	Modèle RNA permettant de déterminer la force d'indentation en fonction des paramètres d'entrées	49
2.10	Détermination du comportement du matériau à partir d'un modèle neuronal [Tom2004]	51
2.11	Comparaison entre méthode d'identification inverse classique et méthode d'identification se basant sur les réseaux de neurones artificiels [Per1999]	52
3.1	Calcul paramétré à l'aide d'ABAQUS	57
3.2	Interaction d'Abaqus avec python [Aba2001]	57
3.3	Structure d'un réseau MLP programmé	58
3.4	Boîte de construction de programme RNA	59
3.5	Illustration de la descente du gradient utilisé dans notre programme	60
3.6	Processus d'apprentissage	60

3.7	Illustration du phénomène de sur-apprentissage	62
3.8	Validation du modèle RNA avec des fonctions analytiques à une seule variable	64
3.9	Validation du modèle RNA avec des fonctions analytiques à plusieurs variables	65
3.10	Comparaison de la réponse simulée par PONN pour différents nombres d'epochs	67
3.11	Influence de nombre de neurones cachés sur la réponse	68
3.12	Influence de nombre de neurones cachés sur l'erreur quadratique moyenne . .	68
3.13	Effet du coefficient α sur la réponse	69
3.14	Effet du coefficient α sur l'erreur quadratique moyenne	70
3.15	Oganigramme de l'algorithme d'optimisation automatique	71
3.16	Modèle schématique de la frappe de la balle sur la raquette.	73
3.17	Différentes étapes de l'analyse	73
3.18	Réaction de la raquette en fonction du temps	74
3.19	Structure du modèle RNA conçu	74
3.20	La moyenne de l'erreur quadratique RMSE en fonction de nombre d'epochs .	76
4.1	Principe d'emboutissage : Géométrie de la maquette numérique	79
4.2	Géométrie des outillages	80
4.3	Maillage du flan	81
4.4	Distribution de l'épaisseur dans l'embouti en mm	82
4.5	Schéma général de l'algorithme de calcul	83
4.6	Structure du réseau de neurone employé	84
4.7	Erreur RMSE en fonction de nombre d'epochs	85
4.8	Variation de l'épaisseur minimale en fonction des points considérés	86
4.9	Epaisseur minimale calculée par le modèle EF en fonction de celle prédite par le modèle RNA	86
4.10	Stratégie d'identification par $(RNA)^{-1}$	88
4.11	Modèle RNA proposé	89
4.12	Résultats du premier cas d'identification	89
4.13	Modèle RNA proposé	90
4.14	Résultats de l'évolution de l'erreur quadratique moyenne au cours des itérations	90
4.15	Résultats du deuxième cas d'identification	90
4.16	Modèle RNA proposé	91
4.17	Résultats du troisième cas d'identification	91
4.18	Modèle numérique du procédé	93
4.19	Erreur quadratique au cours de la phase d'apprentissage	95
4.20	Evolution de l'erreur quadratique moyenne au cours des itérations	95
4.21	Influence du rayon de la matrice sur l'angle du retour élastique	96
4.22	Influence du rayon de poinçon sur l'angle du retour élastique	97
4.23	Influence du jeu sur le retour élastique	97
4.24	Surface de réponse obtenue dans le cas ou $J=0$ mm	98
4.25	Courbes de niveaux obtenues dans le cas ou $J=0$ mm	99
4.26	Evolution RNA du retour élastique en fonction des paramètres du procédé sous forme de surface de réponse	100
4.27	Evolution RNA du retour élastique en fonction des paramètres du procédé sous forme des courbes de niveau	101

5.1	Principe de l'essai de gonflement hydraulique	104
5.2	Découpage de la tôle en forme d'éprouvette	104
5.3	Dispositif de mise en place des grilles	105
5.4	Outil d'emboutissage des joncs de retenu	105
5.5	a- Matrice circulaire, b- Matrice elliptique	106
5.6	Cellule de gonflement hydraulique munie de tous les équipements	106
5.7	Pression en fonction de la hauteur au pôle : (a) Cas d'une matrice circulaire, (b) Cas d'une matrice elliptique	107
5.8	Profil de la pièce après mise en forme pour une pression de 160 bars : Cas d'une matrice circulaire	107
5.9	Epaisseur le long d'un méridien pour une pression de 160 bars : Cas d'une matrice circulaire	108
5.10	Modèle géométrique EF de l'essai : (a) Cas d'une matrice circulaire , (b) Cas d'une matrice elliptique	109
5.11	Courbe d'érouissage de l'acier inox	110
5.12	Représentation de la forme déformée : (a) Cas d'une matrice circulaire, (b) Cas d'une matrice elliptique	110
5.13	Isovaleurs de déplacement à l'increment final (P=24MPa) : (a) Cas d'une matrice circulaire, (b) Cas d'une matrice elliptique	111
5.14	Contrainte équivalente de Von Mises : (a) Cas d'une matrice circulaire, (b) Cas d'une matrice elliptique	111
5.15	Déformation plastique équivalente : (a) Cas d'une matrice circulaire, (b) Cas d'une matrice elliptique	111
5.16	Pression en fonction de la hauteur au pôle : (a) Cas d'une matrice circulaire, (b) Cas d'une matrice elliptique	112
5.17	Nœuds choisis sur la partie utile du flan.	113
5.18	Contraintes au pôle : A droite cas anisotrope et à gauche cas isotrope	113
5.19	Déformations plastiques au pôle : A droite cas anisotrope et à gauche cas isotrope	114
5.20	Rapport des déformations au pôle : A droite cas anisotrope et à gauche cas isotrope	114
5.21	Rapport des contraintes au pôle : A droite cas anisotrope et à gauche cas isotrope	115
5.22	Equilibre du tronçon de la calotte sphérique	116
5.23	Courbes d'érouissage issues de la traction simple et de l'essai equibiaxiale .	117
5.24	Comparaison entre les réponses pression-déplacement	117
5.25	Procédure d'identification	118
5.26	Pression en fonction de la hauteur au pôle : Matrice circulaire	120
5.27	Pression en fonction de la hauteur au pôle : Matrice elliptique	121
5.28	Procédure d'identification par RNA	122
5.29	Pression en fonction de la hauteur au pôle : Cas d'une matrice circulaire . .	125
5.30	Pression en fonction de la hauteur au pôle : cas d'une matrice Elliptique à 0°	125
5.31	Pression en fonction de la hauteur au pôle : Cas d'une matrice Elliptique à 45°	126

Introduction générale

La réactivité à la demande du marché nécessite un raccourcissement du temps de mise au point de gammes de fabrication de produits. Les délais très courts et les exigences en terme de qualité de produits imposent désormais que les procédés de fabrication soient bien maîtrisés et optimisés. Aujourd'hui, les améliorations proviendront essentiellement des gains qui seront réalisés sur la maîtrise des procédés de fabrication. Cela implique une modélisation réaliste, fiable et rapide de l'ensemble des phénomènes physiques qui caractérisent les opérations de formage tels que le comportement du matériau, l'usure des outillages, l'influence des paramètres de l'opération . . . Pour déterminer le meilleur processus de formage, la simulation numérique des procédés de mise en forme constitue un moyen d'investigation privilégié. En donnant accès à des grandeurs difficilement accessibles par l'expérience, elle améliore la compréhension des phénomènes simulés. De plus, elle permet d'analyser un procédé, avant qu'il ne soit réellement mis en œuvre. Elle permet d'estimer, en fonctions de différents paramètres si une pièce est ou n'est pas mise en forme (déchirure, formation de plis, formation des cornes, . . .). Les idées nouvelles peuvent ainsi être testées, sans mobiliser l'outil de production.

Mais la simulation numérique se base sur des modèles mathématiques et des lois de comportement qui doivent satisfaire aux principes généraux de la mécanique et qui soient physiquement acceptables. Ces modèles mathématiques doivent d'une part, tenir en compte des sollicitations que subissent les matériaux au cours de leur déformation et rendent compte de tous les paramètres intervenant dans un procédé et d'autre part, ils doivent être simples, afin de faciliter leur identification et leur implémentation dans les codes de calcul. En effet, la simulation numérique d'un procédé de mise en forme est un processus complexe durant lequel un grand nombre de phénomènes physiques interviennent. La prise en compte de tous ces phénomènes, même de manière simple, exige un temps de calcul trop long.

Pour remédier d'une part, au temps de calcul long qu'une simulation numérique ou une procédure d'optimisation peut prendre et d'autre part, à l'insuffisance des modèles mathématiques, des outils basés sur l'intelligence artificielle sont en cours de développement. Le but de ce travail est de développer une méthode, de modélisation et d'optimisation de procédés de mise en forme de métaux basée sur les Réseaux de Neurones Artificiels couplée à un code de calcul par éléments finis. En effet, les Réseaux de Neurones Artificiels ont émergé comme une nouvelle branche de calcul, qui permet de conduire à la résolution des problèmes rencontrés dans la modélisation et l'optimisation de plusieurs processus. Ils ont montré une remarquable performance pour la modélisation des relations linéaires et non linéaires complexes. Cet outil mathématique est particulièrement utile pour la simulation de n'importe quelle corrélation qui est difficile à décrire avec des modèles physiques et mathématiques en raison de la capacité d'apprendre par des exemples. Il a été développé pour résoudre des problèmes en automatique et il sera transféré dans notre cas pour résoudre des problèmes en modélisation et en optimisation des procédés de mise en forme.

Dans ce travail nous proposons donc, de définir et d'utiliser les réseaux de neurones artificiels couplés au code de calcul par éléments finis Abaqus. Le desir est de construire un outil de modélisation et d'optimisation "rapide et efficace". Il sera utilisé pour substituer un calcul éléments finis (problèmes directs) et/ou pour recalculer des paramètres d'entrée (problèmes inverses). Il permet donc d'optimiser les procédés de mise en forme dans un temps très court. En définitive, la contribution du présent travail est de donner des solutions à des problèmes liés à :

1. La modélisation des problèmes complexes en mise en forme,
2. L'optimisation de procédés de mise en forme.

Ce travail fait, l'objet de cinq chapitres :

Le premier chapitre présente des généralités sur l'optimisation des procédés de mise en forme. Une première partie a été consacrée à une étude bibliographique sur les deux procédés de mise en forme par emboutissage et par hydroformage. On s'intéresse à la description de ces différents procédés et aux problèmes rencontrés lors de la mise en forme des pièces. Dans une deuxième partie, une analyse bibliographique a été menée sur les différentes méthodes d'optimisation qui ont été appliquées dans ce domaine. Une attention particulière a été accordée lors de cette étude, pour situer les méthodes d'optimisation stochastiques par rapport aux méthodes d'optimisation classiques. Nous concluons sur les objectifs et les techniques utilisées dans notre travail.

Le deuxième chapitre est dédié à la présentation de la théorie des réseaux de neurones artificiels. Après avoir exposé un aperçu historique sur les RNA, nous détaillons les différents éléments de base de cet outil mathématique. Nous limitons notre présentation à des aspects qui seront utiles pour la suite de ce travail. Une synthèse sur les principaux travaux utilisant les réseaux de neurones artificiels dans le domaine de la mécanique sera présentée à la fin de ce chapitre.

Le troisième chapitre est consacré à la partie développement. Un algorithme d'optimisation baptisé PONN (Process Optimisation Neural Networks) a été présenté. L'accent est mis sur les différentes parties constituant cet algorithme. La première partie a été consacrée à la présentation de l'étude paramétrique utilisant le code de calcul par éléments finis ABAQUS. La deuxième partie est dédiée au programme de réseaux de neurones, qui constitue le cœur de cet algorithme. La dernière partie a été consacrée à la validation et au couplage EF-RNA.

Le quatrième chapitre, traite un exemple d'application de cet outil pour la modélisation et l'optimisation d'une opération d'emboutissage. Pour ce faire, une base de données numérique a été établie par éléments finis en vue de construire un modèle RNA qui permet de calculer l'épaisseur minimale dans un embouti cylindrique. Cette base de données a été définie par la variation des paramètres intrinsèques au matériau et le paramètre technologique lié au frottement entre outils. Une proposition d'identification inverse des coefficients d'anisotropie pour un modèle élastoplastique selon le critère quadratique de Hill est présentée. La même démarche est utilisée pour la prédiction du retour élastique dans l'embouti. Mais dans ce

cas, un plan d'expériences, par une variation automatique des variables de conception a été construit. Dans bien des cas ce phénomène est associé aux propriétés du matériau cependant dans ce cas, nous faisons le contrôle de procédé en agissant sur les paramètres géométriques de celui-ci.

Le cinquième chapitre est dédié à l'application de cet algorithme neuronal comme stratégie d'identification des modèles de comportement anisotrope à partir des essais expérimentaux. Nous commençons ce chapitre par une présentation d'un essai d'hydroformage ainsi que les résultats expérimentaux. Etant donnée l'écart enregistré entre les réponses numérique et expérimentale, trois approches ont été proposées pour l'identification des paramètres de l'acier AISI304 : Analytique, inverse classique et neuronale. Une synthèse et une comparaison entre les deux dernières approches ont été illustrées à la fin de ce chapitre.

Chapitre 1

Généralités sur l'optimisation des procédés de mise en forme

1.1 Introduction

Les impératifs économiques, obligent les entreprises de réduire les coûts, de produire rapidement et de maîtriser la production. L'amélioration de la qualité technique des pièces mécaniques nécessite l'optimisation de leurs procédés de fabrication. La qualité des pièces fabriquées dépend d'un certain nombre de facteurs qui affectent leurs caractéristiques mécaniques et géométriques au cours de la fabrication. Pour améliorer cette qualité, des outils d'optimisation sont de plus en plus utilisés.

Dans le domaine de la mise en forme, la démarche la plus utilisée consiste à mettre en œuvre une approche exigeant la paramétrisation correcte du procédé concerné. L'objectif est de calculer les paramètres matériels ou géométriques optimaux, présents dans un modèle envisagé, menant à un écart minimum entre le résultat de l'expérience et celui de sa simulation. Ce problème d'identification paramétrique peut être formulé comme un problème d'optimisation. Pour assister les concepteurs dans la démarche de mise au point de gammes de fabrication, on peut envisager l'application des techniques d'optimisation. Dans ce contexte, plusieurs méthodes et techniques sont utilisées depuis une dizaine d'années. Mais la plupart de ces méthodes nécessitent des centaines voire des milliers de calcul pour aboutir à l'optimum.

Ce chapitre permet une description de quelques méthodes d'optimisation classiquement utilisées, ainsi que des méthodes récentes basées sur l'intelligence artificielle telle que les algorithmes génétiques, les réseaux de neurones, les surfaces de réponses, . . . Des généralités sur les deux procédés de mise en forme, respectivement par emboutissage et hydroformage seront présentées. Ces deux procédés vont constituer le cadre des deux applications fondamentales que nous allons considérer.

1.2 Généralités et historique

Avant 1940, les méthodes numériques d'optimisation de fonction à plusieurs variables étaient peu connues. En effet, les premières tentatives de minimisation de fonctions sont attribuées à Cauchy, en 1847, qui a utilisé une méthode dite de la plus grande pente. Cette méthode a été largement modifiée par la suite. En 1959, Davidson [Kha2004] introduit la notion de "variables métriques", lui permettant de minimiser des fonctions à plusieurs variables en un temps record. C'est le début de la période où sont développés les outils applicables à des domaines très variés comme les sciences, l'ingénierie, les mathématiques, l'économie. . .

Ces techniques d'optimisation ont une grande importance dans les applications technologiques, entre autre la mécanique. Au stade de la modélisation et du calcul, l'optimisation est un outil essentiel qui permet d'améliorer les résultats d'une étude et d'aboutir à la meilleure solution possible. Fletcher, en 1975 [Fle1975] a défini l'optimisation comme étant la science qui détermine la "meilleure" solution d'un problème mathématique (figure 1.1).

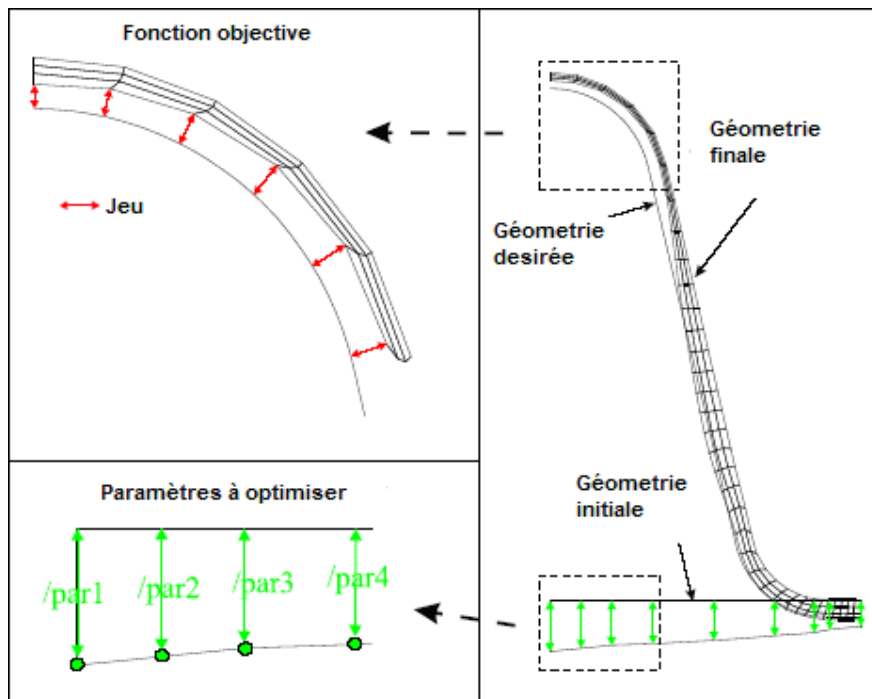


FIG. 1.1 – Exemple d'un problème d'optimisation [Kle2003]

Dans la bibliographie, différentes méthodes d'optimisations sont proposées. On distingue deux familles principales des méthodes d'optimisation : Les méthodes d'optimisation qui se basent sur le calcul du gradient de la fonction à minimiser et les méthodes stochastiques (n'utilisent pas le calcul de gradient). Ces dernières constituent, actuellement, un grand champs d'applications dans le domaine de la mécanique de structures.

Les méthodes du gradient sont classiques en optimisation, comprenant principalement les méthodes de la plus grande pente (ou méthode du gradient à pas optimal), du gradient conjugué, de Gauss-Newton [Gav1995][Gav1996] et de Levenberg-Marquardt [Gil1995], [Gou1998][Gou1999]. Les méthodes dites de Newton nécessitent la détermination de la matrice Jacobienne et la matrice Hessienne de la fonction à minimiser. La matrice Hessienne n'est pas toujours facile à inverser, pour cette raison, ils existent essentiellement deux méthodes dites de Quasi-Newton. La première est celle de BFGS, qui a été utilisée par Batoz et al. [Bat1998], par exemple pour optimiser les paramètres matériels dans un procédé de mise en forme industriel. La deuxième méthode dite DFP qui a été utilisée par Jin et al. [Jin2000], pour identifier les paramètres matériels d'alliages d'aluminium. Ces méthodes de Quasi-Newton sont des méthodes d'ordre 1 qui tentent, par des approximations successives de la matrice Hessienne, de retrouver la méthode de Newton.

Les méthodes stochastiques ou méthode d'ordre zéro (d'exploration directe), utilisent l'espace des paramètres par l'évaluation des valeurs de la fonction objectif au cours des itérations. On peut citer la méthode d'exploration au hasard de Monte-Carlo, les méthodes génétiques, la méthode du Simplexe non linéaire [Kha2004] et les Réseaux de Neurones Artificiels (RNA). Bien que cette dernière utilise l'une des méthodes d'optimisation à base de calcul de gradient pour l'ajustement des paramètres du modèle RNA. Toutes ces méthodes sont caractérisées par des schémas itératifs, exigeant une définition d'un point d'initialisation permettant à l'algorithme de progresser en engendrant une suite de points qui présentent des améliorations successives de la solution.

La méthode des réseaux de neurones artificiels, constitue une technique d'optimisation appartenant aux méthodes d'exploration directe. Elle est très efficace pour des cas où le nombre de variables est relativement important. Elle apparaît bien adaptée au type de problème que nous envisageons de résoudre. Une analyse détaillée de cette méthode sera présentée dans le deuxième chapitre.

En optimisation des procédés de mise en forme on distingue :

- Optimisation de forme : Les paramètres de conception sont des données géométriques du procédé ou du maillage (diamètre, longueur, épaisseur, coordonnées des points...),
- Identification matérielle : Les paramètres de conception sont des caractéristiques mécaniques de la tôle ou d'une façon générale les paramètres des lois de comportement.

1.3 Procédés de mise en forme

La mise en forme par déformation plastique des tôles métalliques continue à être une activité industrielle importante. A titre d'exemple, l'emboutissage représente de 15% à 20% du coût d'étude et de développement d'un nouveau véhicule. Une pièce hydroformée vaut -20% par rapport à une pièce obtenue par estampage et soudage [Sch1997]. Les fabricants qui ont cherché à former de la tôle mince se sont rapidement heurtés à des difficultés majeures. En effet, une réduction importante non prévue d'épaisseur peut engendrer une déchirure locale

de l'embouti, des rayons de congé faible des outillages peuvent provoquer une forte concentration de contrainte et une mauvaise utilisation des lubrifiants peut causer un frottement intensif entre les outils. Pour remédier à ces difficultés et éviter des ajustements ultérieurs des variables de conceptions, il est intéressant de rechercher la meilleure combinaison possible de ces variables dès la phase de conception de la gamme de fabrication en faisant appel à des outils d'optimisation et de simulation numérique.

En mise en forme le concepteur dispose d'un grand nombre de variables à régler, que l'on peut regrouper en trois familles :

- Les variables liées à la forme, telles que les rayons de raccordement, les nervures, la géométrie du flan, etc.
- Les variables liées au procédé, telles que le coefficient de frottement, le jeu entre les outils, etc.
- Les variables liées au matériau, telles que le coefficient d'anisotropie, le coefficient d'écroutissage, etc.

Il est donc, très difficile et coûteux de modifier l'une des ces variables et de voir son influence par des essais réels. C'est pourquoi il est très utile de faire appel à des essais virtuels. Ces derniers prennent de plus en plus de succès dans la phase de définition des gammes de fabrication. Des conférences internationales consacrées entièrement à ce sujet témoignent de l'intérêt scientifique de ce domaine.

Parmi les procédés de mise en forme on s'intéresse particulièrement, aux deux procédés suivants : l'emboutissage et l'hydroformage. Ils seront présentés dans les deux prochains paragraphes.

1.3.1 Emboutissage

Emboutir : " Travailler une plaque de métal au marteau et sur une enclume, de manière à la rendre concave d'un coté et convexe de l'autre " [Mer1998].

Par cette définition, nous sommes à même d'imaginer qu'il y'a fort longtemps qu'on emboutit des tôles. L'emboutissage consiste alors à fabriquer, à partir d'un flan plan de faible épaisseur, une pièce de forme complexe généralement non développable (Figure 1.2). Le flan étant plaqué contre la matrice avec une certaine force par l'intermédiaire d'un serre-flan, et mis en forme par avancée du poinçon à l'aide d'une presse.

Le processus d'emboutissage se décompose de la manière suivante :

- Définition de la direction de l'emboutissage,
- Définition du contour et de la position du flan initial,
- Descente du serre-flan pour plaquer la tôle sur la matrice,
- Mise en forme de la tôle par avancée du poinçon,
- Découpage des habillages de l'embouti afin d'obtenir la pièce finale.

Au cours de sa déformation, le métal est soumis simultanément à deux modes de sollicitations :

- Des déformations en expansion sur le nez du poinçon qui s'effectuent au détriment de l'épaisseur.
- Des déformations en retrait qui résultent d'un écoulement de matière sous serre-flan convergeant vers l'intérieur de la matrice et auxquelles s'associe un champ de contraintes compressif dans le plan de la tôle.

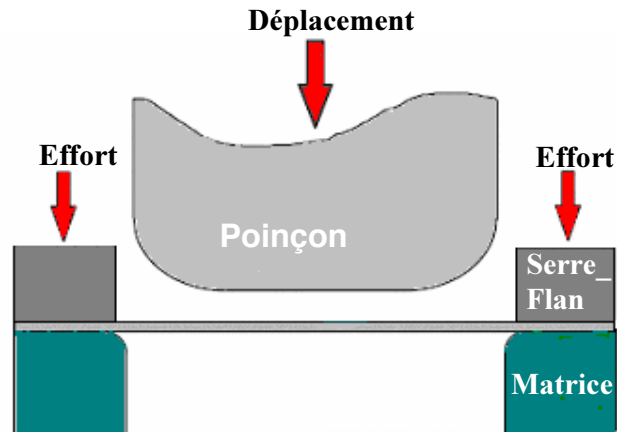


FIG. 1.2 – Procédé d'emboutissage

Tout l'art de l'emboutissage consiste en fait à réaliser le meilleur compromis possible entre les déformations des deux types en jouant sur les divers paramètres qui contrôlent l'écoulement du métal dans l'outil.

La réalisation d'un emboutissage s'avère donc complexe, et sa réussite est souvent liée à l'expérience. En effet, l'emboutissage se définit par une transformation permanente d'une feuille de métal en une forme tridimensionnelle sous l'effet des efforts appliqués par les outillages. Ce qui permet de mettre en exergue les aspects suivants :

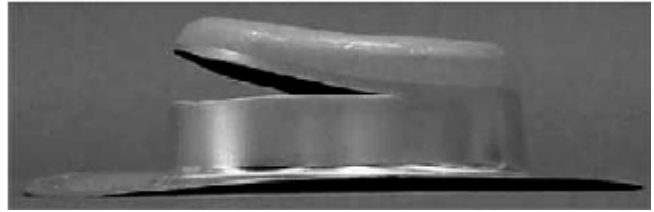
- Les grandes transformations géométriques d'une structure mince.
- Le comportement élastoplastique anisotrope du matériau.
- Les contacts et les frottements entre les outils et la tôle.

Les principaux paramètres intervenant dans ce procédé peuvent être résumés comme suit :

- L'épaisseur initiale de la tôle,
- La loi de comportement du matériau,
- La position du flan initial,
- le dimensionnement du contour initial,
- La force de serrage, les coefficients du frottement ainsi que la forme et la position de joncs.



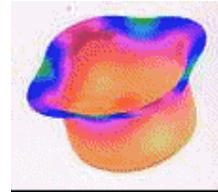
a) Plissement



b) Déchirure



c) Retour élastique



d) Cornes d'emboutissage

FIG. 1.3 – Problèmes rencontrés dans les procédés de mise en forme particulièrement par emboutissage

Les principaux problèmes rencontrés dans ce procédé sont [Hsu2002] :

- L'apparition des plis (figure 1.3a)
- Les déchirures dues à des conditions d'écouissage trop importantes (figure 1.3b)
- Le retour élastique (figure 1.3c)
- L'apparitions des cornes d'emboutissage (figure 1.3d)

Ainsi, de nombreux chercheurs ont désiré mieux connaître le processus afin de l'améliorer. En effet, il y'a eu le développement d'outils d'aide à la décision et d'autres outils d'analyse, de simulation et d'optimisation. Des codes de calcul par la méthode des éléments finis sont développés constituant une évolution technologique importante, et qui sont dédiés à la simulation numérique d'emboutissage.

L'intérêt de la simulation numérique de l'emboutissage par des essais virtuels apporte les avantages suivants :

- Elle peut substituer en partie l'expérimentation coûteuse en argent et en temps.
- Elle est utilisée en amont dans le processus de conception permettant, ainsi de réduire les délais de mise au point.
- Elle permet également de comparer entre différentes configurations afin d'optimiser la solution. Dans beaucoup des cas, cela permet une amélioration notable de la qualité de la pièce, un gain important de la matière et par conséquent une réduction du coût de produit.

Ainsi, dès le stade du bureau d'études et tout au long de la phase de conception, la simulation numérique se comporte comme un outil d'analyse et d'aide à la conception. Elle permet d'évaluer de manière qualitative les difficultés de faisabilité des pièces, par rapport au cahier de charge.

Nous pouvons classer les méthodes de simulation de l'emboutissage comme suit :

- Les outils de simulation du processus, qui par la connaissance du flan dans son état initial (forme, matériau), permettent de déterminer les caractéristiques des configurations intermédiaires et finales de l'objet (trajectoires des points matériels, répartition des déformations et épaisseur, contraintes résiduelles) en tenant compte des sollicitations et conditions géométriques imposées par la présence d'outils (poinçon, matrice serre-flan).
- Les outils d'analyse de l'emboutissabilité d'une pièce, qui à partir de la connaissance de la géométrie désirée, permettent d'étudier l'influence des différents paramètres pour définir :
 - La géométrie de flan initial (forme, contour),
 - Le process (conditions aux limites, forces de serrage, frottement, positions et formes de joncs),
 - Le matériau (limite élastique, écrouissage, anisotropie).

Donc, les méthodes de simulation numérique et d'optimisation de l'emboutissage doivent, dans la mesure du possible, prendre en compte différentes non linéarités (grandes transformations élastoplastiques, contact-frottement), qui combinées, forment un système fortement non linéaire.

A ce niveau, l'optimisation par simulation numérique des procédés est un outil qui permet de prévoir l'influence de la variation d'un paramètre donné au cours de la déformation de la matière. Il permet de tester et de dimensionner ces procédés dès la phase de conception du produit par recours à des essais expérimentaux et virtuels.

En effet, plusieurs travaux ont été menés sur l'optimisation du procédé en utilisant différentes méthodes d'optimisation. Ces travaux visent à déterminer les paramètres optimaux du matériau à géométrie connue ou de déterminer une géométrie optimale pour un matériau donné. Comme exemple, pour le premier cas, citons les travaux de Naceur et al. [Nac2004] qui portent sur le développement d'une approche inverse simplifiée combinée avec un algorithme de BFGS pour la recherche des paramètres matériels optimaux dans le cas d'une opération d'emboutissage. Cette approche inverse tient compte de la connaissance à priori de la forme d'objet et comporte une discrétisation de la surface 3D par des éléments triangulaires de la facette membrane/shell et le calcul du gradient inverse du tenseur de déformation pour estimer les contraintes. Comme exemple, pour le deuxième cas nous pouvons citer les travaux de Park et al. [Par1999] qui portent sur l'optimisation de la forme du flan initial par calcul d'erreur entre le flan obtenu numériquement et celui désiré (figure 1.4).

L'expression de l'erreur est donnée par : $Erreur = \sqrt{\frac{1}{N} \sum_i^N d_i^2}$

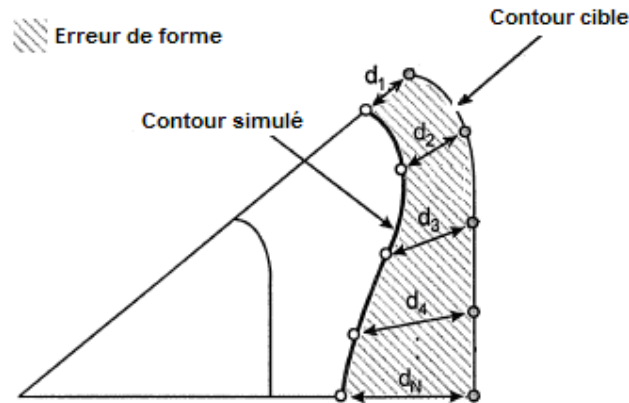


FIG. 1.4 – Optimisation de la forme initiale du flan

1.3.2 Hydroformage

L'hydroformage est un procédé de mise en forme, actuellement très utilisé puisqu'il présente plusieurs avantages par rapport à d'autres procédés. Cette technique de mise en forme utilise la pression d'un fluide pour mettre en forme la tôle. Donc par rapport au procédé d'emboutissage par exemple le liquide peut jouer le rôle du poinçon ou de la matrice. Cette technologie trouve, aujourd'hui, de plus en plus d'applications pour les pièces automobiles telles que les longerons, les collecteurs d'échappement, les berceaux support moteur, ... (figure 1.5).

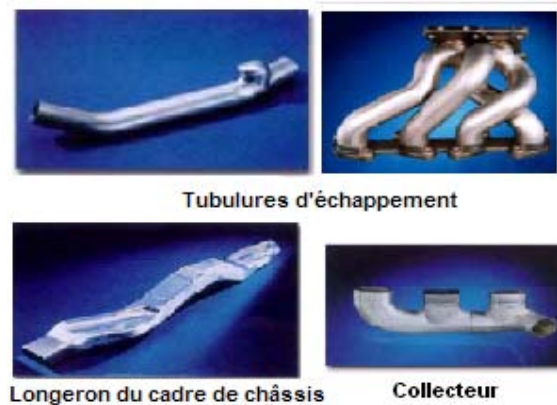


FIG. 1.5 – Exemples des pièces obtenues par hydroformage

On distingue principalement trois procédés d'hydroformage :

1. L'hydroformage des tubes : Un effort de compression généré par deux vérins, est généralement appliqué sur les deux extrémités du tube permettant ainsi un apport de matière

à la zone déformée (figure 1.6). Cet apport a pour effet de compenser l'amincissement du tube et d'améliorer sa formabilité [Ben2005].

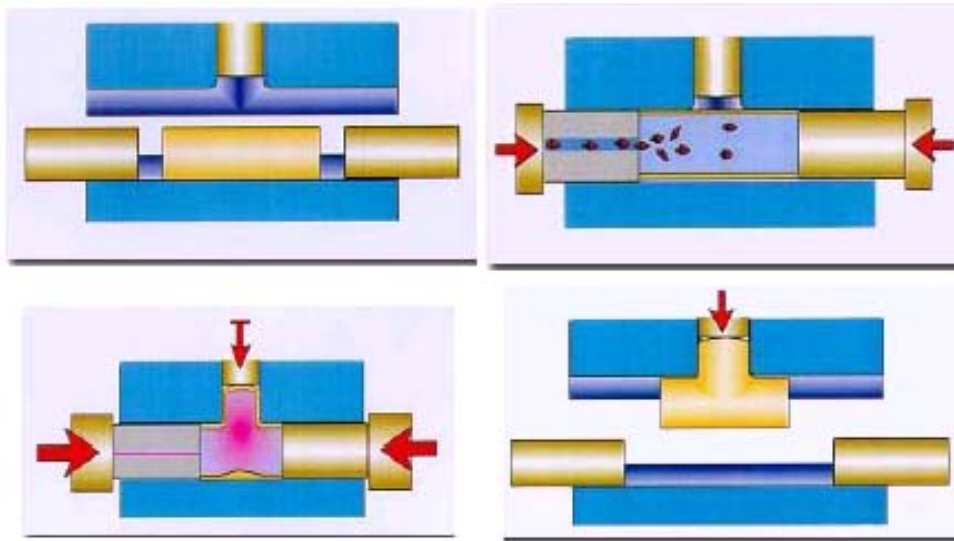


FIG. 1.6 – Principe du procédé d'hydroformage des tubes

2. L'hydroformage de flan simple : Il consiste à appliquer une pression sur une tôle pouvant se gonfler suivant la forme de la matrice. Ainsi la tôle est formée contre une matrice qui lui donne la forme finale [Ben2005]. La tôle peut être aussi formée sous l'action conjuguée d'un poinçon en déplacement et d'un liquide sous pression (figure 1.7).

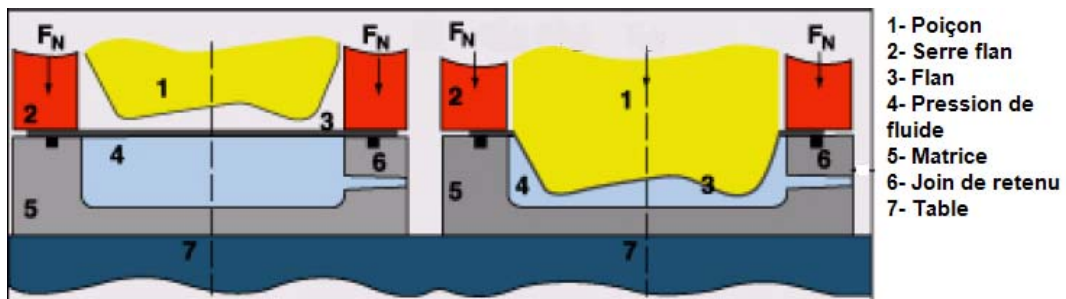


FIG. 1.7 – Principe d'hydroformage des flans

3. L'hydroformage de flans doubles : Au cours de ce procédé, deux plaques sont formées simultanément à l'intérieur d'une même matrice comportant une partie supérieure et une partie inférieure. Ce procédé a l'avantage de fournir deux pièces simultanément.

L'hydroformage offre plusieurs avantages [Ben2005] :

- La mise en forme des pièces de forme complexes avec une bonne reproductibilité et une précision dimensionnelle

- Plus léger (Moins de joints de soudure, Moins de pièces),
- Grande rigidité (utilisation de matériaux moins épais, écrouissage inhérent du procédé),
- Jusqu'à 27% d'économie par rapport au poids comparé à l'emboutissage (Coûts d'outils moins élevés, économie de production des pièces creuses qui nécessitent un plus grand rapport résistance/poids, les changements de matériaux ne requièrent pas d'outils neufs).
- Qualité de pièce améliorée. La distribution uniforme de la déformation peut amener :
 - Moins de retour élastique,
 - Moins de distorsion des surfaces,
 - Meilleure tolérance des dimensions extérieures,
 - Pas de frottement entre outils et tôle.

En dépit des avantages cités ci-dessus, l'hydroformage présente certaines limitations aussi bien d'ordre technologique qu'économique :

- Procédé lent et cadence de production peu élevée,
- Nécessite des matrices de haute qualité (polies),
 - Les forces de frottements augmentent avec la rugosité des matrices,
 - Les matrices rugueuses peuvent causer l'amincissement des parois et la rupture des pièces.

Les principaux problèmes dans ce procédé que les chercheurs tentent d'éviter ou les minimiser sont analogues aux problèmes rencontrés dans le procédé d'emboutissage. En effet, on peut citer (figure 1.8) :

- Striction
- Rupture
- Plissement

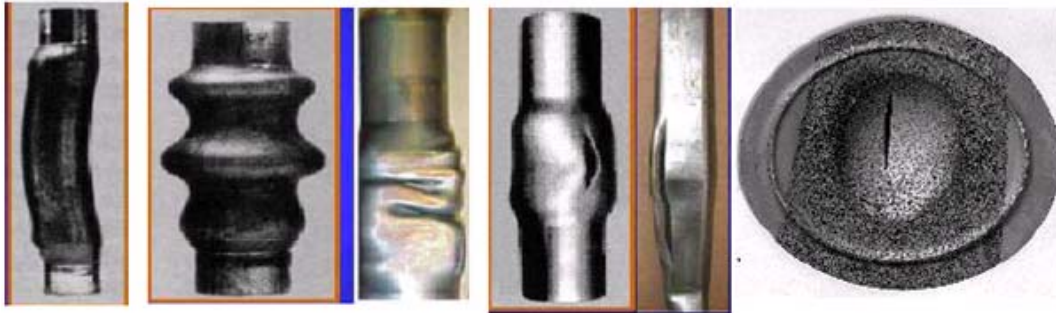


FIG. 1.8 – Principaux modes de défaillances (Flambage, Plissement et Rupture)

Les travaux réalisés dans le domaine de l'optimisation et du contrôle des procédés d'emboutissage et d'hydroformage sont basés sur une forte demande du secteur industriel, automobile en particulier, et correspondent aux avancées scientifiques actuelles dans le domaine de la simulation et du prototypage numérique.

L'optimisation des opérations d'emboutissage ou d'hydroformage consiste à rechercher, par voie de modélisation et de simulation numérique, des formes géométriques, des propriétés matérielles et des paramètres de procédés. Cela permet d'aboutir, après opération de mise en forme, à des composants de qualités bien déterminées, tant au niveau géométrique qu'au niveau propriétés matérielles ou encore intégrité des composants.

1.4 Problème d'optimisation

La simulation par éléments finis de processus de mise à forme offre de nombreux avantages [Pon2005]. Celle-ci permet de tester et comparer ces différents procédés rapidement et à faible coût. Cependant, comme toute technique de simulation, la méthode des éléments finis constitue une modélisation de la réalité, et nécessite donc un certain nombre de modèles approximant le comportement réel. De la qualité de ces modèles, dépend la fiabilité de la solution obtenue. Plus ceux-ci seront précis, meilleure sera la correspondance entre le résultat obtenu par simulation et la réalité [Kle2000].

Le développement de modèles de lois constitutives pour des comportements inélastiques se compose principalement de 4 étapes qui sont :

- La réalisation de tests expérimentaux,
- La formulation d'un modèle basé sur les connaissances physiques et sur les résultats expérimentaux obtenus,
- Le recalage des paramètres du modèle afin de retrouver les résultats expérimentaux (identification paramétrique),
- Enfin la vérification du modèle obtenu sur d'autres simulations.

Historiquement les méthodes d'optimisation reposent sur l'idée suivante : Transformer le problème de conception optimale en un problème de minimisation. Le procédé est défini en indiquant :

- Le matériau
- La géométrie des outillages,
- L'ensemble des paramètres du procédé : température, lubrifiants, géométrie des outils, etc.

La simulation numérique permet alors de procéder à une analyse qualitative : Déformations, écoulement, forces nécessaires, apparition de défauts, usure des outils, etc. A partir de ces résultats, on définit des critères d'optimisation. Si le procédé n'est pas satisfaisant (si l'un des critères d'optimisation n'est pas respecté), on modifie alors le procédé en changeant les valeurs de certains paramètres d'entrée. Pour chaque critère retenu la mesure de la "non-qualité" du procédé, est obtenue par son expression mathématique qui traduit une fonction coût. Le vecteur optimal des paramètres est alors solution du problème. La simulation numérique du procédé de mise en forme apparaît comme le point incertain des méthodes inverses. Le succès de ces méthodes repose en effet sur une estimation fiable et précise du problème direct. Ceci explique leurs apparitions tardives et leurs développements avec l'avènement des méthodes des éléments finis (figure 1.9). En dehors de l'optimisation de forme, ces méthodes ont connu un large succès en identification de paramètres [Kha2004].

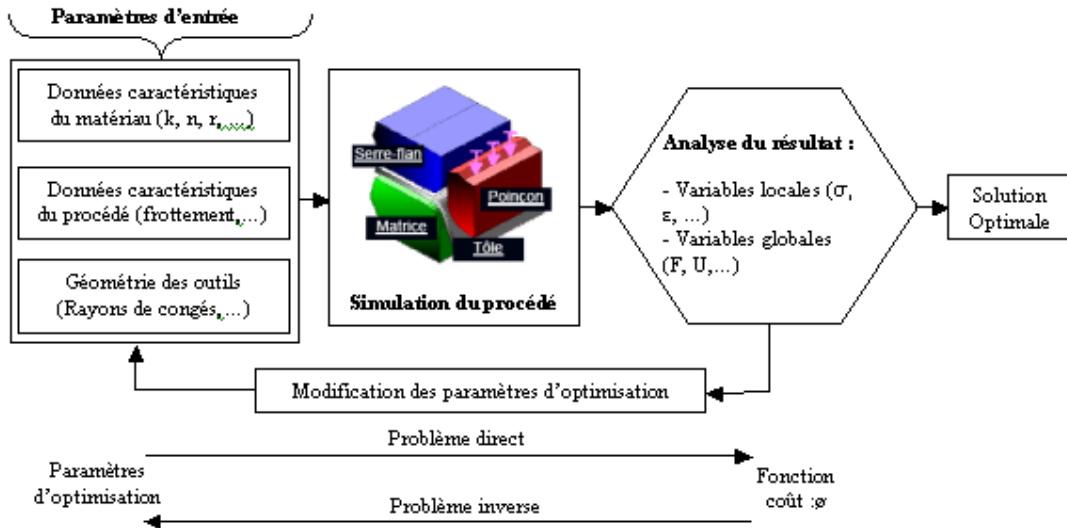


FIG. 1.9 – Problème direct et problème inverse

L'étape qui nous occupera plus dans les chapitres qui suivent, est l'identification des paramètres du modèle proposé. Celle-ci consiste à calculer la valeur de ces paramètres qui fournit le meilleur modèle possible, c'est-à-dire qui conduit à un écart minimum entre le résultat obtenu par l'expérience et celui fourni par sa simulation.

Il convient avant tout de se fixer une définition mathématique de l'écart entre la courbe expérimentale et le résultat de la simulation.

Les données expérimentales se présentent généralement sous la forme d'un échantillon de points où les mesures ont été réalisées (par exemple, le déplacement u d'un point pour différentes valeurs du temps $t = t_1, \dots, t_i, \dots, t_n$). Numériquement, on définit la distance entre les courbes expérimentale et simulée comme la somme des carrés des différences d'ordonnées entre chaque point expérimental et le point de même abscisse sur la courbe obtenue par simulation. Considérons le i^{ieme} point expérimental (figure 1.10), ainsi que le point de la courbe de simulation ayant la même abscisse, on peut ainsi calculer la différence entre les ordonnées de ces deux points. De plus, afin de pouvoir comparer entre eux des écarts provenant de problèmes différents ou du même problème mais avec un nombre de points expérimentaux différents, on définit la notion d'écart relatif entre les deux courbes.

$$F(x) = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{u_i^{EF}(x) - u_i^{\text{exp}}}{u_i^{\text{exp}}} \right)^2} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{\Delta u_i}{u_i^{\text{exp}}} \right)^2} \quad (1.1)$$

Où x est le vecteur des valeurs des paramètres matériels, n est le nombre de points expérimentaux, u_i^{exp} est l'ordonnée du i^{ieme} point obtenu expérimentalement, et u_i^{EF} est l'ordonnée du i^{ieme} point obtenu par simulation.

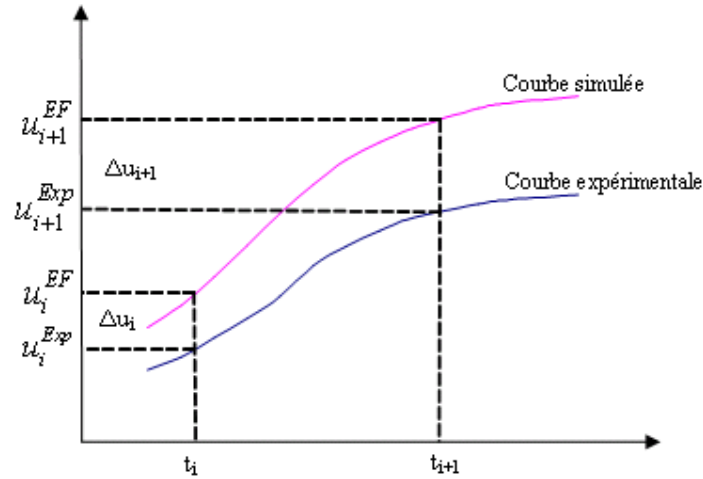


FIG. 1.10 – Calcul de l'écart entre la courbe numérique et la courbe expérimentale

On obtient ainsi la forme canonique du problème d'identification paramétrique envisagé. Trouvez $x^* = (x_1^*, x_2^*, \dots, x_p^*)^T$ tel que : $F(x)$ soit minimale.

Ce problème présente donc une forme classique d'optimisation. Ce type de formulation est largement utilisé dans la littérature. Le choix d'une méthode d'optimisation efficace permettant de minimiser cette fonction objectif est nettement plus complexe. On présente ci-dessous les techniques d'optimisation appropriées les plus utilisées dans la littérature, ainsi qu'une méthode originale issue des méthodes récentes dite méthode non paramétrée. Cette dernière est basée sur l'intelligence artificielle. Elle est peu utilisée en optimisation des structures. Nous avons implémenté celle-ci afin de la tester et de la comparer dans plusieurs applications numériques en termes d'efficacité, de précision, et de robustesse.

1.5 Méthodes d'optimisation

1.5.1 Méthodes par calcul de gradient

1.5.1.1 Méthode de Newton

La méthode de Newton est une méthode d'ordre 2 puisqu'il s'agit d'évaluer la matrice Hessienne au point considéré. Considérons le développement de Taylor au second ordre de la fonction objectif " $F(x)$ " autour du point x^k

$$F(x^k + \Delta x) = F(x^k) + \nabla F^T(x^k)\Delta x + \frac{1}{2}\Delta x^T \nabla^2(F(x^k))\Delta x = \tilde{F}(x) \quad (1.2)$$

Pour simplifier la notation, soit :

$$\Delta x = s$$

$$\nabla F^T(x^k) = (g^k)^T$$

$$\nabla^2(F(x^k)) = H^k$$

L'équation précédente (1.2) s'écrit alors sous la forme suivante :

$$F(x^k + s) = F(x^k) + (g^k)^T s + \frac{1}{2} s^T H^k s \quad (1.3)$$

Avec H^k est la matrice Hessienne. La direction de descente s est alors, choisie de façon à minimiser l'approximation précédente au point $x^k + s^k$. On obtient alors :

$$H^k s^k = -g^k \quad (1.4)$$

En supposant que la matrice Hessienne est définie positive, on aura ainsi :

$$(g^k)^T s^k = -(s^k)^T H^k s^k < 0 \quad (1.5)$$

On obtient alors l'actualisation de x ainsi :

$$x^{k+1} = x^k - (H^k)^{-1} g^k \quad (1.6)$$

Cette méthode est bien reconnue par son efficacité. Mais le principal inconvénient réside dans le calcul des dérivées secondes de F qui s'avère le plus souvent coûteux et très difficile à réaliser. Un certain nombre d'algorithmes se proposent ainsi de contourner cette difficulté en utilisant des approximations de la matrice Hessienne. On peut mentionner le cas particulier où F peut s'écrire sous forme de moindres carrés. On obtient alors une approximation de la matrice Hessienne en ne considérant que les produits des gradients. Ceci sera utilisé dans notre travail par la suite pour l'identification des paramètres du matériau à partir d'un essai d'hydroformage. L'objectif est d'être comparé avec une méthode stochastique. Cette approximation, qui est à la base des algorithmes de Gauss-Newton ou Levenberg-Marquardt, est largement utilisée en identification de paramètres rhéologiques. Cette méthode est bien adaptée surtout pour les problèmes de petites dimensions puisque le calcul de la matrice Hessienne est facile. Alors que si le problème présente un grand nombre de variables, il est généralement conseillé de coupler celle-ci avec la méthode du gradient conjugué ou une méthode de Quasi-Newton. Ou bien, lorsque l'amélioration relative de la fonction objectif devient trop faible, on passe automatiquement à la méthode du gradient conjugué.

1.5.1.2 Méthode de Quasi-Newton

Les méthodes de Quasi-Newton sont des méthodes d'ordre 1 qui tentent par des approximations successives à la méthode de Newton. Il s'agit de retrouver une approximation de la matrice Hessienne pour chaque itération sans calcul des dérivées seconde ni de l'inverse de cette dernière. Si on considère le développement en série de Taylor à l'ordre 1 du vecteur gradient défini dans le paragraphe précédent autour du point x^k :

$$g^{(k+1)} = g^{(k)} + H^{(k)}(x^{k+1} - x^k) \quad (1.7)$$

Si on définit les vecteurs suivants tel que :

$$y^k = g^{(k+1)} - g^{(k)} \quad s^{(k)} = x^{k+1} - x^k \quad (1.8)$$

Et on remplace ces deux dernières expressions dans (1.5) pour obtenir :

$$y^{(k)} = H^{(k)} s^{(k)} \quad (1.9)$$

Pour une nouvelle approximation des paramètres x^{k+1} telle que $g^{(k+1)}$ est nul, on déduit que :

$$y^k = -g^{(k)} \quad (1.10)$$

En remplaçant dans l'équation précédente il vient que :

$$s^{(k)} = (H^{(k)})^{-1} y^k = -(H^{(k)})^{-1} g^{(k)} \quad (1.11)$$

Cette dernière équation est identique à celle formulée dans le cas de la méthode de Newton décrite dans le paragraphe précédent. Le principe, donc des méthodes de Quasi-Newton est de remplacer l'inverse de la matrice Hessienne par une matrice par exemple $S^{(k)}$ qui sera actualiser pour chaque itération et que celle ci soit une approximation de l'inverse de la matrice Hessienne. Pour cela, on distingue deux techniques d'actualisation selon qu'on travaille avec celle proposés par Broyden, Fletcher, Goldfarb et Shanno nommée BFGS [Kle2000] ou bien par celle proposée par Davidon, Fletcher et Powell nommée DFP [Kle2000]. Le principe de deux techniques est basé sur une actualisation au cours des itérations de la forme :

$$S^{(k)} = S^{(k-1)} + \delta^{(k)} \quad (1.12)$$

En utilisant l'équation décrite en (1.11) :

$$S^{(k)} y^{k-1} = s^{(k-1)} \quad (1.13)$$

Il s'agit donc de calculer $S^{(k)}$ à partir de $S^{(k-1)}$ et de y^{k-1} , tout en respectant la condition exprimée dans (1.13) ci-dessus.

1.5.1.3 Méthode du gradient conjugué

Cette méthode fait partie des techniques classiques d'optimisation d'ordre 1. Le principe de celle-ci, est de calculer une direction le long de laquelle la valeur de la fonction objectif $F(x)$ diminue ou direction de descente, $F(k)$. Il s'agit ensuite de calculer le minimum de $F(x)$ le long de cette direction de descente. Cette étape, appelée méthode de recherche d'une ligne, est une minimisation à une dimension de la fonction objectif. On obtient ainsi un nouveau point où l'on recalcule une nouvelle direction de descente et ainsi de suite jusqu'à convergence.

1.5.1.4 Méthode de Levenberg-Marquardt

Cette méthode est très proche de la méthode de Newton décrite précédemment [Gel1995]. La seule différence réside dans l'introduction d'un paramètre λ , appelé paramètre de Levenberg-Marquardt, permettant de stabiliser la méthode de Newton. Ce paramètre est actualisé automatiquement en fonction de la convergence de chaque itération. Une stabilisation est possible grâce à un procédé réitératif (si une itération diverge, on la recommence au départ en augmentant le paramètre λ jusqu'à obtenir une itération convergente). Cependant, le phénomène de divergence forte lorsqu'on s'approche de l'optimum, inhérente à la méthode de Newton, n'est en rien supprimé ici. Tout au plus, la divergence peut être réduite. Il en résulte que, comme la méthode de Newton, il est conseillé de passer automatiquement à la méthode du gradient conjugué lorsque ce phénomène de divergence apparaît.

1.5.1.5 Méthode GCMMA

La méthode GCMMA (Globally Convergent Method of Moving Asymptotes) fait partie des méthodes appelées méthodes d'approximations convexes [Kle2003] dont le principe est de remplacer le problème non convexe implicite de base par une succession de problèmes explicites convexes simples. Ainsi, à chaque itération, une fonction objectif approchée est calculée (pas d'approximation des contraintes vu que celles-ci sont de type linéaire). Celle-ci est d'ordre 1, c'est-à-dire qu'elle respecte la valeur exacte de la fonction objectif au point de départ de l'itération en cours, ainsi que la valeur de la dérivée en ce point. Le problème simple ainsi formé est résolu aisément pour obtenir le minimum de cette fonction approchée. Ce point constituera la nouvelle approximation des paramètres optimaux recherchés et servira de point de départ pour une nouvelle itération et une nouvelle approximation de la fonction objectif.

Soit $F(x)$ la fonction objectif à minimiser, et x^k le point obtenu à l'itération k . Supposons $U(k)$ et $L(k)$, les asymptotes mobiles de l'itération k , calculées par la méthode d'actualisation choisie. Supposons, de même, connu le $\rho(k)$ correspondant à la fonction à approximer actualisée itération après itération. Après avoir calculé les dérivées premières de $F(x)$ par rapport à chaque x_i , on écrit l'approximation du premier ordre suivante :

$$\tilde{F}(x) = \sum_{i=1}^n \left(\frac{p_i^{(k)}}{U_i^{(k)} - x_i} + \frac{q_i^{(k)}}{x_i - L_i^{(k)}} \right) + r^{(k)} \quad (1.14)$$

avec :

$$p_i^{(k)} = (U_i^{(k)} - x_i^{(k)})^2 \left(\left(\frac{\partial F(x^{(k)})}{\partial x_i} \right)_+ + \left(\frac{\rho^{(k)}}{2} \right) (U_i^{(k)} - L_i^{(k)}) \right) \quad (1.15)$$

$$q_i^{(k)} = (x_i^{(k)} - L_i^{(k)})^2 \left(\left(\frac{\partial F(x^{(k)})}{\partial x_i} \right)_- + \left(\frac{\rho^{(k)}}{2} \right) (U_i^{(k)} - L_i^{(k)}) \right) \quad (1.16)$$

$$r^{(k)} = F(x^{(k)}) - \sum_{i=1}^n \left(\frac{p_i^{(k)}}{U_i^{(k)} - x_i^{(k)}} + \frac{q_i^{(k)}}{x_i^{(k)} - L_i^{(k)}} \right) \quad (1.17)$$

Où les symboles $()_+$ et $()_-$ signifient respectivement :

$$(y)_+ = \begin{cases} y & \text{si } y > 0 \\ 0 & \text{si } y \leq 0 \end{cases} \quad (1.18)$$

$$(y)_- = \begin{cases} -y & \text{si } y < 0 \\ 0 & \text{si } y \geq 0 \end{cases} \quad (1.19)$$

Si on prend le cas d'un problème à une seule variable. La figure 1.11 montre la fonction $F(x)$ et son approximation convexe au point A. $\tilde{F}(x)$ présente ainsi un minimum entre les deux asymptotes verticales U et L .

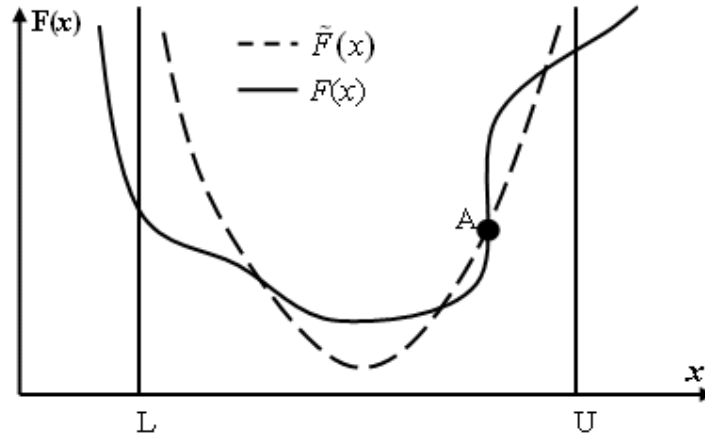


FIG. 1.11 – Approximation convexe utilisant le GCMMA

L'actualisation de ρ , de U et de L au fur et à mesure des itérations est un point capital de la méthode GCMMA. Svanberg [Kle2003] propose une loi d'actualisation de ρ qui semble, en pratique, efficace. Celle-ci est basée sur la différence entre la valeur de la fonction objective prise au point de l'itération en cours $F(x(k))$ et la valeur au même point de l'approximation convexe de l'itération précédente. Cette différence permet ainsi d'adapter la courbure de l'approximation réalisée.

De même, afin d'assurer la qualité des approximations réalisées, il est important d'actualiser efficacement les positions des asymptotes mobiles. Le principe de la "méthode à deux points" [Zha1994][Zha1997] est d'ajuster automatiquement les asymptotes, de façon à ce que l'approximation réalisée à l'itération k passe par le point $x(k-1)$ (figure 1.12).

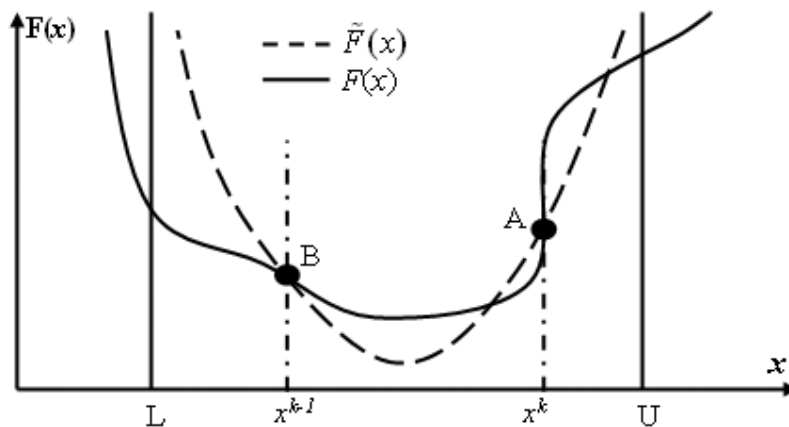


FIG. 1.12 – GCMMA avec méthode à deux points

Cette dernière méthode a été implémentée par Kleiner mann [Kle2000] en vue de résoudre des problèmes d'identification paramétrique. Celle-ci a donné, pour certaines itérations, une convergence très rapide. Par contre, d'autres itérations conduisaient à une divergence tout aussi importante. C'est pourquoi il a tenté de mettre au point des méthodes se basant sur les combinaisons des méthodes illustrées ci-dessus.

1.5.1.6 Méthodes combinées

Plusieurs combinaisons dans différentes applications, ayant pour objectif d'obtenir une méthode robuste (convergence dans la plupart des cas), précise (valeurs optimales des paramètres déterminées avec précision) et efficace (faible temps de calcul) ont été testées. Les huit méthodes d'optimisation suivantes ont été testées par Ponthot et al [Pon2006]. Les différentes applications numériques réalisées ont clairement mis en évidence la principale difficulté des problèmes d'identification paramétrique, à savoir obtenir une méthode d'optimisation robuste permettant d'assurer la convergence vers l'optimum dans le plus grand nombre de cas possibles. En effet, chaque problème présente une fonction objectif dont la forme peut être complètement différente du problème précédent. Ainsi, chaque méthode décrite ci-dessus fournit une bonne convergence pour certains problèmes, mais une très mauvaise, voire pas de convergence du tout dans d'autres cas. C'est pourquoi Ponthot et al en 2006 [Pon2006] ont proposé de combiner certaines de ces méthodes, avec passage automatique de l'une vers l'autre lorsque la première devient inefficace. Il en résulte de cette combinaison sept méthodes d'optimisation qui sont :

- Gradient Conjugué,
- BFGS,
- GCMMA,
- Newton + Gradient Conjugué
- Levenberg Marquardt + Gradient Conjugué
- Newton + GCMMA + Gradient Conjugué
- Levenberg-Marquardt + GCMMA + Gradient Conjugué

Ces dernières ont été comparées en terme de consommation de temps de robustesse et d'exactitude. En se basant sur des statistiques de 11 applications numériques. Les applications envisagées concernent l'identification et l'optimisation des procédés de mise en forme. Il s'avère que bien que la méthode de "Levenberg Marquardt + Gradient Conjugué" pourrait être une bonne alternative, sauf que la méthode de "Levenberg-Marquardt + GCMMA + Gradient Conjugué" qui est le plus adaptés aux problèmes inverses considérés, parce qu'elle montre la robustesse et l'exactitude mais elle n'est pas rapide. Cette méthode est très efficace mais raisonnablement coûteuse en chaque application numérique considérée [Pon2006].

1.5.2 Méthodes Stochastiques

1.5.2.1 Algorithmes Evolutionnaires (EA)

Les algorithmes génétiques, les stratégies d'évolution, et ce que l'on appelle maintenant en général les algorithmes évolutionnaires (figure 1.13), sont des techniques d'optimisation stochastiques inspirées de la théorie de l'évolution selon Darwin.

Pour un problème de minimisation J , un algorithme génétique simple sera exécuté de la manière suivante :

1. Génération aléatoire d'une population de N individus ;
2. Evaluation de la qualité du phénotype de chaque individu ;
3. Sélection d'un couple de parents au moyen d'une procédure appropriée ;
4. Croisement des deux individus avec une probabilité P_c pour générer deux enfants ;
5. Mutation des deux enfants avec une probabilité P_m ;
6. Répéter les étapes 3, 4, et 5 jusqu'à ce que la nouvelle population contienne N individus
7. Itérer à partir de l'étape (2) jusqu'à ce que l'algorithme converge.

Dans ce modèle, un processus de sélection retient les individus les mieux adaptés et les transmet à un processus d'exploration qui les utilisera pour créer de nouveaux individus. La figure 1.13 représente le schéma de fonctionnement d'un algorithme génétique simple. Après le processus de sélection intervient un processus de création qui s'inspire également de la génétique puisqu'il fait intervenir les notions de reproduction, de croisement et de mutation. Le croisement correspond à la création de deux enfants en combinant les gènes de deux parents choisis par le processus de sélection. Alors que la mutation revient à introduire une modification aléatoire dans le génotype d'un individu.

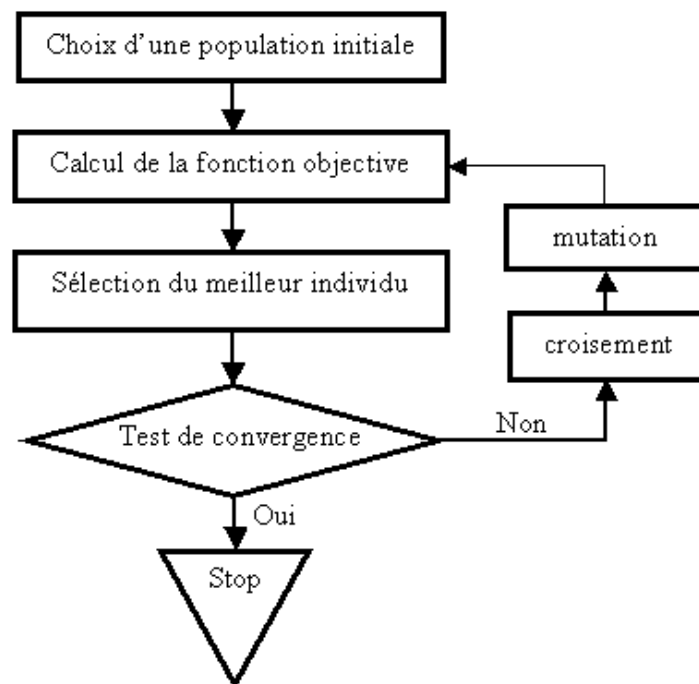


FIG. 1.13 – Organigramme de l'Algorithme Evolutionnaire

Ces méthodes sont maintenant très utilisées en optimisation numérique, lorsque les fonctions à optimiser sont complexes, irrégulières, mal connues ou en optimisation combinatoire. Ces méthodes diffèrent des méthodes illustrées ci-dessus (Levenberg-Marquardt, GCMMA, Gradient Conjugué, Newton...) par trois principaux aspects :

1. Elles ne nécessitent pas le calcul de gradient
2. Elles étudient une population dans un ensemble alors que les méthodes déterministes traitent un individu qui évoluera vers l'optimum.
3. Elles font intervenir des opérations aléatoires.

Elles ont été appliquées pour la résolution des plusieurs problèmes de type physique, mécanique,... Citons à titre d'exemples l'optimisation de structures mécaniques [Tro1993] [Ler1993] [Ham2002] de profils d'ailes d'avion, de tuyères de réacteurs ou de processus chimiques industriels. L'expérience prouve aussi que si les composantes ainsi que les paramètres de l'évolution sont soigneusement réglés, il est possible d'obtenir des algorithmes extrêmement efficaces et rapides [Bou2001]. Mais cette étape de réglage peut être très délicate. Donc, parmi les inconvénients de ces méthodes est le réglage des paramètres. En effet, des approches qui permettra de proposer des techniques d'ajustement de paramètres et aussi de comprendre plus finement quand et pourquoi un AE est efficace [Gol1989].

1.5.2.2 Plan d'expériences

A l'origine, les Plans d'Expériences (PE) s'appliquent à l'expérimentation (non numérique) et concernent la stratégie de recherche à suivre pour répondre à un certain nombre de questions; l'expérimentateur cherche comment sélectionner les expériences à faire, quelle est la meilleure stratégie pour :

- Conduire le plus rapidement possible aux résultats espérés,
- Eviter des expériences inutiles,
- Apporter une bonne précision,
- Conduire à la modélisation et à l'optimisation des phénomènes étudiés

Un plan d'expériences peut être utilisé comme une méthode d'optimisation, pour trouver une ou des solutions au problème posé, mais aussi comme une étape préliminaire à l'optimisation et a, alors, pour objectif le choix des variables à optimiser et des fonctions à prendre en compte dans une formulation mathématique classique pour résoudre le problème par une méthode de gradient ou par une méthode a exploration directe par exemple [Gou2001]. Notre intérêt s'est porté sur les plans d'expérience permettant de préparer une base de données en fonction des variables intervenant dans un procédé de mise en forme. On distingue essentiellement les plans d'expériences suivants :

- Plans de criblage : Pour trouver les facteurs les plus influents sur une réponse.
- Plans de modélisation = plans pour surfaces de réponse : modèles du 1^{ier} ou 2^{ieme} degré
- Plans de mélanges : Adaptés aux facteurs dépendants
- Plans complets/fractionnaires :
 - Plans factoriels complets : Toutes les combinaisons des niveaux de facteurs sont présentes

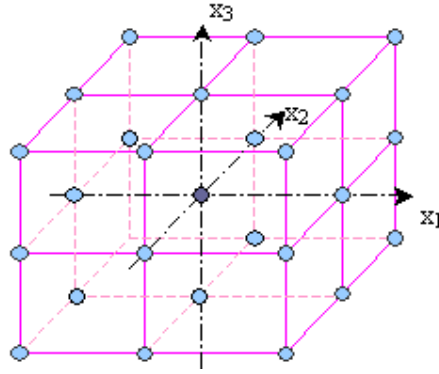


FIG. 1.14 – Plan Factoriel complet

- Plans factoriels fractionnaires : Tous les niveaux de chaque facteur sont présents, mais pas toutes les combinaisons possibles de facteurs

Les plans d'expériences ne font pas tous appel à des surfaces de réponse : les plus simples utilisent seulement 2 niveaux (réponse linéaire), et la notion de surface de réponse n'apparaît pas explicitement. Les approximations les plus utilisées sont des polynômes de degré 1 ou 2. Si un modèle est insuffisant (pas assez précis), il sera possible d'ajouter des termes au polynôme et des points au P.E. pour obtenir une meilleure approximation.

Nous intéressons plus au plan de type factoriel complet étant donné que nous le utilisons dans les chapitres qui suivent. Un plan factoriel complet, 3^n par exemple permet 3 niveaux par facteur mais il est très grand en nombre d'expériences ce qui alourdi son traitement. En contre partie, il permet une meilleure représentation du problème étudié. En effet, un balayage total de l'intervalle de conception est effectué. Ce type de plan trouve surtout son intérêt dans le cas d'un problème industriel. La figure 1.14 présente l'architecture de ce plan. Ce type des plans sera utilisé dans nos applications en amont d'un modèle basé sur les Réseaux de Neurones Artificiels pour la préparation des données. Parmi les travaux récents, qui témoignent sur l'intérêt de cet outil, on peut citer ceux de Roux et al. [Rou2002] qui ont utilisés les techniques de plan d'expériences dans le domaine d'optimisation des structures et ceux de Delamizière et al. en 2001 qui ont proposé une approche d'optimisation par plan d'expériences, des propriétés du matériau pour des pièces obtenues par emboutissage [Del2001].

1.5.2.3 Surface de réponse

Les surfaces de réponse sont issues des techniques de plans d'expériences. Elles consistent à construire une expression approchée f de la fonction coût explicite en fonction des variables de conception. Les deux méthodes (plan d'expériences et surfaces de réponse) sont utilisées pour la recherche de l'optimum d'un phénomène physique décrit par un modèle mathématique [Gou1999]. soit :

- Réponse : Grandeur à laquelle on s'intéresse (y)
- Facteurs : Variables qui peuvent être continues ou discrètes (x_1, x_2, \dots, x_n), qualitatives (pas de logique de classement) ou quantitatives.
- Niveaux : Valeurs prises par un facteur dans les expériences

La différence entre plan d'expérience et surface de réponse est que :

- Plan d'expériences : Liste des combinaisons de niveaux de facteurs pour lesquels la réponse du modèle doit être observée.
- Surface de réponse : Ensemble des réponses correspondant à tous les points du domaine, associée à un modèle mathématique f (modèle postulé, approximation) de la réponse :

$$y = f(x_1, x_2, \dots, x_n)$$
(avec 2 variables, la représentation géométrique du P.E. et de la surface de réponse associée est possible dans un espace de dimension 2 ou 3, sous forme de lignes ou de surface d'isovaleurs)

Cette méthode consiste, donc, à construire une expression approchée \tilde{f} de la fonction coût f explicite en fonction des variables de conception. Cette fonction notée \tilde{f} est réalisée à partir du calcul préalable de la fonction f pour un nombre limité d'évaluations choisis par l'utilisateur en fonction de la méthode d'approximation utilisée (splines, moindres carrés...). L'approximation \tilde{f} est ensuite minimisée, et non la fonction réelle. L'intérêt de cette démarche est d'utiliser, pour la recherche de minimum une méthode rapide à évaluer, puisque l'approximation est explicite. De plus il est possible de séparer l'étape de minimisation et l'étape de simulation du procédé, ce qui facilite le contrôle de la convergence des analyses non linéaires.

Plusieurs travaux en optimisation des procédés de mise en forme ont été réalisés en utilisant la méthode des surfaces de réponse avec des stratégies et d'approximation plus au moins différentes. Citons à titre d'exemple, l'optimisation des paramètres du procédé d'emboutissage par combinaison de l'approche inverse améliorée et la méthode de surface de réponse qui a été développé dans les travaux de thèse de ben Elechi [Ben2004]. Aussi, les travaux de Bahloul et al. [Bah2006-a] qui portent sur l'optimisation des paramètres du procédé de pliage par surface de réponse. Nous utilisons, les surfaces de réponse, par la suite, en conjonction avec les réseaux de neurone artificiels pour l'analyse des résultats en aval d'un modèle RNA.

1.5.2.4 Réseaux de neurones artificiels

L'histoire des réseaux de neurones artificiels appliqués aux sciences de l'ingénieur est récente. L'intérêt général pour cette thématique de recherche a été relancé en 1986, suite à la découverte de l'algorithme d'apprentissage par retropropagation du gradient d'erreur pour les réseaux multicouches. Cette avancée a permis d'envisager l'application des techniques connexionnistes à des problèmes réels (par exemple industriels).

Actuellement, l'intérêt de transférer des méthodes développées dans une discipline pour l'analyse des problèmes dans d'autres disciplines a évidemment augmenté. En effet, les réseaux de neurones qui ont été développés pour la résolution des problèmes en automatique

ont été, transférés, actuellement, pour la résolution des problèmes mécaniques. Cette technique est appliquée, en particulier, sur les procédés de mise en forme. Une étude détaillée sur le fondement de cette méthode ainsi qu'un aperçu sur les différents travaux dans le domaine de la mécanique seront exposés dans le deuxième chapitre.

1.6 Conclusion

Dans ce chapitre nous avons présenté des généralités sur l'optimisation des procédés de mise en forme. Nous avons commencé par un historique sur l'application des techniques d'optimisation en mise en forme. Puis parmi ces procédés, nous étions intéressés à l'emboutissage et l'hydroformage. Ces deux procédés vont constituer le champ d'application de notre algorithme d'optimisation par la suite. Une présentation succincte des deux procédés a été faite. Ensuite nous avons présenté les méthodes d'optimisation. Dans ce paragraphe nous avons distingué les méthodes d'optimisation classiques (qui utilise le calcul de gradient) et les méthodes stochastiques. Une étude bibliographique détaillée sur l'application de la méthode d'optimisation à base de RNA sera présentée dans le deuxième chapitre. Cette dernière constitue la base de notre algorithme d'optimisation.

Chapitre 2

Les Réseaux de Neurones Artificiels

2.1 Introduction et historique

En 1943, Mac Culloch et Pitts ont proposé le premier modèle de neurone biologique [Dav1993]. Ce dernier, appelé aussi neurone à seuil, a été inspiré des récentes découvertes en biologie. Ce sont des neurones logiques (0 ou 1). Ces deux physiciens ont montré que les neurones formels peuvent réaliser des fonctions logiques.

En 1949, le psychologue Donald Hebb a introduit le terme connexionisme pour parler des modèles massivement parallèles et connectés. Il a proposé de nombreuses règles de mise à jour des poids dont la plus connue à cette époque est la "règle de Hebb" [Dav1993].

En 1958, le psychologue Frank Rosenblatt a développé le modèle du perceptron [Ros1958]. Il s'agit d'un réseau de neurones, capable d'apprendre à différencier des formes simples et à calculer certaines fonctions logiques. Il est inspiré du système visuel. Il a réussi à l'appliquer pour la reconnaissance des formes. Au début des années 60, les travaux de Rosenblatt ont vécu un grand intérêt dans le milieu scientifique. Mais en 1969, deux scientifiques américains appelés Minsky et Papert [Dav1993] ont publié un livre dans lequel ont démontré les limites du perceptron proposé par Rosenblatt. En particulier, son incapacité réside à la résolution des problèmes non linéaires.

Les travaux se sont ralentis considérablement jusqu'aux années 80. En 1982, Hopfield a fini par démontrer l'intérêt des réseaux entièrement connectés [Dav1993]. Parallèlement, Werbos a conçu un mécanisme d'apprentissage pour les réseaux multicouches de type perceptron : la rétropropagation (Back-Propagation). Cet algorithme qui permet de propager l'erreur vers les couches cachées sera popularisé en 1986 dans un livre "Parallel Distributed Processing" par Rumelhart et al. [Rum1986].

Depuis ces travaux, les applications des réseaux de neurones n'ont cessé de croître. Il a d'ailleurs été démontré qu'un réseau MLP (Multi Layer perceptron) avec seulement deux couches peut approximer n'importe quelle fonction de R^n dans R^m avec une précision arbitraire [Wat1987].

Dans ce chapitre, une introduction sur l'intérêt d'utilisation des réseaux de neurones artificiels est décrite ainsi qu'en quoi cet outil mathématique peut nous en servir. On présente ensuite quelques notions de bases sur les réseaux de neurones. Puis, nous présentons la phase d'apprentissage d'un réseau de neurones. Nous intéresserons par la suite au réseau de type MLP et nous expliquerons la démarche de calcul des erreurs par la technique de retropropagation. Ce chapitre sera terminé par une présentation des diverses applications destinées à montrer la variété des domaines où les réseaux de neurones sont susceptibles d'apporter des solutions efficaces.

2.2 L'intérêt de l'utilisation des réseaux de neurones artificiels

Les réseaux de neurones formels, tels que nous les allons définir dans les paragraphes qui suivent, possèdent la propriété d'approximation qui peut être énoncée de la manière suivante :

Toute fonction bornée suffisamment régulière peut être approchée avec une précision arbitraire dans un domaine fini de l'espace de ses variables, par un réseau de neurones comportant une couche de neurones cachés en nombre fini, possédant tous la même fonction d'activation et un neurone de sortie linéaire [Hor1989].

En plus, les réseaux de neurones nécessitent moins de paramètres ajustables (les poids des connexions) que d'autres outils mathématiques couramment utilisés. Dans la pratique, on n'utilise pas les réseaux de neurones pour réaliser des approximations de fonctions connues. Le plus souvent, le problème qui se pose est de trouver une relation entre un ensemble de sorties dans un processus donné, et l'ensemble des entrées correspondant à travers les mesures effectuées. On suppose que cette relation existe malgré que :

- Les mesures sont en nombre fini,
- Elles sont certainement entachées de bruit,
- Toutes les variables qui déterminent le résultat du processus ne sont pas forcément mesurées.

En d'autres termes, l'ingénieur cherche un modèle du processus qu'il étudie à partir des mesures dont il dispose : on dit qu'il effectue une modélisation "boîte noire" [Per1999]. Dans la "syntaxe" des réseaux de neurones, les données à partir desquelles on cherche à construire le modèle s'appellent des exemples. Donc la raison pour laquelle les réseaux de neurones sont jugés d'être une bonne alternative est qu'on peut trouver une approximation à partir des mesures disponibles par un modèle RNA.

De manière générale, un réseau de neurones permet de faire un meilleur usage des mesures disponibles que les méthodes d'approximation non linéaires conventionnelles [Moz2000]. Ce gain peut être considérable lorsque le processus à modéliser dépend de plusieurs variables tel que dans le cas des procédés de mise en forme là où il intervient plusieurs types de non-linéarités et plusieurs paramètres matériels et technologiques.

2.3 Notions de base sur les réseaux de neurones artificiels

2.3.1 Du Neurone biologique au neurone artificiel

Un neurone biologique est une cellule vivante consacrée au traitement de l'information (figure 2.1). De son corps cellulaire ou soma, rayonnent de nombreuses dendrites (jusqu'à 100000) qui reçoivent des signaux provenant d'autres neurones ou cellules sensorielles. Ces signaux sont traités par le neurone qui transmet à son tour un signal, si certaines conditions sont réunies le long de son axone à d'autres neurones ou à de cellules effectrices (cellule musculaire par exemple) : On dit que le neurone est alors activé.

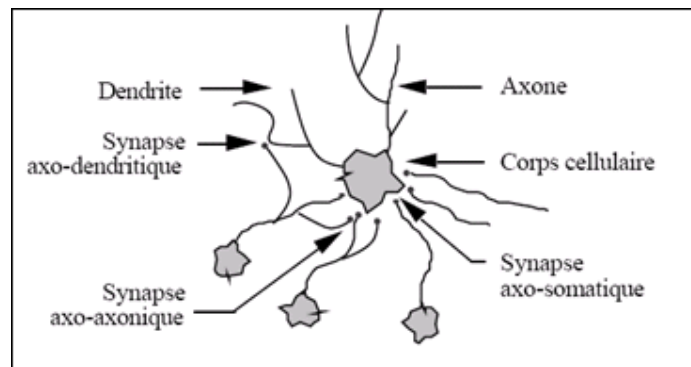


FIG. 2.1 – Un neurone avec son arborisation dendritique

La figure 2.2 montre la structure d'un neurone artificiel. Chaque neurone artificiel est un processeur élémentaire. Il reçoit un nombre variable d'entrées en provenance de neurones amont. A chacune de ces entrées est associée un poids w représentatif de la force de la connexion. Chaque processeur élémentaire est doté d'une sortie unique qui se ramifie ensuite pour alimenter un nombre variable de neurones aval. A chaque connexion est associé un poids.

Le modèle général du neurone artificiel est composé des éléments suivants :

- Une ou plusieurs entrées pondérées,
- Un sommateur,
- Une fonction de transfert,
- Une sortie.

Avec :

- x_i est l'entrée i ,
- w_{ij} est la valeur du poids synaptique reliant l'entrée i au neurone j ,
- Σ est la somme pondérée des entrées x_i ,
- $F()$ est la fonction de transfert,
- s est la sortie du neurone.

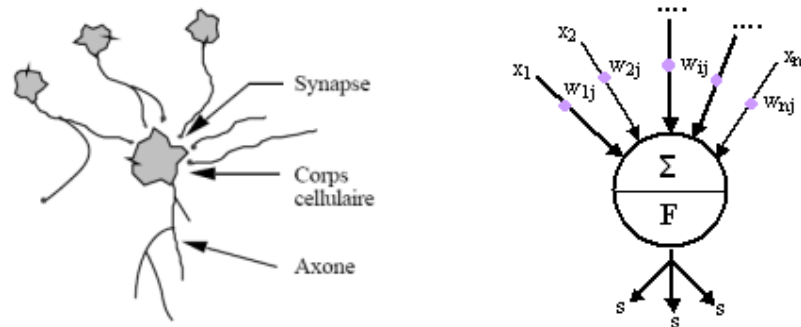


FIG. 2.2 – Mise en correspondance neurone biologique / neurone artificiel

Définition : Un neurone est une fonction algébrique non linéaire, paramétrée à valeurs bornées [Dre2002].

2.3.2 Les réseaux de neurones

Définition : Les réseaux de neurones artificiels (RNA) sont des processeurs élémentaires fortement connectés fonctionnant en parallèle. Chaque processeur élémentaire calcule une sortie unique sur la base des informations qu'il reçoit [Dre2002].

De point de vue modélisation mathématique, on peut définir un réseau de neurone artificiel par les quatre éléments suivants :

- La nature des entrées et des sorties : Elles peuvent être :
 - Binaire : (-1 ; +1) ou (0,1)
 - Réelles.
- La fonction d'entrée totale qui définit le prétraitement effectué sur les entrées : Elle peut être :
 - Booléenne
 - Linaire
 - Affine
 - Polynomiale de degré supérieur à deux
- La fonction d'activation du neurone qui définit son état en fonction de son entrée totale : Elle peut être :
 - Une fonction binaire à seuil ; dans ce cas on s'arrange pour que la forme de la fonction soit telle qu'on puisse utiliser la fonction de Heaviside (figure 2.3.a) ou la fonction signe (figure 2.3.b) :
 - Une fonction linéaire à seuil (figure 2.4.a) : Soit SATUR la fonction à seuil
 - Une fonction sigmoïde (figure 2.4.b).
- La fonction de sortie qui calcule la sortie du réseau en fonction de son état d'activation ; en général cette fonction est considérée comme la fonction identité.

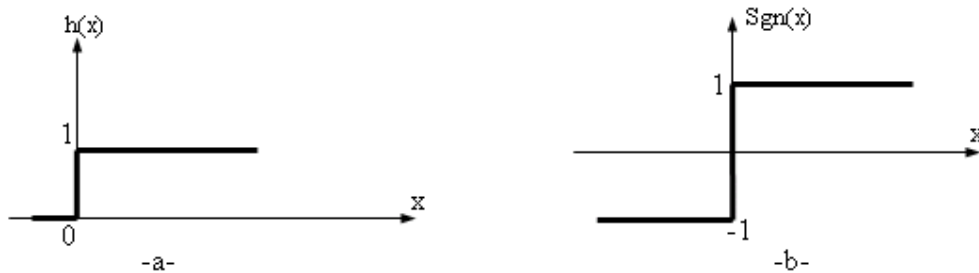


FIG. 2.3 – a-Fonction de Heaviside, b-Fonction signe

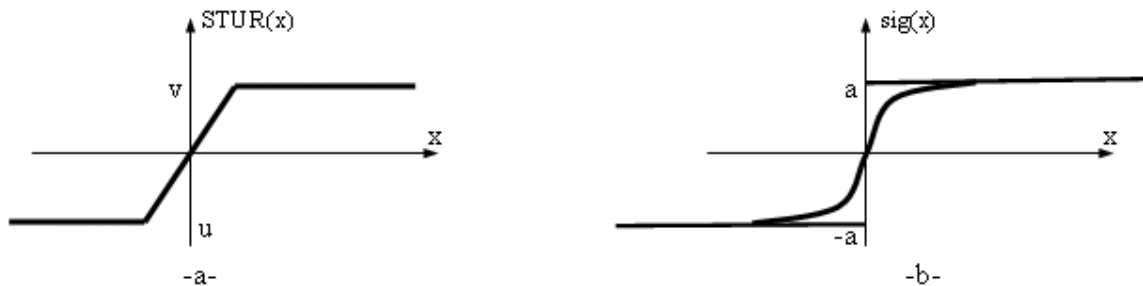


FIG. 2.4 – a-Fonction linéaire à seuil, b-Fonction sigmoïde

On distingue deux types de réseaux de neurones : Les réseaux non bouclés et les réseaux bouclés.

2.3.2.1 Réseau de neurones non bouclé

Un réseau de neurones non bouclé est donc représenté graphiquement par un ensemble de neurones connectés entre eux, l'information circulant des entrées vers les sorties sans "retour en arrière" [Bis1995]. On constate que le graphe d'un réseau non bouclé est acyclique. En effet, si on se déplace dans ce type de réseau à partir d'un neurone quelconque en suivant les connexions, on ne peut pas revenir au neurone de départ. Les neurones qui effectuent le dernier calcul de la composition de fonctions sont les neurones de sortie. Ceux qui effectuent des calculs intermédiaires sont les neurones cachés (figure 2.5).

Les réseaux non bouclés à couche sont structurés tel que les neurones qui appartiennent à une même couche ne soient pas connectés entre eux, chacune des couches recevant des signaux de la couche précédente, et transmettant le résultat de ses traitements à la couche suivante. Les deux couches extrêmes correspondent à la couche d'entrée qui reçoit ses entrées du milieu extérieur d'une part, et à la couche de sortie qui fournit le résultat des traitements effectués d'autre part. Les couches intermédiaires sont appelées couches cachées, leur nombre est variable.

Les réseaux de neurones non bouclés à couches dont les neurones cachés ont une fonction d'activation sigmoïde, sont souvent appelés des perceptrons multicouche (ou MLP pour Multi-Layer Perceptron) [Dre2002].

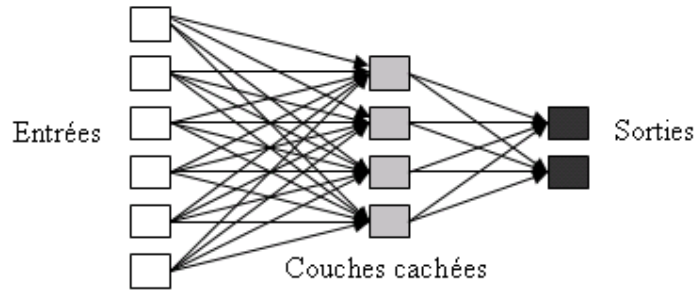


FIG. 2.5 – Structure d'un réseau de neurones non bouclé.

2.3.2.2 Réseau de neurones bouclé (ou récurrent)

Un réseau de neurones bouclé est schématisé par un graphe des connexions qui est cyclique. Lorsqu'on se déplace dans le réseau en suivant le sens des connexions, il est possible de trouver au moins un chemin qui revient à son point de départ (un tel chemin est désigné sous le terme de "cycle"). Ainsi, un retard entier multiple de l'unité de temps choisie est attaché à chaque connexion d'un réseau de neurones bouclé (ou à chaque arrête de son graphe). Une grandeur à un instant donné ne peut pas être fonction de sa propre valeur au même instant. Tout cycle du graphe du réseau doit avoir un retard non nul.

La figure 2.6 représente un exemple de réseau de neurones bouclé. Les chiffres dans les carrés indiquent le retard attaché à chaque connexion, exprimé en multiple de l'unité de temps. Ce réseau contient un cycle, qui part du neurone 3 et revient à celui-ci en passant par le neurone 4. La connexion de 4 vers 3 ayant un retard d'une unité de temps.

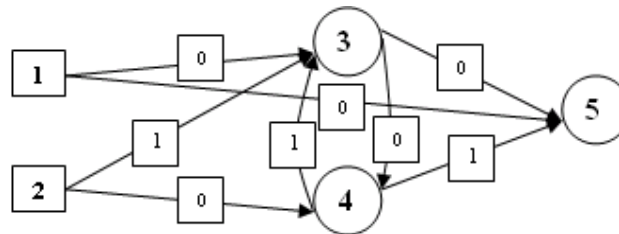


FIG. 2.6 – Un réseau de neurones bouclé à deux entrées.

2.4 Apprentissage des réseaux de neurones

On distingue deux types d'apprentissage : Un apprentissage "supervisé" et un apprentissage "non supervisé" [Ner1993].

2.4.1 L'apprentissage supervisé

L'apprentissage "supervisé" pour les réseaux de neurones formels, consiste à calculer les coefficients synaptiques de telle manière que les sorties du réseau soient, pour les exemples utilisés lors de l'apprentissage, aussi proches que possibles des sorties "désirées". Ils peuvent être la classe d'appartenance de :

- La forme que l'on veut classer,
- La valeur de la fonction que l'on veut approcher,
- La sortie du processus que l'on veut modéliser,
- La sortie souhaitée du processus à commander.

On connaît donc, en tout point ou seulement en quelques points les valeurs que doit avoir la sortie du réseau en fonction des entrées correspondantes : C'est en ce sens que l'apprentissage est "supervisé". Cela signifie qu'un "professeur" peut fournir au réseau des "exemples" de ce que celui-ci doit faire. Ce type d'apprentissage sera adopté par la suite dans notre algorithme d'optimisation.

La plupart des algorithmes d'apprentissage des réseaux de neurones formels sont des algorithmes d'optimisation : Ils cherchent à minimiser par des méthodes d'optimisation non linéaire (Premier chapitre) une fonction coût qui constitue une mesure de l'écart entre les réponses réelles du réseau et ses réponses désirées. Cette optimisation se fait de manière itérative, en modifiant les poids en fonction du gradient de la fonction coût : Le gradient est estimé par une méthode spécifique aux réseaux de neurones, dite méthode de rétropropagation, puis il est utilisé par l'algorithme d'optimisation proprement dit. Les poids sont initialisés aléatoirement avant l'apprentissage, puis modifiés itérativement jusqu'à obtention d'un compromis satisfaisant entre la précision de l'approximation sur l'ensemble d'apprentissage et la précision de l'approximation sur un ensemble de validation disjoint du précédent. Ce type d'apprentissage sera adopté dans notre programme RNA dans le chapitre suivant.

2.4.2 L'apprentissage non supervisé

Un réseau de neurones non bouclé peut être également utilisé dans un but de visualiser ou d'analyser des données : On dispose d'un ensemble de données, représentées par des vecteurs de grande dimension et l'on cherche à les regrouper selon des critères de ressemblance qui sont inconnus a priori. Ce type de tâches est connu en statistique sous le nom de méthodes "d'agrégation". On peut utiliser les réseaux de neurones non bouclés pour réaliser une tâche assez voisine : A partir des données décrites par des vecteurs de grande dimension, on cherche à trouver une représentation de ces données dans un espace de dimension beaucoup plus faible (typiquement de dimension 2) tout en conservant les "proximités" ou "ressemblances" entre ces derniers. Il n'y a pas là, donc de "professeur", puisque c'est au réseau de découvrir les ressemblances entre les éléments de la base de données, et de les traduire par une proximité dans la "carte" de dimension 2 qu'il doit produire.

2.4.3 Calcul d'erreur

2.4.3.1 Notation et définitions

Si l'on considère un réseau de n neurones numérotés de 0 à $n-1$ (soit $\nu = n-1$), on définit le vecteur colonne des états, le vecteur potentiels V et la matrice synaptique W . L'état du réseau est défini par le vecteur d'états. Le coefficient w_{ij} est la valeur du poids synaptique associé à la connexion qui lie la sortie du neurone j à une entrée du neurone i . La i^{ieme} ligne de la matrice W contient ainsi tous les coefficients synaptiques du neurone i . La matrice W est une matrice carrée $n.n$. Si le neurone i n'est pas connecté à la sortie du neurone j , le coefficient synaptique w_{ij} est maintenu à 0.

Il conviendra que toutes les entrées synaptiques x_j d'un neurone i sont connectées aux sorties de chaque neurone j du réseau, ainsi pour tout i , $x_j = \sigma_j$. Le réseau doit pouvoir être sensible à son environnement, il dispose pour cela de neurone d'entrée à entrée particulière dite externe. Une entrée externe n'est pas connectée à la sortie d'un autre neurone. Le réseau agit sur son environnement par l'intermédiaire d'un ensemble de neurones matérialisant ses sorties.

Dans le modèle neuronal que nous allons utiliser, l'entrée externe d'un neurone est non pondérée, elle s'ajoute à ses entrées synaptiques. Le potentiel de ces neurones est alors défini :

$$V_i(t + \tau) = \sum_{j=0}^{n-1} (W_{ij} \cdot \sigma_j(t)) + I_i(t) \quad (2.1)$$

Il devrait donc exister deux types de neurones dans un réseau : Ceux disposant d'une entrée externe et les autres, mais cela alourdirait les notations et les expressions analytiques pour peu de choses. Aussi, on considère sauf indication contraire que le potentiel de tout neurone est défini par l'équation (2.1). Lorsqu'un neurone ne dispose pas d'entrée externe, la variable $I_i(t)$ est toujours nulle.

Le fait d'utiliser des notations différentes pour distinguer les neurones de sortie des autres neurones, apporterait aussi des difficultés de notation. Pour cela, on définit un vecteur booléen de n composantes ω , chacune étant associée à un neurone, tel que : $\omega_i = 0$ si le neurone i n'est pas un neurone de sortie et $\omega_i = 1$ si le neurone i est un neurone de sortie.

Tout neurone d'un réseau peut en être une entrée ou une sortie du réseau sans qu'il y ait de contrainte. Un neurone qui n'est ni une entrée ni une sortie est un neurone dit caché du réseau.

Il est utile de reconnaître les synapses actives de celles qui ne le sont pas. Par exemple ils peuvent nous en servir pour la reconnaissance des coefficients synaptiques modifiables durant un apprentissage ou alors, d'un point de vue algorithmique pour éviter d'effectuer d'inutiles multiplications par 0 durant l'utilisation du réseau. Dans ce but, on associe généralement à un réseau de neurones un graphe d'adjacence dont les sommets représentent les neurones et

les arcs les synapses. Les arcs sont orientés de la sortie d'un neurone vers l'entrée d'un autre. On définit de même une matrice d'adjacence C , telle que C_{ij} est égale à 1 s'il existe une connexion du neurone pré-synaptique j vers le neurone post-synaptique i .

2.4.3.2 Fonctionnement d'un réseau

Un neurone artificiel fait une somme pondérée des potentiels d'actions qui lui parviennent (chacun de ces potentiels est une valeur numérique qui représente l'état du neurone qui l'a émis), puis s'active suivant la valeur de cette sommation pondérée. Si cette somme dépasse un certain seuil, le neurone est activé et transmet une réponse (sous forme de potentiel d'action) dont la valeur est celle de son activation. Si le neurone n'est pas activé, il ne transmet rien.

En 1943, W. S. Mc Culloch et W. E. Pitts ont proposé un modèle mathématique très simplifié du neurone biologique. Ce modèle, appelé neurone artificiel, est un automate capable de prendre deux états, 0 (inactif) ou 1 (actif). Il dispose d'une sortie correspondant à l'axone et transmettant son état à d'autres neurones artificiels et des entrées pondérées. Ils correspondent aux dendrites recevant des états qui proviennent d'autres neurones ou des grandeurs provenant des capteurs. Le nombre qui pondère chaque entrée est appelé "coefficient synaptique" (figure 2.7).

Soit W_{ij} le coefficient synaptique, ou poids synaptique, correspondant à la synapse transmettant l'activité du neurone j vers le neurone i . Soient $x_j(t)$ les valeurs présentées sur chacune des ν entrées du neurone i à l'instant t . Soit θ_i une quantité associée au neurone i appelé "seuil". On définit $V_i(t)$ le potentiel d'action du neurone i à l'instant t de la façon suivante [Tou1992] :

$$V_i(t) = \sum_{j=1}^{\nu} (W_{ij} \cdot x_j(t)) \quad (2.2)$$

L'état $\sigma_i(t + \tau)$ du neurone i à l'instant $t + \tau$ s'écrit :

$$\sigma_i(t + \tau) = 0 \text{ si } V_i(t) \leq \theta_i \quad (2.3)$$

$$\sigma_i(t + \tau) = 1 \text{ si } V_i(t) > \theta_i \quad (2.4)$$

La durée τ est caractéristique du neurone. On considèrera pour ce rapport que la durée est la même pour tous les neurones d'un réseau. L'état $\sigma_i(t + \tau)$ constitue le signal qui est transmis à d'autres neurones.

Il est commode de traiter le seuil θ_i comme un coefficient synaptique, la synapse correspondante étant considérée comme connectée à un neurone dont l'état est maintenu à -1. L'écriture des expressions analytiques s'en trouve souvent légèrement simplifiée. Ce neurone comporte donc $\nu + 1$ entrées. Supposons que le seuil correspond au coefficient synaptique W_{i0} , on aura :

$$W_{i0} = \theta_i \text{ et } x_0 = -1$$

On définit alors l'évolution de l'état d'un neurone par les expressions suivantes :

$$V_i(t) = \sum_{j=0}^{\nu} (W_{ij} \cdot x_j(t)) \quad (2.5)$$

$$\sigma_i(t + \tau) = f(V_i(t)) \quad (2.6)$$

Dans le cas des neurones de Mc Culloch et Pitts, la fonction f est l'échelon de Heaviside (figure 2.7).

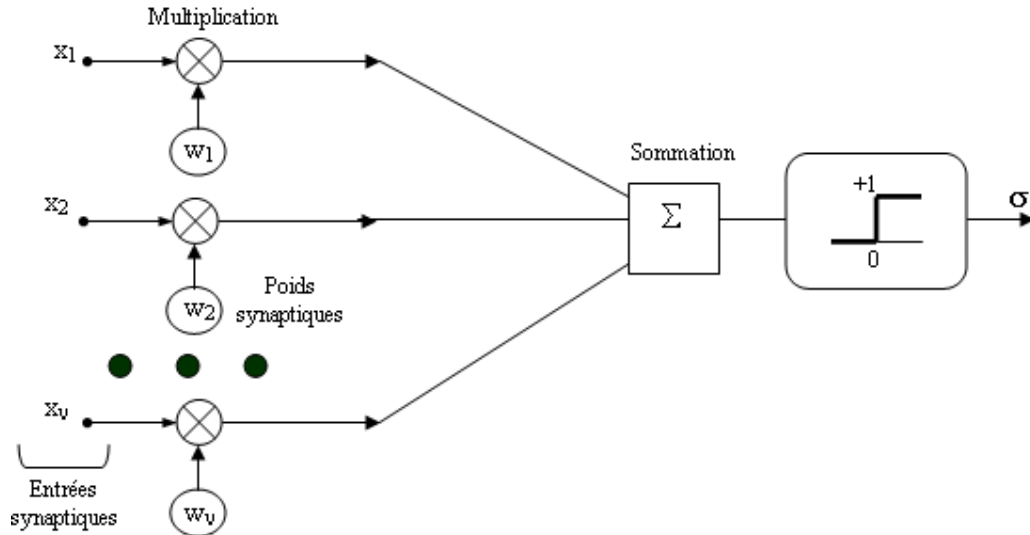


FIG. 2.7 – Le neurone artificiel de Mc Culloch et Pitts

2.4.3.3 L'algorithme de rétro-propagation des erreurs

Soit un exemple k qui est formé d'un couple de vecteurs $I(k)$ et $d(k)$ qui représentent respectivement l'état imposé à l'entrée du réseau et l'état de sortie désiré. Pour chaque neurone du réseau correspond une composante de $d(k)$ et $I(k)$. Si le neurone i n'est pas un neurone d'entrée, $I_i(k) = 0$, s'il n'est pas un neurone de sortie, la valeur de $d_i(k)$ n'est pas définie et la composante $\omega_i = 0$. Si le neurone i est un neurone de sortie $\omega_i = 1$.

La différence entre le vecteur $d(k)$ et le vecteur des états des neurones $\sigma(k)$ peut être exprimée par un coût quadratique $J(k)$:

$$J(k) = \sum_{i=0}^{n-1} \omega_i (d_i(k) - \sigma(k))^2 \quad (2.7)$$

Le coût quadratique doit être minimisé pour tout exemple k . Cela revient à minimiser le coût quadratique global $J = \sum J(k)$:

$$J(k) = \sum_{k=0}^{s-1} \sum_{i=0}^{n-1} \omega_i (d_i(k) - \sigma(k))^2 \quad (2.8)$$

Où s est le nombre d'exemples de l'ensemble d'apprentissage.

L'algorithme de rétro-propagation des erreurs [Rum1986] est une méthode d'évaluation du gradient dont les calculs ont été astucieusement organisés de façon à réduire leur complexité temporelle et spatiale. Pour toute structure neuronale non bouclée l'algorithme fonctionne de la façon ci-dessous.

A chaque itération d'un algorithme de gradient, le moyen le plus simple de réduire la valeur de J est d'ajouter à chaque poids W_{ij} une quantité ΔW_{ij} définie de la façon suivante :

$$\Delta W_{ij} = -\eta \frac{\partial J}{\partial W_{ij}} \quad \text{avec } \eta > 0 \quad (2.9)$$

Le coefficient η appelé pas de gradient doit être choisi suffisamment grand, pour réduire au mieux le nombre d'itérations, mais pas trop pour que l'algorithme reste stable.

En introduisant le potentiel $V_i(k)$ du neurone i pour un exemple k et en utilisant la formule des dérivations composées, on obtient :

$$\Delta W_{ij} = -\eta \sum_{k=0}^{s-1} \frac{\partial J(k)}{\partial V_i(k)} \frac{\partial V_i(k)}{\partial W_{ij}} \quad (2.10)$$

Les neurones du réseau sont numérotés tel que si l'état du neurone i dépend de celui du neurone j , alors $j < i$. Tous ces neurones peuvent disposer d'une entrée et/ou d'une sortie externe. En remplaçant la dépendance par rapport au temps dans les expressions (2.5) et (2.6) par l'indice k de l'exemple courant présenté au réseau, on obtient :

$$V_i(k) = \sum_{j=0}^{n-1} W_{ij} \sigma_j(k) + I_i(k) \quad (2.11)$$

$$\sigma_i(k) = f(V_i(k)) \quad (2.12)$$

on en déduit alors :

$$\frac{\partial V_i(k)}{\partial W_{ij}} = \sigma_j(k) \quad \text{pour } j < i \quad (2.13)$$

On pose :

$$\delta_i(k) = -\frac{\partial J(k)}{\partial V_i(k)} \quad (2.14)$$

Soit i , le numéro d'un neurone quelconque du réseau, soit $J_i(k)$ la quantité :

$$J_i(k) = \sum_{m=i}^{n-1} \omega_m (d_m(k) - \sigma_m(k))^2 \quad (2.15)$$

On a :

$$J_i(k) = \omega_i (d_i(k) - \sigma_i(k))^2 + J_{i+1}(k) \quad \text{si } i \leq n-1 \quad (2.16)$$

On a l'identité suivante :

$$\delta_i(k) = -\frac{\partial J(k)}{\partial V_i(k)} = -\frac{\partial J_i(k)}{\partial V_i(k)} = -\frac{\partial J_p(k)}{\partial V_i(k)} \quad \forall p \leq i \quad (2.17)$$

D'où, en utilisant les expressions (2.12) et (2.16) et la formule des dérivations composées :

$$\delta_i(k) = 2\omega_i \dot{f}(V_i(k)) (d_i(k) - \sigma_i(k)) - \sum_{l=i+1}^{n-1} \frac{\partial J_{i+1}}{\partial V_l} \frac{\partial V_l}{\partial V_i} \quad (2.18)$$

Où $\dot{f}(V_i(k))$ désigne la dérivée de la fonction de transfert. On convient que la somme à droite de l'expression (2.16) n'existe pas si $i = n-1$.

Comme $l \geq i+1$, selon l'identité (2.17) on aura :

$$-\frac{\partial J_{i+1}}{\partial V_l} = \delta_l(k) \quad (2.19)$$

D'autre part :

$$\frac{\partial V_l(k)}{\partial V_i(k)} = \frac{\partial \left\{ \sum_{j=0}^{l-1} W_{lj} \sigma_j(k) + I_l(k) \right\}}{\partial \sigma_i(k)} \frac{\partial \sigma_i(k)}{\partial V_i(k)} = W_{li} f'(V_i(k)) \quad (2.20)$$

En remplaçant les termes calculés en (2.19) et (2.20) dans (2.18), on obtiendra l'expression définitive de $\delta_i(k)$:

$$\delta_i(k) = f'(V_i(k)) \left\{ 2\omega_i (d_i(k) - \sigma_i(k)) + \sum_{l=i+1}^{n-1} W_{li} \delta_l(k) \right\} \quad (2.21)$$

En résumé, si le neurone i est un neurone de sortie, $\delta_i(k)$ contient le terme $(d_i(k) - \sigma_i(k))$ qui représente l'erreur que commet le neurone i par rapport à la sortie désirée $d_i(k)$. De plus, la valeur de $\delta_i(k)$ dépend des valeurs de $\delta_l(k)$, où l est le numéro de neurones placés sur des couches supérieures à celle de neurone i . Cette information se propage donc en sens inverse de l'activation de $\sigma(k)$ et ne dépend que des erreurs commises par les neurones de sortie ainsi que des poids du réseau. Tout se passe comme si l'algorithme permettait d'attribuer une erreur commise à tout neurone du réseau, même ceux qui sont cachés. Ces considérations justifient le nom "retro-propagation des erreurs" pour cet algorithme. Ce dernier permet de calculer une valeur exacte du gradient : Il s'agit d'un algorithme de gradient total. On le présentera dans le paragraphe suivant :

2.4.3.4 L'algorithme du gradient total

Soit $\epsilon(t)$ une mesure de l'erreur commise par le réseau pour tous les exemples de l'ensemble d'apprentissage à l'itération t . Soit ϵ_{max} , l'erreur maximale acceptable et soit C_{ij} le booléen tel que :

- $C_{ij} = 1$ s'il existe une connexion du neurone j vers le neurone i ,
- $C_{ij} = 0$ sinon

Tous les exemples sont présentés en séquence durant une epoch d'apprentissage, qui constitue pour une itération de l'algorithme de gradient. Il y'en a autant que nécessaire pour que l'erreur $\epsilon(t)$ soit inférieure à ϵ_{max} . Avec un tel critère d'arrêt, l'algorithme peut ne pas s'arrêter si le coût quadratique du minimum vers lequel tend la matrice synaptique est supérieur à ϵ_{max} . En pratique, le critère d'arrêt de l'algorithme est aussi généralement plus sophistiqué pour éviter d'être bloqué dans cette situation.

Tant que $\epsilon(t) > \epsilon_{max}$, faire

Pour $k=0$ à $s-1$ faire

Pour $i=0$ à $n-1$ faire —Calcul de l'état du neurone—

$$V_i(k) = \sum_{j=0}^{i-1} W_{ij} \sigma_j(k) + I_i(k)$$

$$\sigma_i(k) = f(V_i(k))$$

Fin pour

— calcul de l'erreur par rétropropagation—

Pour $i=n-1$ à 0 faire

$$\delta_i(k) = f'(V_i(k)) \left\{ 2\omega_i(d_i(k) - \sigma_i(k)) + \sum_{l=i+1}^{n-1} W_{li} \delta_l(k) \right\}$$

Fin pour

Fin pour

—Calcul de la modification des poids après une présentation des exemples—

Pour $i=0$ à $n-1$ faire

Pour $j=0$ à $n-1$ faire

$$\Delta W_{ij} = \eta C_{ij} \sum_{k=0}^{s-1} \delta_i(k) \sigma_j(k)$$

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij}$$

Fin pour

Fin pour

Erreur (I, d, W) —Fonction qui retourne l'erreur commise par le réseau—

$t = t+1$

Fin Tant que

2.4.3.5 L'algorithme du gradient stochastique LMS

L'algorithme présenté au paragraphe précédent peut être très lent : Quelque soit la taille de l'ensemble d'apprentissage grand ou petit, le gradient n'est connu qu'après une epoch d'apprentissage. Cela implique qu'il n'y a qu'une modification des poids synaptiques par epoch. Il pourrait paraître plus intéressant d'utiliser une estimation du gradient obtenue avec des sous-ensembles de l'ensemble d'apprentissage. De cette façon, il y'aurait davantage de mise à jour des coefficients synaptiques par epoch et l'algorithme de gradient s'en trouverait plus accéléré. A la limite, on calcule une estimation du gradient pour chaque exemple présenté, et donc les coefficients synaptiques sont modifiés pour chaque exemple selon la quantité :

$$\Delta W_{ij} = \eta C_{ij} \delta_i(k) \sigma_j(k) \tag{2.22}$$

Où η est une quantité positive permettant la convergence de l'algorithme.

2.5 Principaux travaux dans le domaine de la mécanique

Plusieurs travaux de recherche témoignent de l'efficacité de cette technique à modéliser des comportements non linéaires complexes. A titre d'exemples, citons les travaux récents de Jeffrey Kuo et al. [Jef2006] en 2006 et Singh et al. [Sin2005] en 2005 qui portent sur la prédiction de l'amincissement de la tôle dans un procédé de formage hydrodynamique. Egalement, Xiong [Xio2005] s'est intéressé à l'évolution du dommage dans le cas d'un procédé de fromage. En 2005, Sanjay [San2005] a présenté des travaux de recherche qui portent sur la modélisation de l'usure d'outil dans le forage par analyse statistique et réseaux neurologiques artificiels.

Les premiers travaux utilisant les réseaux de neurones artificiels en mécanique, commencent à voir le jour dans le domaine de la robotique. L'utilisation des réseaux de neurones dans ce domaine a émergé dans les années 80 du fait de leur capacité à approcher des fonctions multidimensionnelles à partir d'un nombre réduit d'échantillons d'apprentissage. De nombreuses

architectures de réseaux ont été étudiées pour le contrôle de robot, avec des approches diverses de la commande, du rôle des réseaux et des apprentissages [Kuh1997]. Le premier article dans le domaine de la mécanique des structures a été publié par Adeli et Yeh en 1989 [Ade1989]. Et depuis cette date, les applications des RNA en mécanique se sont développées rapidement. Ils sont utilisés comme un outil de classification, de modélisation et d'identification.

En 1999, Pernot et al. [Per1999] ont illustré trois cas d'application des RNA dans leur publication. Par un réseau de type perceptron multicouche, ils ont construit un filtre des lois de comportement (figure 2.8). A travers un ensemble de points de la courbe contrainte-déformation, le réseau doit déterminer la loi de comportement du matériau. Ainsi, le réseau reçoit comme entrées les points (ϵ_i, σ_i) et répond par :

- Lc1 : Si le comportement est linéaire élastique.
- Lc2 : Si le comportement est parfaitement plastique caractérisé par le point $(\epsilon_{ref}, \sigma_{ref})$.
- Lc3 : Si le comportement est élastoplastique caractérisé par la contrainte seuil de plasticité σ_{ref} .
- Other : Si le comportement n'est pas connu.

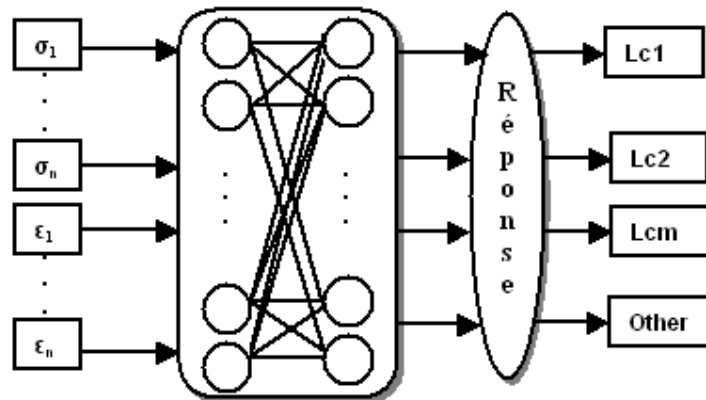


FIG. 2.8 – Réseau filtre des lois de comportement [Per1999]

Dans les mêmes travaux de recherche, un réseau de type perceptron multicouche a été construit pour simuler le comportement du béton sous l'effet d'une sollicitation de compression. La loi analytique de SARGUIN a été utilisée pour construire la base de données. Les équations de la loi sont présentées dans [Per1999]. Ils ont défini un réseau dont les entrées sont la déformation axiale ϵ et le paramètre σ_{ref} pour en déduire la variable de sortie représentative de la contrainte de chargement σ .

Des nouvelles méthodes utilisant les RNA pour la modélisation de la nano-indentation ont été traitées aussi par Muliana et al. [Mul2002]. Dans cette étude, ces algorithmes sont utilisés pour prévoir la réponse du métal pour une gamme de matériau et une géométrie

d'indentateur données. La validité des réseaux de neurones à modéliser la réponse est obtenue par comparaison avec des réponses obtenues par la méthode des éléments finis.

La topologie du réseau de neurone utilisé a deux couches cachées (figure 2.9).

Les entrées sont :

- Δ : Déplacement de l'indentateur,
- $\frac{E_f}{E_c}$: Rapport de rigidité,
- t_f : Épaisseur du film.

La sortie est :

- F : Force de l'indentation

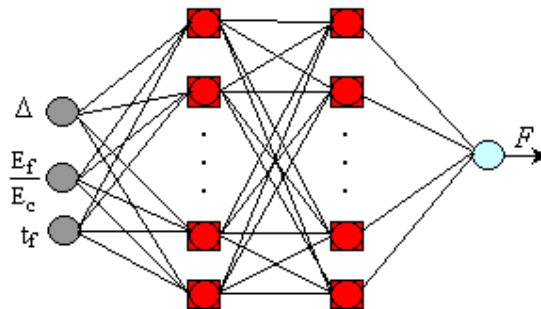


FIG. 2.9 – Modèle RNA permettant de déterminer la force d'indentation en fonction des paramètres d'entrées

L'apprentissage est réalisé pour chaque couple entrée/sortie dont le calcul est obtenu par EF en utilisant le code ABAQUS. Ensuite, les résultats obtenus par RNA ont été comparés à ceux obtenus par EF. Les résultats ont donné une bonne concordance.

Pour tester l'efficacité des RNA à modéliser le comportement du film mince, trois problèmes ont été traités dont la sortie est l'effort d'indentation F :

1. Le premier modèle RNA a comme paramètres d'entrées :
 - Δ : Déplacement de l'indentateur,
 - n : Coefficient d'érouissage.
2. Le deuxième modèle RNA a comme paramètres d'entrées :
 - Δ : Déplacement de l'indentateur,
 - n : Coefficient d'érouissage,
 - σ_0 : Seuil de plasticité.
3. Le troisième cas consiste à construire un modèle RNA qui tient compte de l'effet de l'érouissage du film.

Les modèles neuronaux proposés dans ce travail montrent de bons résultats vis-à-vis d'un modèle réalisé par la méthode des éléments finis.

L'avantage majeur de la modélisation par RNA est qu'on peut accéder à la réponse d'un problème dans un temps de calcul très court [Ham2002]. Dans cet article, l'outil neuronal a été utilisé pour substituer un calcul éléments finis. L'objectif de l'étude est de prédire la hauteur des bavures pendant l'opération de découpage en fonction de la variation des jeux et des rayons des outils. Pour cela le modèle RNA est conçu pour simuler les configurations suivantes :

- 12 valeurs de jeu : c (%) = {1, 3, 5, 7, 9, 10, 12, 14, 15, 17, 19, 20}
- 4 géométries des outils sont choisies tel que :
 - Une matrice présentant un rayon de $R_{wd} = 0.01\text{mm}$
 - Quatre poinçons de différents rayons de raccordement : $R_{wp} = \{0.01, 0.06, 0.12, 0.2\}$

Par suite, 48 simulations ont été exécutées pour la construction du modèle RNA pour chaque paramètre géométrique illustré précédemment. Le modèle permet de calculer la hauteur de bavure dans une tôle découpée. Une bonne corrélation a été observée entre la prédiction par le modèle RNA et les résultats expérimentaux.

En 2003, Lin a utilisé les RNA pour la modélisation d'un essai d'emboutissage hydrodynamique d'une pièce en forme de T. Le modèle proposé a permis la prédiction de la longueur maximale de la pièce en T qui peut être embouti en fonction des paramètres du procédé. Les paramètres tenus en compte sont l'épaisseur du matériau, le diamètre du flan et celui de la matrice. Le réseau neurologique a été entraîné à partir des résultats issus de la simulation par éléments finis. Les entrées du modèle sont : Le diamètre du flan (D), le diamètre de la matrice (d) et l'épaisseur du flan (t). La sortie est la longueur maximale de la pièce définie par le paramètre L . Une base de donnée de 24 combinaisons (entrées, sortie) a été générée par un modèle EF. Une fois le modèle est conçu, trois jeux de paramètres ont été testés. Le maximum d'erreur entre la réponse EF et celui RNA a été inférieur à 2.17%. Donc, ce résultat basé sur une illustration par éléments finis fournit ainsi la possibilité de gérer ce procédé par RNA [Lin2003].

La simulation numérique remplace l'expérience dans plusieurs cas pour plusieurs raisons : D'une part, l'expérience nécessite un matériel plus cher et un temps d'expérience plus lent. D'autre part, les modèles de comportement qui sont discrétisés dans des codes de calcul présentent toujours des lacunes surtout pour la phase inélastique. C'est pour ces deux raisons qu'en 2004, Tomonari et Mark ont justifié leur proposition pour la modélisation de la plasticité et de la plasticité cyclique en se basant sur l'outil neurologique [Tom2004]. Le modèle RNA proposé a été implémenté dans un code de calcul par éléments finis (figure 2.10). Ils ont montré que la déformation pour un comportement en plasticité cyclique d'un matériau a été estimée dans des limites de 3% d'erreur.

Genel [Gen2004] a utilisé aussi des modèles basés sur l'application des réseaux de neurones artificiels pour la prédiction des propriétés de fatigue en utilisant les informations issues des essais de traction pour 73 aciers. Les propriétés de fatigue de ces aciers ont été collectées à partir de la littérature. Ils ont été ensuite exploitées pour l'apprentissage des modèles. La déformation totale qui caractérise la courbe limite de déformation est exprimée de la façon

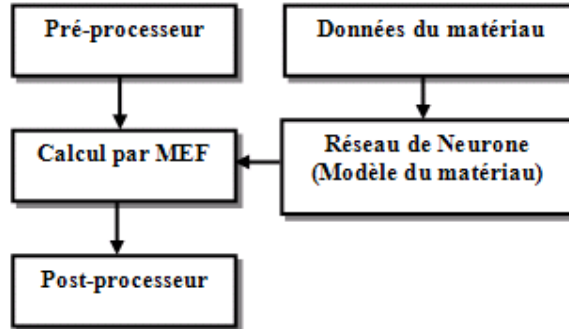


FIG. 2.10 – Détermination du comportement du matériau à partir d’un modèle neuronal [Tom2004]

suivante :

$$\frac{\Delta\varepsilon}{2} = \frac{\Delta\varepsilon_e}{2} + \frac{\Delta\varepsilon_p}{2}$$

$$\frac{\Delta\varepsilon}{2} = \frac{\sigma'_f}{E}(2N)^b + \varepsilon'_f(2N)^c$$

Avec :

σ'_f : Coefficient de résistance à la fatigue
 ε'_f : Coefficient de ductilité à la fatigue

Les entrées du modèle de Genel :

- E : Module d’Young
- R_A : Réduction de la section
- BHN : Dureté
- σ_y : Contrainte limite d’élasticité
- σ_u : Dureté

Les sorties du modèle de Genel sont :

- σ'_f : Coefficient de résistance à la fatigue
- ε'_f : Coefficient de ductilité à la fatigue
- c, b : Coefficients

Les résultats donnés par ce modèle indiquent que les valeurs des coefficients de résistance et de ductilité à la fatigue sont prédits respectivement à 99 et 98%.

Les réseaux de neurones artificiels sont aussi utilisés comme une technique d’identification et d’ajustement des paramètres matériels et géométriques dans plusieurs activités industrielles et plus particulièrement dans les procédés de mise en forme.

En effet, Pernot et al. [Per1999] ont proposé une méthode d’identification des paramètres matériels basée sur l’utilisation des RNA. Une comparaison avec les méthodes classiques

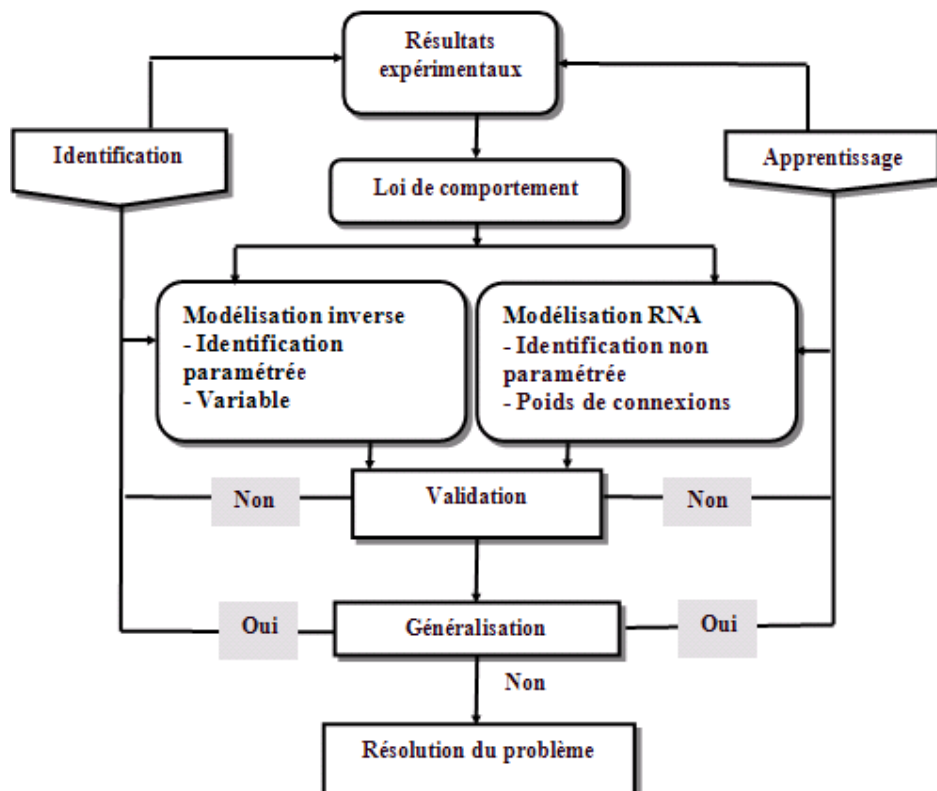


FIG. 2.11 – Comparaison entre méthode d'identification inverse classique et méthode d'identification se basant sur les réseaux de neurones artificiels [Per1999]

d'identification a été présentée dans cet article (figure 2.11). La méthode proposée dite méthode non paramétrée, ne s'intéresse pas à l'aspect physique des paramètres mis à l'identification. C'est pour cela qu'ils les classent au modèles "Blak-Box". L'ajustement des coefficients internes du modèle se fait directement à partir de l'expérience, alors que dans le cas d'une méthode d'identification classique l'ajustement des paramètres se fait après simulation et comparaison par rapport à l'expérience. Ils ont montré que l'utilisation des réseaux de neurones est une alternative à l'identification des lois de comportements.

En 2001, Huber et al. [Hub2001] ont largement utilisé cette même procédure pour l'identification des paramètres élastique et plastique de la matière lors d'un essai d'indentation pyramidale des films minces. Les données d'apprentissages sont générées par des essais expérimentaux et numériques.

Cette technique d'identification a été également utilisée par Martin et al. [Mar2003] pour l'identification de coefficient d'écrouissage et les paramètres d'endommagement d'un matériau ductile. Dans cette étude, un essai avec un petit poinçon (Small Punch Test) a été utilisé pour la détermination de la réponse du matériau sous un chargement de type complexe. En-

suite la courbe résultante qui représente l'effort en fonction du déplacement du poinçon, a été transférée vers un réseau neurologique. Ces courbes ont été générées par éléments finis pour chaque combinaison des paramètres matériels définis précédemment. Pendant la phase d'apprentissage le réseau neurologique produit une fonction rapprochée pour le problème inverse reliant les paramètres matériels à la forme de la courbe, effort en fonction du déplacement du poinçon. Cette technique a été utilisée pour trois différents matériaux. Les paramètres identifiés sont par la suite vérifiés et examinés en simulant un essai de traction. Les paramètres identifiés ici, sont ceux du modèle simulant la loi de comportement de Gurson, Tveergard et Needleman (GTN) qui prend en considération l'effet d'endommagement. Le modèle neuronal proposé par Martin et al. est capable de générer une fonction approchée des paramètres en fonction de la forme de la courbe effort déplacement du poinçon. L'exploitation du modèle a été faite expérimentalement. Les résultats obtenus dans ce papier sont très encourageants pour l'application de cet outil dans l'identification et l'optimisation des procédés de mise en forme.

Parmi d'autres applications de cet outil, on peut citer celles qui utilisent ces techniques pour la modélisation et l'identification de comportement des systèmes mécaniques. En 2001, waszczyszyn et al. [Was2001] ont montré que les RNA peuvent être appliqués pour la résolution des problèmes de types :

- Simulation : Chercher la réponse d'un système mécanique en sachant ses caractéristiques et les variables d'excitation.
- Simulation inverse : Chercher l'excitation à partir de la réponse du système mécanique ainsi que ses caractéristiques.
- Identification : Chercher les caractéristiques du système mécanique en sachant les variables d'excitation ainsi que la réponse.

Zmudzki et Kusiak [Zmu2005] ont développé un algorithme d'optimisation pour la détermination des paramètres optimaux des variables de conception. Cet algorithme a été appliqué dans l'analyse inverse d'un essai de compression à chaud.

En 2005, Bahloul [Bah2005] a développé un modèle numérique basé sur l'utilisation de RNA pour l'étude de procédé de pliage. L'objectif était d'établir des modèles numériques permettant de prédire les différentes réponses mécaniques et géométriques tel que les efforts de pliage, de choc dynamique et du retour élastique. Les résultats des applications traitées dans sa thèse permettent de conclure que les RNA donne une bonne concordance avec les résultats expérimentaux.

2.6 Conclusion

Dans ce chapitre nous avons détaillé la technique des réseaux de neurones artificiels. Au début, nous avons exposé les propriétés fondamentales de réseau de neurone suivi d'un ensemble d'arguments qui justifie le choix de cette méthode dans la modélisation et l'optimisation des systèmes industriels. Nous avons expliqué le fonctionnement d'un neurone formel et d'un réseau de neurone. Nous avons aussi, présenté les règles d'apprentissage d'un réseau non bouclé

multicouche. Finalement, un aperçu sur les principaux travaux publiés a été exposé. Il en ressort que l'utilisation de cet outil mathématique de modélisation et d'optimisation non paramétrique ne cesse d'augmenter dans plusieurs activités de recherche. Des nombreuses publications sur ce sujet ont permis de résoudre des problèmes présentant une complexité difficile à décrire avec des modèles mathématiques. A la base de ce qui a été montré dans ces deux premiers chapitres, nous entamons la partie développement d'un algorithme d'optimisation. Cet algorithme sera basé sur cet outil mathématique qui fera l'objet d'un troisième chapitre.

Chapitre 3

Développement d'un algorithme d'optimisation basé sur les RNA

3.1 Introduction

Pour modéliser un procédé de mise en forme d'une façon proche de la réalité, il faut prendre en compte la variabilité de tous les paramètres de l'opération. Une simulation numérique d'un tel procédé, que ce soit par une méthode analytique ou par une méthode des éléments finis, sera très coûteuse en temps de calcul. Il devient fortement nécessaire de développer des modèles de calcul "très rapide" permettant de prendre en compte de tous les paramètres intervenant dans un tel processus de fabrication.

Le désir de développer un système artificiel capable d'accomplir des tâches intelligentes dans la modélisation et l'optimisation des procédés de mise en forme dans un temps court, est l'origine du développement d'un outil de modélisation et d'optimisation à base de RNA. Ce type d'analyse est très utile pour la maîtrise de la production, la réduction de la variabilité et le risque de défaillance en contrôlant en amont les paramètres influents.

Nous présentons dans ce chapitre la démarche de développement que nous avons suivi pour coupler un programme à base de réseaux de neurones artificiels (RNA) avec le code de calcul par éléments finis ABAQUS. Cet algorithme se décompose essentiellement de trois parties :

1. Calcul paramétrique en utilisant le code de calcul ABASUS.
2. Construction d'un modèle RNA.
3. Exploitation du modèle RNA.

Cet algorithme est référencé dans certain passage par le nom PONN (Process Optimisation Neural Networks).

Nous nous intéressons plus particulièrement à la partie réseau de neurones constituant le cœur de cet algorithme. Ensuite nous présentons quelques exemples de validation.

3.2 Etude paramétrique

Pour construire un modèle neuronale nous avons besoin d'un plan d'expériences. Donc, nous avons fait appel à une étude paramétrique reliant les variables de sorties aux variables d'entrées dans un processus bien déterminé. Pour ce faire, on a développé un programme sous Python couplé au code de calcul Abaqus. Ce programme porte l'extension PSF. Il est structuré de la façon suivante (figure 3.1) :

1. Définition des paramètres de conception qui varient pour chaque calcul :
 - Paramètres géométriques
 - Paramètres matérielsCes paramètres peuvent être discrets ou continus
2. Précision des intervalles de variation de chaque paramètre (domaine de définition)
3. Combinaison des différentes valeurs prises par ces paramètres afin de former un ensemble des combinaisons.
4. Génération des fichiers .inp et lancer les calculs.
5. Extraction des réponses correspondantes.

Le domaine des valeurs possibles ou de la valeur de référence pour un paramètre donné peut être redéfinis temporairement lors de sa définition. On peut aussi imposer des contraintes de conception de type géométrique ou matériel.

Ainsi on obtient une base de données numérique reliant les variables de conception aux réponses correspondantes.

Chaque paramètre défini dans le fichier script doit être mis dans le fichier (.inp) sous la forme <Parameter>.

L'automatisation de calcul avec Abaqus se fait en deux modes :

- En utilisant Python et la commande *PARAMETER (figure 3.2).
- En utilisant Python en conjonction avec ABAQUS/CAE (figure 3.2).

Dans notre algorithme nous avons utilisé la première procédure d'automatisation.

Une fois un plan d'expériences est obtenu contenant les combinaisons de tous les paramètres d'entrés et leurs réponses, on construit le réseau de neurone. Ce programme est développé sous FORTRAN. Il permet de générer implicitement une fonction reliant les variables de conceptions aux variables d'entrées. Il est décrit dans le paragraphe suivant.

3.3 Programme RNA

Parmi les différents types de réseaux de neurones qui ont été présentés dans le deuxième chapitre, nous pouvons signaler qu'un réseau de type MLP a été choisi. Ce type est le bon candidat pour les applications que nous envisagerons par la suite. En effet, plusieurs travaux utilisent ce type de réseau. On peut citer à titre d'exemples les travaux de Shinc et

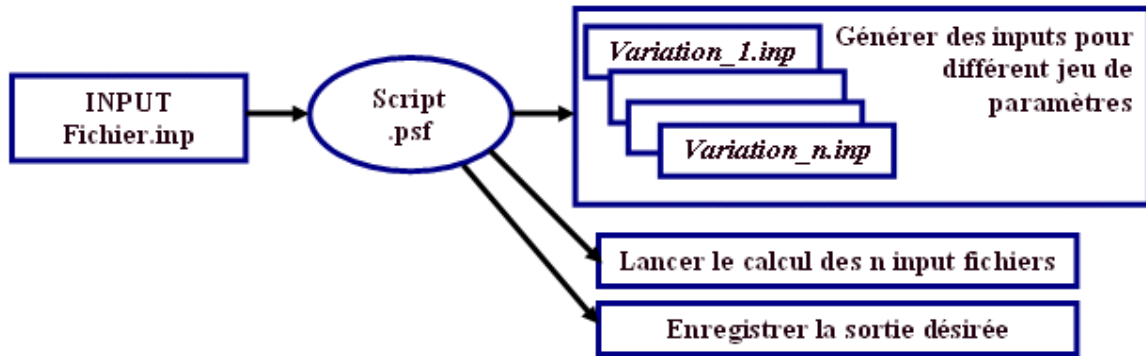


FIG. 3.1 – Calcul paramétré à l'aide d'ABAQUS

al.[Shi2003], de Tercej et al. [Ter2003], de Malinov [Mal2003], de Yuan et al. [Yua2004] et enfin de Dobrzanski [Dob2004]. Sa structure de connexion et son apprentissage par retro-propagation fais rappel au réseau à chaque amélioration des bons résultats pour ajuster ses paramètres internes.

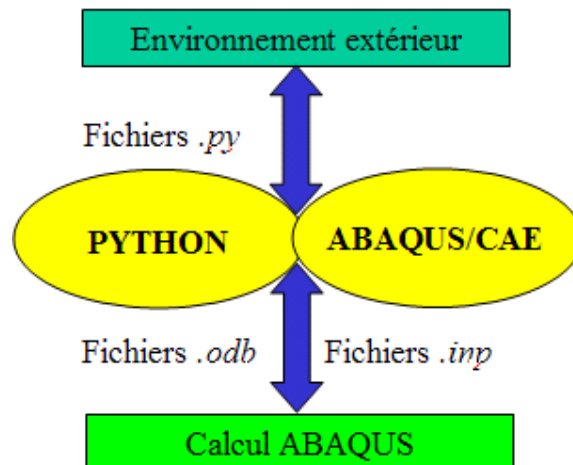


FIG. 3.2 – Interaction d'Abaqus avec python [Aba2001]

Pour la programmation et le développement de ce modèle RNA de type MLP (Multi-Layers Perceptron), nous rappellerons les considérations suivantes :

3.3.1 Structure

Les neurones sont organisés en couches : Chaque neurone est connecté à toutes les sorties des neurones de la couche précédente et sa sortie sera distribuée à tous les neurones de la couche suivante. Les entrées de la première couche sont définies comme étant les paramètres

d'entrées du réseau (figure 3.3). Les fonctions d'entrée et d'activation (chapitre 2) sont identiques pour tous les neurones d'une même couche, mais peuvent différer selon la couche. Le programme qui gère cette MLP est illustré par la figure 3.4.

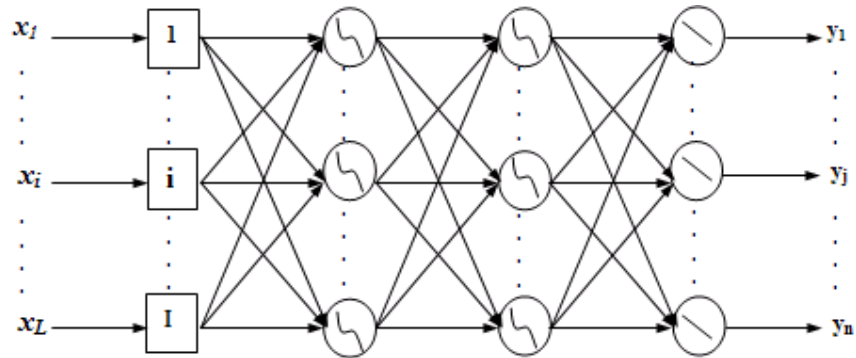


FIG. 3.3 – Structure d'un réseau MLP programmé

3.3.2 Normalisation des données

L'objectif de cette partie de modélisation est de ramener les données dans des intervalles de valeurs restreintes. Ceci est nécessaire pour éviter la saturation prématurée de la fonction d'activation de type sigmoïde (Chapitre 2) et permet de maintenir les coefficients synaptiques dans des intervalles relativement restreints. D'un autre point de vue, il s'agit de ramener toutes les entrées de même ordre de grandeur ce qui améliore la convergence de l'algorithme. Il existe plusieurs méthodes de normalisation :

1. "rescaling" : Simple réduction de chaque variable d'entrée X dans l'intervalle $[0,1]$. Appelée, aussi min-max normalisation.

$$\hat{X} = \frac{X - \min}{\max - \min} \quad (3.1)$$

Où $\max(\cdot)$ et $\min(\cdot)$ sont les valeurs min et max établies sur les données d'apprentissage. Cela signifie que les données sont ramenées dans un hypercube.

2. "Z-score normalisation" appelée aussi standardisation : Cette méthode fonctionne bien dans les cas où on ne connaît pas le minimum et le maximum des données d'entrée ou quand on a des valeurs qui dominent une normalisation min-max.

$$\hat{X} = \frac{X - \text{moyenne}}{\text{Std}} \quad (3.2)$$

Où Std est l'écart type appelé aussi "standard deviation"

3. Normalisation par échelle décimale : Il s'agit de normaliser les données par déplacement de la virgule décimale des valeurs. Le nombre de virgules décimales déplacées dépend de la valeur absolue maximale.

$$\hat{X} = \frac{X}{10^i} \quad (3.3)$$

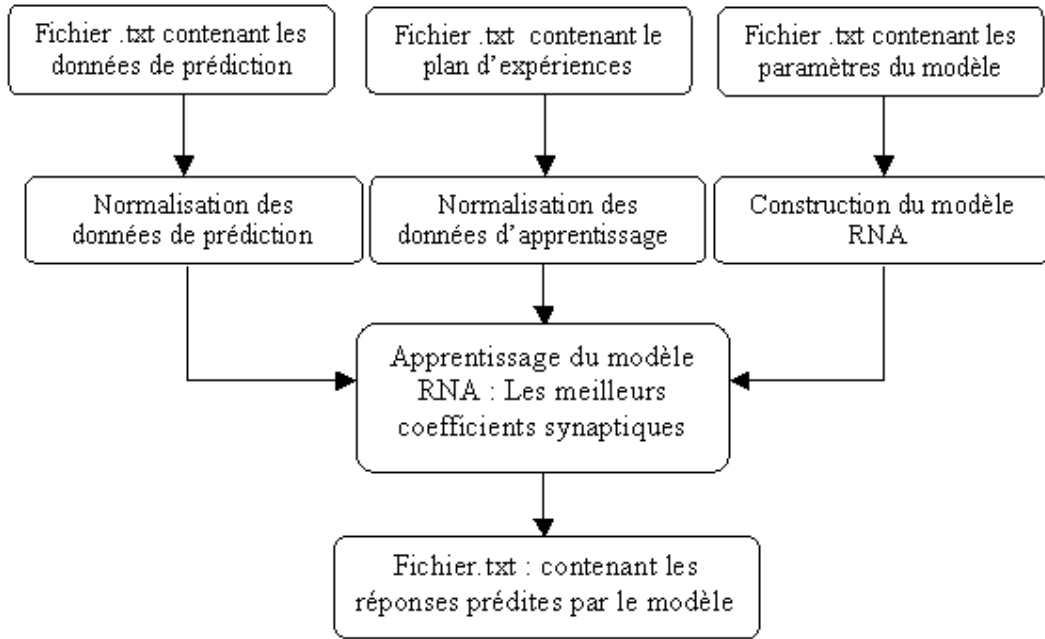


FIG. 3.4 – Boîte de construction de programme RNA

Où i est le petit valeur entier tel que $Max(|\hat{X}|) < 1$.

4. Normalisation sigmoïde : Il s'agit de transformer non linéairement les données entre -1 et 1, en utilisant la fonction sigmoïde. On calcule la moyenne et l'écart type des données. Les points qui ont un écart type moyen sont placés dans la région linéaire de la fonction sigmoïde alors que les autres points sont situés le long des queues de la fonction sigmoïde.

$$X' = \frac{1 - e^{-a}}{1 + e^{-a}} \text{ avec } a = \frac{X - \text{moyenne}}{Std} \quad (3.4)$$

La méthode que nous avons employée dans notre algorithme est celle de " rescaling " car elle est simple et s'adapte bien à la fonction d'activation de type sigmoïde.

3.3.3 Ajustement du modèle par apprentissage

Une fois que nous avons fais un bloc de traitement des données dans l'algorithme (normalisation), nous construisons un bloc pour l'apprentissage du modèle RNA (figure 3.6). Cela signifie que nous recherchons l'optimum des coefficients synaptiques. L'étude bibliographique effectuée nous a conduit au choix de la méthode de retropropagation de l'erreur pour l'optimisation des coefficients synaptiques. Elle consiste simplement en une descente de gradient qui est une méthode d'optimisation universelle. On cherche à minimiser une fonction erreur (qui représente l'erreur entre la sortie désirée et la sortie obtenue), en suivant les lignes de plus grande pente. Une fonction d'erreur rapportée à une dimension peut se représenter ainsi :

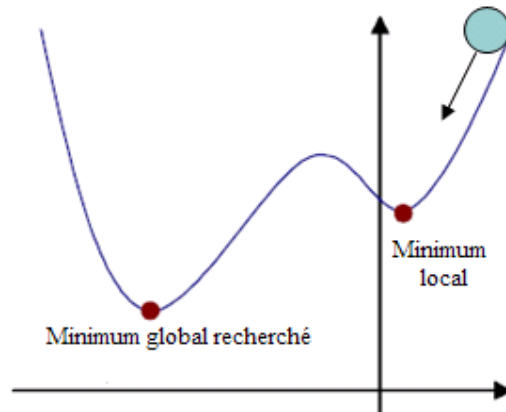


FIG. 3.5 – Illustration de la descente du gradient utilisé dans notre programme

On peut représenter la descente du gradient comme une bille que l'on poserait sur la courbe, et qui descendrait logiquement la pente (le gradient est la dérivée en plusieurs dimensions et représente donc la pente de la courbe) (figure 3.5).

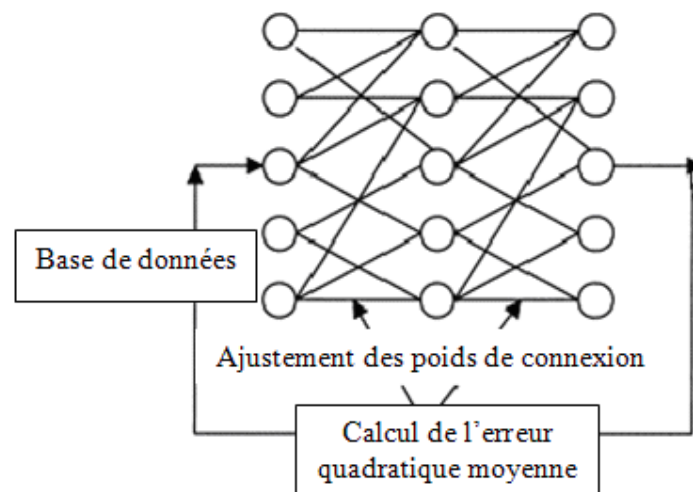


FIG. 3.6 – Processus d'apprentissage

La retropropagation de base utilise le gradient de l'erreur globale, mais s'arrête dans le premier minimum local rencontré. Diverses améliorations ont été apportées à l'algorithme :

- La descente stochastique : Il s'agit de minimiser itérativement l'erreur due à chaque exemple. Mais les apprentissages s'influencent les uns les autres, cela permet de passer sur des petits minima locaux. Une présentation aléatoire des exemples dans notre programme donne de meilleurs résultats.

- La descente avec inertie : Il s’agit d’introduire un moment d’inertie (Coefficient de moment= α), qui correspond dans notre image de la bille (figure 3.5), à l’inertie qu’elle acquiert en descendant la courbe : Son élan lui permettra de ne pas s’arrêter dans le premier minimum local rencontré. C’est la méthode que nous adoptons par la suite. Ce terme permet d’éviter les oscillations de l’algorithme pour des valeurs importantes de η . Donc les différentes étapes lors de son exécution sont présentées dans le paragraphe suivant :

3.3.4 Exécution

Une présentation généralisée des transformations mathématiques a été établie dans le deuxième chapitre de ce mémoire. Notre programme fonctionne suivant l’ordre des étapes suivantes :

Etape 1. Sélectionner un taux d’apprentissage η et un coefficient de moment α de façon aléatoire.

Etape 2. Prendre des valeurs initiales aléatoires des coefficients synaptiques compris entre 0 et 1.

Etape 3. Calculer la sortie du réseau y_m en utilisant les relations suivantes (Figure 3.3).

$$y_m = f(v_m), \quad v_m = \sum_{i=1}^L W_{mi}x_i + b_i \quad (3.5)$$

Etape 4. Calculer l’erreur entre la valeur de la sortie du réseau et celle désirée suivant la relation suivante.

$$E = \frac{1}{2} \sum_{j=1}^P \sum_{m=1}^N (d_{jm} - y_{jm})^2 \quad (3.6)$$

Où P représente le nombre total des paires d’exemples d’apprentissage, N est le nombre des sorties du réseau, d_{jm} est la sortie désirée et y_{jm} est la sortie du modèle RNA.

Etape 5. Arrêter si le critère d’arrêt est atteint, sinon passer à l’étape 6.

Etape 6. Calculer la contribution d’erreur pour chaque neurone couche par couche.

Etape 7. Mettre à jour les poids de connexion.

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} + \alpha W_{ij}(t-1) \quad (3.7)$$

$$W_{ij}(t+1) = W_{ij}(t) + \Delta W_{ij} \quad (3.8)$$

Avec η est le taux d’apprentissage et α est le coefficient de moment.

Etape 8. Répéter dès l’étape 3.

3.3.5 Critères d’arrêt et considérations pratiques

Plusieurs critères d’arrêt peuvent être utilisés avec l’algorithme de retropropagation des erreurs. Le plus commun consiste à fixer un nombre maximum de périodes d’entraînement,

ce qui fixe effectivement une limite supérieure sur la durée d'apprentissage. Ce critère est important car la retropropagation des erreurs n'offre aucune garantie quant à la convergence de l'algorithme. Il peut arriver par exemple, que le processus d'optimisation reste compris dans un minimum local. Sans un tel critère, l'algorithme pourrait ne pas se terminer. Un deuxième critère commun consiste à fixer une borne inférieure sur l'erreur quadratique moyenne ou encore sur la racine carrée de cette erreur. Dépendant de l'application, il est parfois possible de fixer a priori un objectif à atteindre. Lorsque l'indice de performance choisi diminue en dessous de cet objectif, on considère simplement que le MLP a suffisamment bien appris ses données et on arrête l'apprentissage.

Les deux critères précédents sont utilisés dans notre programme, mais ils comportent aussi des limitations. Le critère relatif au nombre maximum de périodes d'entraînement n'est pas du tout lié à la performance du réseau. Le critère relatif à l'erreur minimale mesure quant à lui un indice de performance. Mais ce dernier peut engendrer un phénomène dit de sur-apprentissage (figure 3.7). Ce phénomène n'est pas désirable dans la pratique, surtout si l'on ne possède pas une grande quantité de données d'apprentissage ou si ces dernières ne sont pas de bonne qualité.

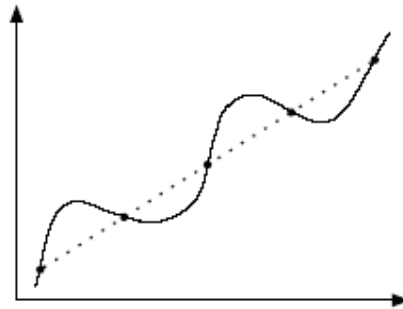


FIG. 3.7 – Illustration du phénomène de sur-apprentissage

Un processus d'apprentissage par correction des erreurs, comme celui de la retropropagation vise à réduire autant que possible l'erreur que commet le réseau. Mais cette erreur est mesurée sur un ensemble de données d'apprentissage. Si les données sont bonnes, c'est-à-dire qu'elles représentent bien le processus physique sous-jacent que l'on tente d'apprendre ou de modéliser et que l'algorithme a convergé vers un optimum global, alors il devrait bien performer sur d'autres données issues du même processus physique. Cependant, si les données d'apprentissage sont partiellement corrompues par du bruit ou par des erreurs de mesure, alors il n'est pas évident que la performance optimale du réseau sera atteinte en minimisant l'erreur, lorsqu'on la testera sur un jeu de données différent de celui qui a servi à l'entraînement. On parle alors de la capacité du réseau à généraliser, c'est-à-dire de bien prédire avec des nouvelles données. Par exemple, la figure 3.7 illustre le problème du sur-apprentissage dans le contexte d'une tâche d'approximation de fonction. La droite en pointillés montre une fonction linéaire que l'on voudrait approximer en ne connaissant que les points noirs. La courbe en trait plein montre ce qu'un réseau pourrait apprendre. On constate que la courbe

passer par tous les points d'entraînement et donc que l'erreur est nulle. En évidence, ce réseau ne généralisera pas bien si on échantillonne d'autres points sur la droite.

Pour remédier à ce problème, nous avons utilisé dans le programme un autre critère d'arrêt pour régler les paramètres du réseau (coefficient de moment, nombre d'époque, taux d'apprentissage) qui se base sur une technique dite de validation croisée. Cette technique consiste à utiliser deux ensembles indépendants de données pour entraîner notre réseau : un pour l'apprentissage (l'ajustement des poids) et l'autre pour la validation. Le critère d'arrêt consiste alors à stopper l'apprentissage lorsque l'ensemble de validation présente un minimum d'erreur. Cette dernière traduit la différence entre les points réels et les points prédits par le réseau. Une autre considération pratique consiste à permuter l'ordre de présentation des exemples d'apprentissage pour chaque période d'entraînement. En effet, la performance de la méthode de la descente du gradient peut dépendre fortement de cet ordre de présentation, qui engendre des trajectoires différentes dans l'espace des paramètres pouvant amener à différents minimums locaux. Donc le fait de permuter les données à chaque période nous garantit que l'on ne tombera pas systématiquement sur les mêmes minimums locaux. Ainsi on obtient un exécutable contenant les poids optimaux.

3.3.6 Validation

Le programme RNA est totalement paramétrable et il possède deux modes : Direct et inverse.

- Direct : Le programme calcule la valeur de sortie associée quand on lui fournit une valeur d'entrée.
- Inverse : Le programme permet de prédire le ou les facteurs d'entrées associés à une valeur de sortie donnée.

Pour chaque couche de neurone on peut choisir :

- Le nombre de neurones par couche.
- La fonction d'activation, bien que nous utilisons dans tous les exemples une fonction de type sigmoïde.
- Les coefficients d'apprentissage et de moment.
- La méthode de normalisation, bien que nous avons utilisé uniquement la méthode rescaling illustrée précédemment.

Ce programme a été testé sur des expressions analytiques au début, puis nous l'avons mis en œuvre sur des applications réelles par couplage avec le code de calcul par éléments finis Abaqus.

3.3.6.1 Fonctions analytiques

Ce programme est développé sous Fortran et il est testé sur des exemples qui représentent des fonctions analytiques. En effet, cela présente un intérêt pratique important puisque ce programme possède les avantages évoqués précédemment et qu'il peut être couplé avec d'autres logiciels de calcul. Il a aussi un autre intérêt : Il peut être utilisé pour n'importe quelle autre application. On voit immédiatement dans les paragraphes qui suivent la comparaison avec des fonctions analytiques : Des courbes analytiques par conséquent qui collent

bien avec les résultats prédits par ce programme.

1. Fonction à une seule variable

Pour la prédiction des fonctions à une seule variable présentées dans figure 3.8, nous avons utilisé une architecture d'un réseau de type 1-2-1 : 1 neurone dans la couche d'entrée (x), 2 neurones dans la couche cachée et 1 neurone au niveau de la couche de sortie (Y). Pour calibrer les réponses nous avons ajusté les autres paramètres du modèle RNA : Nombre d'époch, le taux d'apprentissage et le coefficient de moment. Pour toutes les fonctions, le nombre maximal d'époch servi pour la phase d'apprentissage du réseau n'a pas dépassé 1000. La convergence de l'algorithme est rapide pour ces fonctions. Le maximum d'erreur entre la réponse analytique et celle prédite par le modèle RNA était de 10^{-6} , enregistrée dans le cas de la fonction sinus.

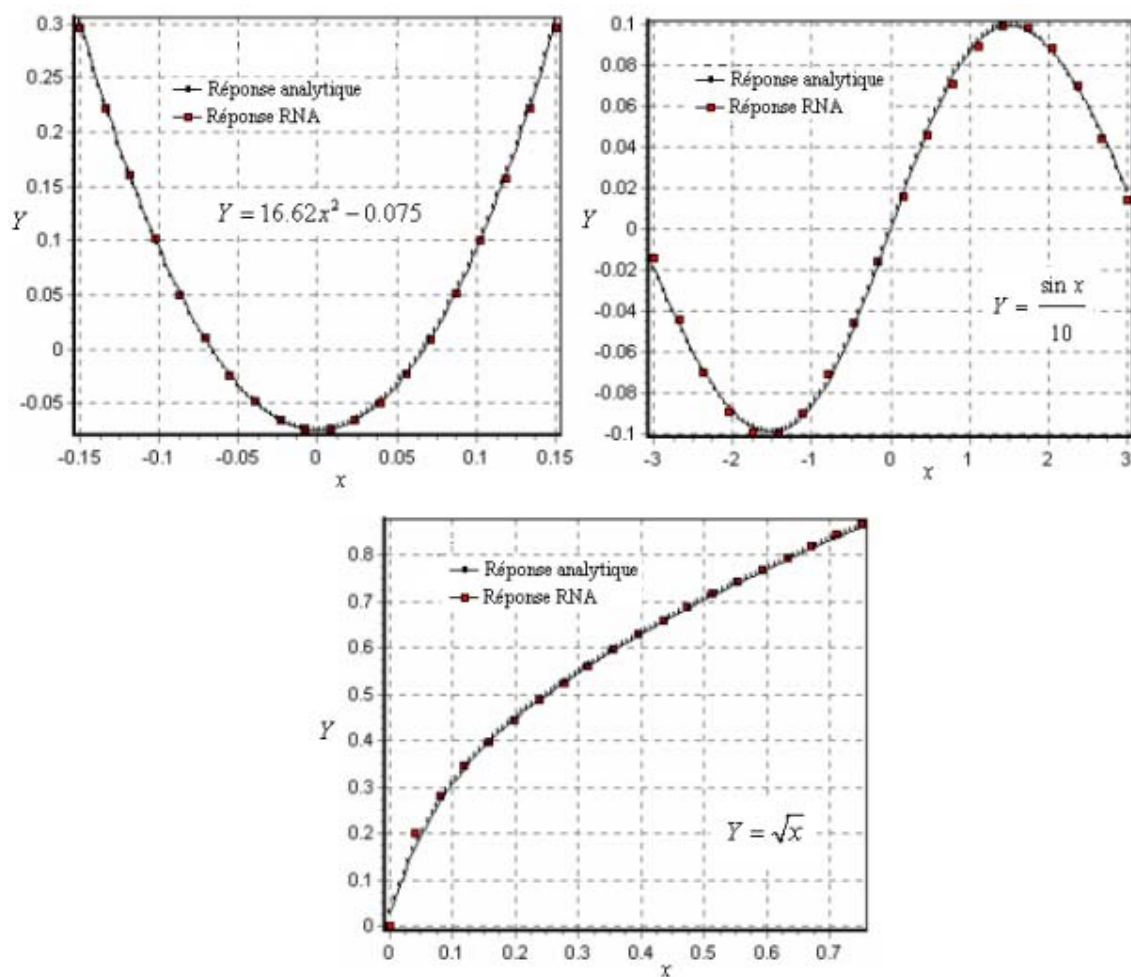


FIG. 3.8 – Validation du modèle RNA avec des fonctions analytiques à une seule variable

2. Fonction à plusieurs variables

Dans ce cas, nous avons testé le programme avec deux fonctions analytiques dont les expressions sont :

$$Y_1 = 2x_1^2 - 3x_2^2, \text{ avec un choix arbitraire des entrées tel que : } \begin{cases} x_1 \in [-20, 20] \\ x_2 \in [-30, 10] \end{cases}$$

$$Y_2 = \frac{x_1^2}{x_2^2 - x_3^2}, \text{ avec un choix arbitraire des entrées tel que : } \begin{cases} x_1 \in [-86, 64] \\ x_2 \in [74.6, 89.6] \\ x_3 \in [46, 83.5] \end{cases}$$

Pour l'ajustement des réponses prédites avec les réponses analytiques, nous avons commencé à faire varier le nombre de neurone de la couche cachée. Ce dernier est fixé à 5 neurones après un ensemble de tests manuels. Ensuite nous avons joué sur les autres paramètres pour l'ajustement de deux réponses.

Pour un taux d'apprentissage égal à 0.1 et un coefficient de moment égal à 0.12, le nombre d'épochs conduisant à la convergence du programme pour la première fonction (Y_1) est de l'ordre de 1000. Une erreur de $0.2 \cdot 10^{-5}$ est atteinte pour la deuxième fonction avec les mêmes coefficients décrits précédemment et un nombre d'épochs pour l'apprentissage du réseau égal à 1500. La structure du réseau pour la première fonction est 2-5-1, alors que celui de la deuxième fonction est 3-5-1.

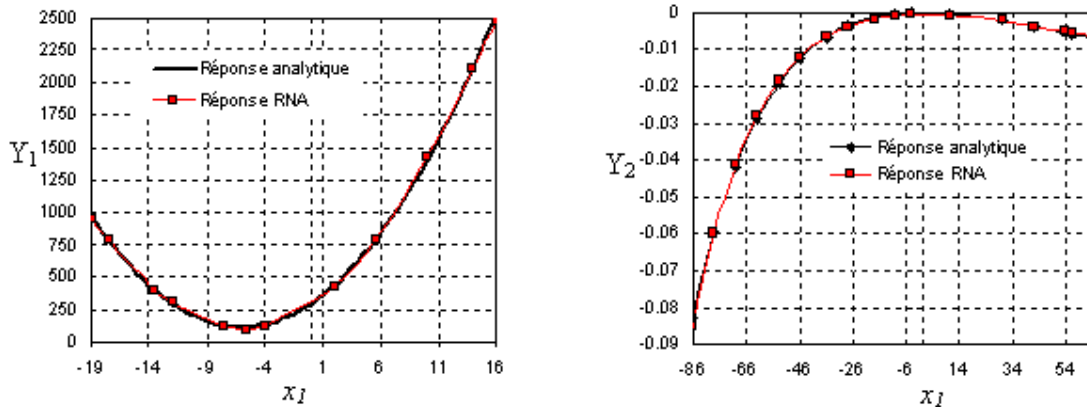


FIG. 3.9 – Validation du modèle RNA avec des fonctions analytiques à plusieurs variables

3.3.6.2 Prédiction par attribut

Soit à prédire la dureté d'un matériau trempé à différentes températures et refroidi dans différents milieux (eau, huile, air). On attribut les valeurs suivantes pour chaque milieu :

1. Refroidissement dans l'eau.
2. Refroidissement dans l'huile.

3. Refroidissement dans l'air.

Le tableau suivant présente une comparaison entre les valeurs expérimentales et les valeurs prédites par PONN. D'après ces exemples de validation, on constate que les réseaux de

Milieu		Température de trempe (c °)	Dureté HRC expérimentale	Dureté HRC Prédite
Eau	1	750	70	72.56
	1	800	80	81.16
	1	850	82	79.63
Huile	2	750	60	58.53
	2	825	65	65.12
	2	850	70	69.9
Air	3	750	40	40.14
	3	825	47	47.1
	3	850	50	49.98

TAB. 3.1 – Dureté HRC en fonction des paramètres milieu et température de trempe

neurones représentent donc une alternative qui peut être très efficace pour les problèmes que les algorithmes classiques ne peuvent pas résoudre. Leur aptitude à la généralisation représente l'un de leurs grands avantages.

Au début, l'apprentissage du réseau est très sensible aux paramètres du modèle (le taux d'apprentissage (η), le coefficient de moment (α) dans les expressions d'ajustement des poids de connexion et le nombre d'epoch). Après son ajustement, le modèle donne de bons résultats puisqu'il parvient à approcher différentes fonctions avec une bonne prédiction (figures 3.8 et 3.9). Dans le paragraphe suivant, on présente l'influence des ces paramètres sur la performance du programme.

3.4 Influence des paramètres du modèle RNA

Le réglage des valeurs des paramètres du modèle se fait dans la phase d'apprentissage. Ces paramètres affectent fortement les résultats des simulations. Les paramètres de notre modèle RNA qui seront étudiés par la suite sont respectivement, le nombre d'epoch (Nepoch), le nombre de neurones cachés (Nhidden) et le coefficient de moment (α). L'étude ci-dessous permet de mettre en évidence l'influence de différents paramètres sur la convergence des courbes prédites vers la solution exacte. Pour se faire, nous avons choisi la fonction analytique suivante : $Y = 2x_1^3 + 3x_2^2 - x_3$

Pour chaque paramètre d'entrée, nous avons défini son domaine de définition. Ces derniers sont choisis à différents ordres de grandeur afin de mettre en évidence l'utilité du bloc de normalisation dans le programme.

Avec :

- $x_1 \in [0, 0.7]$
- $x_2 \in [2, 37]$
- $x_3 \in [10, 1000]$

3.4.1 Influence du nombre d'époch

Dans ce paragraphe, des nombreuses simulations ont été réalisées en faisant varier le nombre d'époch. Les autres paramètres sont gardés à des valeurs constantes telles que :

- Le coefficient de moment : $\alpha = 0.19$
- Le nombre de neurone dans la couche cachée : $N_{hidden} = 10$
- Structure du réseau : 3-10-1

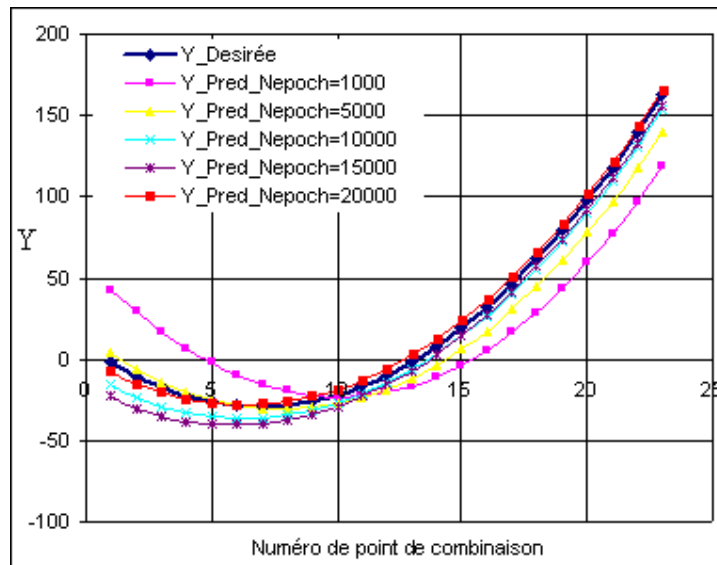


FIG. 3.10 – Comparaison de la réponse simulée par PONN pour différents nombres d'époques

On constate d'après la figure 3.10, que le nombre d'époques a une grande influence sur la prédiction du modèle. En effet, ce paramètre ne doit pas être aussi petit pour éviter le manque d'apprentissage et ne doit pas être très élevé pour éviter le phénomène de sur-apprentissage (figure 3.7).

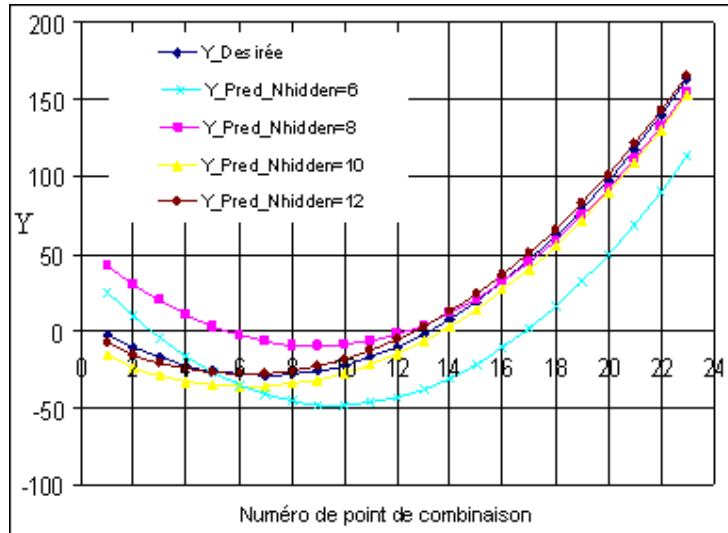


FIG. 3.11 – Influence de nombre de neurones cachés sur la réponse

3.4.2 Influence de nombre de neurones cachés

Pour étudier l'influence de nombre de neurones cachés sur les performances de notre modèle lors de la phase d'apprentissage, quatre valeurs ont été prises en considérations variant de 6 à 12 par incrément de 2 neurones. Les autres paramètres sont fixés aux valeurs suivantes :

- Le coefficient de moment : $\alpha = 0.19$
- Le nombre d'époch : $N_{epoch} = 25000$
- Structure du réseau : 3-6-1, 3-8-1, 3-10-1 et 3-12-1

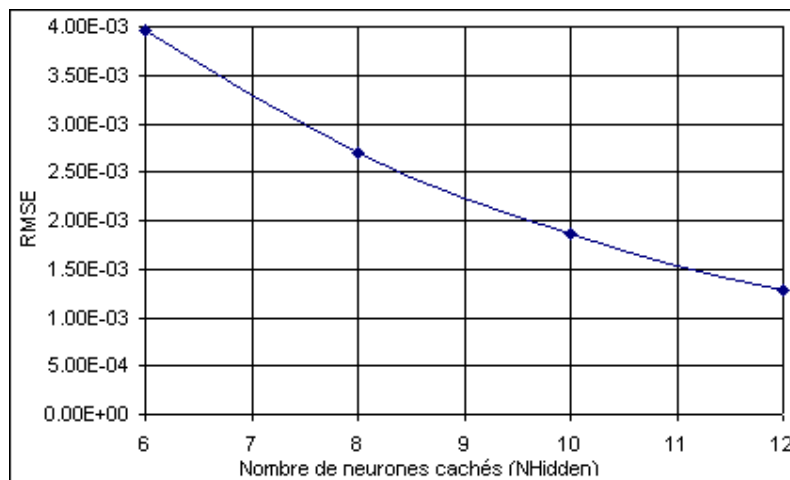


FIG. 3.12 – Influence de nombre de neurones cachés sur l'erreur quadratique moyenne

D'après les figure 3.11 et 3.12, on peut dire que les nombres de neurones dans les couches cachées influent beaucoup sur la prédiction. D'après la figure 3.12, il est constaté qu'il existe un nombre de neurone bien déterminé à partir du quel il n'y aura plus de diminution sur l'erreur quadratique moyenne. Donc, des simulations virtuelles seraient judicieusement réalisées jusqu'à la stabilisation de l'erreur. Dans cet exemple, la valeur minimale de l'erreur quadratique moyenne est atteinte pour un nombre de neurones dans la couche cachée égal à 12.

3.4.3 Influence de coefficient de moment

Le réglage du paramètre α vers sa valeur optimale est effectué pour une discrétisation de celle-ci en plusieurs valeurs. Pour chaque valeur du coefficient de moment nous avons repré-

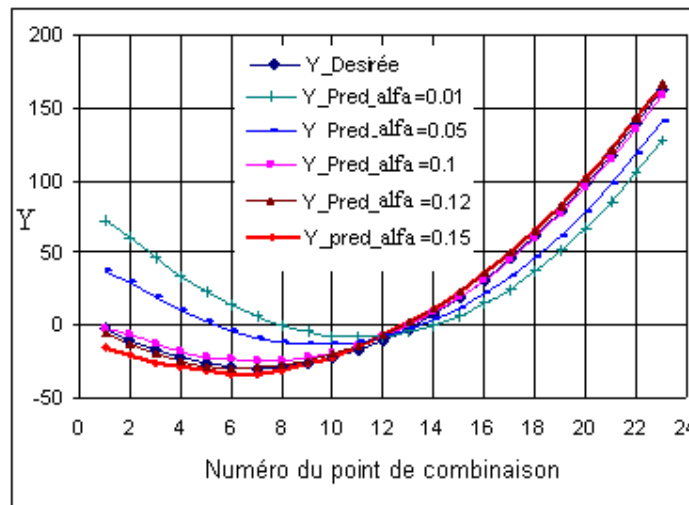


FIG. 3.13 – Effet du coefficient α sur la réponse

senté la réponse prédite et nous avons prélevé à chaque fois l'erreur quadratique minimale. Les valeurs des autres paramètres sont considérées comme suit :

- Le nombre de neurone dans la couche cachée : $N_{hidden} = 12$
- Structure du réseau : 3-1-1
- Le nombre d'époch : $N_{epoch} = 20000$

Sur la figure 3.14 nous avons représenté l'évolution de l'erreur quadratique moyenne en fonction de α . On a constaté qu'il existe un coefficient de moment optimal pour lequel l'erreur est minimale. Dans cet exemple, il correspond à une valeur $\alpha=0.12$. Donc, il est nécessaire de passer par une série de simulations manuelles afin d'aboutir à l'optimum de ce paramètre. On peut tirer quelques remarques qui seront adoptées dans la suite pour d'autres applications :

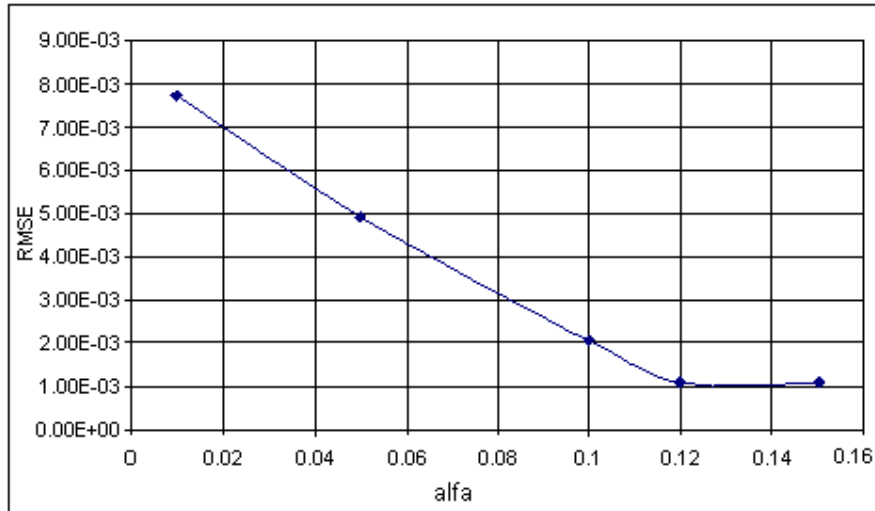


FIG. 3.14 – Effet du coefficient α sur l'erreur quadratique moyenne

1. Une valeur très faible de ce paramètre entraîne une convergence très lente de l'algorithme et une mauvaise prédiction.
2. Une valeur très élevée de α provoque une convergence rapide de l'algorithme et une mauvaise prédiction de la réponse.
3. La valeur optimale de ce paramètre est trouvée comprise dans notre modèle entre 0.1 et 0.2.

3.5 Algorithme

Le programme PONN a été développé sous FORTRAN alors que la partie couplage avec Abaqus a été écrite en langage de programmation orienté objet sous PYTHON. L'objectif est de générer à partir d'un ensemble des variables une base de données numérique à l'aide d'un modèle EF sous Abaqus et de prédire pour d'autres valeurs de ces variables situées dans leurs domaines de faisabilité leurs réponses par le modèle RNA.

Le programme comporte deux grandes parties : Une qui génère et exécute des fichiers inp pour la construction de la base de données tandis que l'autre partie exécute le programme PONN pour chercher les réponses en fonction des variables d'entrées ou bien de chercher les variables d'entrées optimales donnant une ou des réponses bien déterminées. Une fois le modèle RNA (PONN) est construit et qui représente une boîte noire caractérisant une fonction objectif reliant les variables d'entrées aux variables de sorties d'un système ou d'un processus quelconque il pourrait être exploiter dans la cadre d'une modélisation ou d'optimisation.

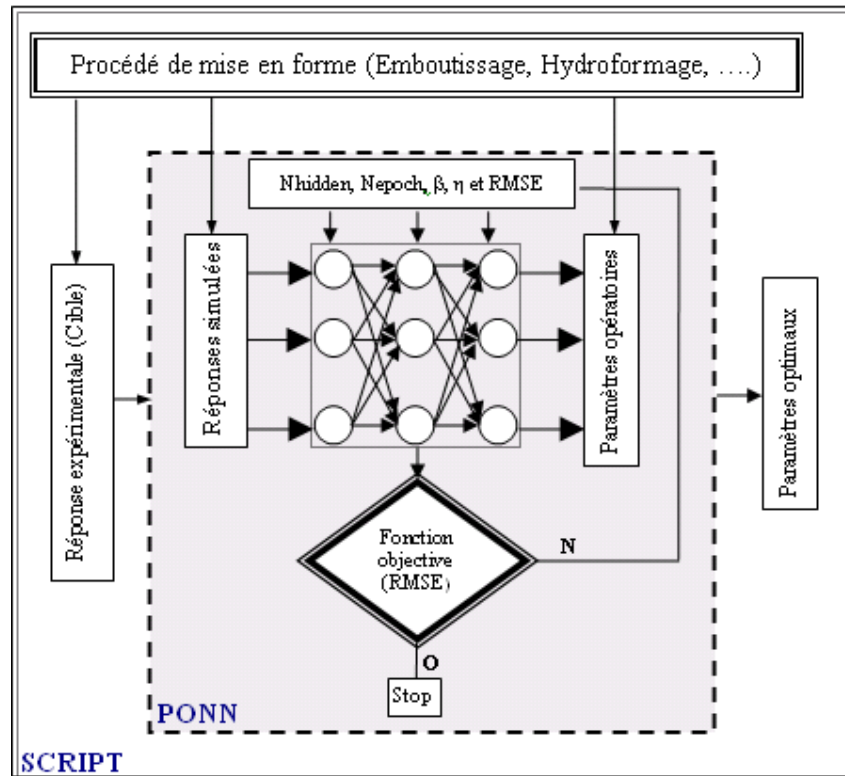


FIG. 3.15 – Oganigramme de l’algorithme d’optimisation automatique

Etant donnée que le champ d’application de notre étude est lié aux procédés de mise en forme, un algorithme d’optimisation a été développé selon la figure 3.15.

3.6 Exemple de validation

3.6.1 Mise en situation

Traditionnellement la réalité virtuelle traite la représentation et le mouvement de différentes structures sans tenir compte des propriétés de leur matériau pour être simulé rapidement. Or réellement, ces structures contiennent des modèles déformables basés généralement sur des vraies propriétés matérielles. Ces propriétés varient largement d’une structure à une autre, et par conséquent leur importance pour le comportement final du système doit être prévues [And1998], [Pop1999], [Jan2000], [Aba2001] et [Jan2002].

La structure se déforme généralement sous l’action des efforts externes agissant sur celle-ci tels que : Collision avec les autres objets, pesanteur, Le problème consiste à lier entre les entrées comme les excitations et les sorties qui représentent les réponses et les perturbations, résultantes du chargement de la structure pour l’application en temps réel en utilisant les vraies propriétés des matériaux qui la constitue [Bie2000], [Tan1998], [Kan1996], [Fos2002],

[Gib1984]. Donc les techniques traditionnelles de simulation en temps réel ne permettent pas de représenter exactement les déformations et les interactions complexes qui peuvent avoir lieu dans une telle structure. En effet, l'analyse par la méthode des éléments finis (MEF) basée sur la mécanique des milieux continus est largement considérée comme étant une alternative pour ce genre des simulations [Jan2002], [Pop1999], [And1998]. Cependant, en raison de leur nature fortement non linéaire, son application directe à la rétroaction en temps réel de force et visualisation de déformation de structure, n'a pas été pratiquée pour la plupart des applications. Cette limitation est principalement due à la longue période de calcul fournie par les approches standard de la méthode. Pour remédier à ça, des recherches actuelles utilisent une méthodologie s'appuyant sur des simulations numériques par des modèles à base de réseau de neurones artificiels et qui nécessitent une base de données antérieurement déterminée à partir des éléments finis. Comme illustration, de ce qu'on vient d'introduire et de validation du modèle développé, nous avons proposé de simuler l'impact d'une balle de tennis sur la raquette.

Le choix de cet exemple pour la validation de cet algorithme a été fait pour les raisons suivantes :

- Présence des non-linéarités géométriques,
- Présence des non-linéarités matérielles,
- Prise en compte de l'effet dynamique du système en faisant des simulations en dynamique explicite sur Abaqus.

3.6.2 Modélisation par éléments finis

Une simulation par la méthode des éléments finis est illustrée par la figure 3.16 en vue de construire le modèle neuronal. Les valeurs de sortie générées par la MEF correspondant aux couples des paramètres d'entrées, sont introduites dans le modèle RNA servant ainsi comme une base de données pour la phase de son entraînement. Après apprentissage, ce dernier va générer implicitement une relation entre les entrées et les sorties du système. Pour un tel système, une analyse de cause effet doit être faite pour identifier les causes principales et secondaires de la déformation et de la réaction de la raquette. Parmi les facteurs identifiés, on a retenu les plus importants et nécessaires qui sont les suivants :

- La vitesse de la balle : 5 valeurs sont retenues.
- L'angle d'impact : 4 valeurs sont retenues.
- La zone d'impact : 13 zones sont retenues (figure 3.16).

La combinaison de tous ces facteurs nécessite 260 expériences numériques ($13 \times 5 \times 4$). Ces dernières engendrent un plan d'expériences de type factoriel complet en tenant compte de toutes les interactions des trois variables.

Les cordes sur la raquette de tennis sont modélisées en utilisant des éléments de type (TRUSS). On suppose qu'elles sont élastiques linéaires avec un module de Young de 5 GPa et un coefficient de Poisson de 0.3. Sa masse volumique est égale à 600 kg/m^3 . Les cordes sont soumises à une pre-tension de 50 N. L'armature est supposée comme étant un corps rigide en utilisant des éléments rigides. Elle peut se déplacer dans toutes les directions (Six degrés de liberté). La balle de tennis est modélisée par une sphère, en utilisant 150 éléments de type coque (figure 3.16). On suppose qu'elle est en caoutchouc dont le matériau est

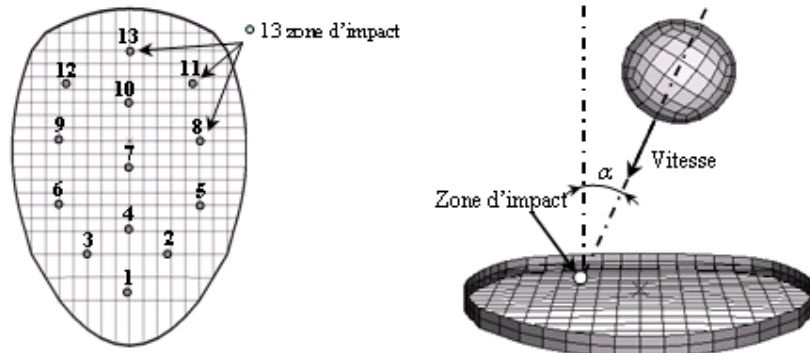


FIG. 3.16 – Modèle schématique de la frappe de la balle sur la raquette.

de type hyperélastique suivant la loi de Mooney-Rivlin [Ogd1984], [Mar1984], [Leg1984]. L'interaction à l'interface entre la balle et la raquette a été modélisée suivant la loi de "Coulomb" avec un coefficient de frottement égal à 0.1.

Le temps de calcul par éléments finis, fait pour l'étude de toutes les combinaisons possibles des paramètres d'entrées était environ de 13 heures sur une machine équipée d'un processeur Pentium 4 avec une mémoire vive de 512 Mégaoctets de RAM et une fréquences d'horloge de 3Ghz.

Les déformées de la raquette et de la balle à différents temps d'analyse sont présentées sur la figure 3.17. Les déformations de la balle et des cordes de la raquette pourraient être interprétées par une série de séquences successives d'impacts.

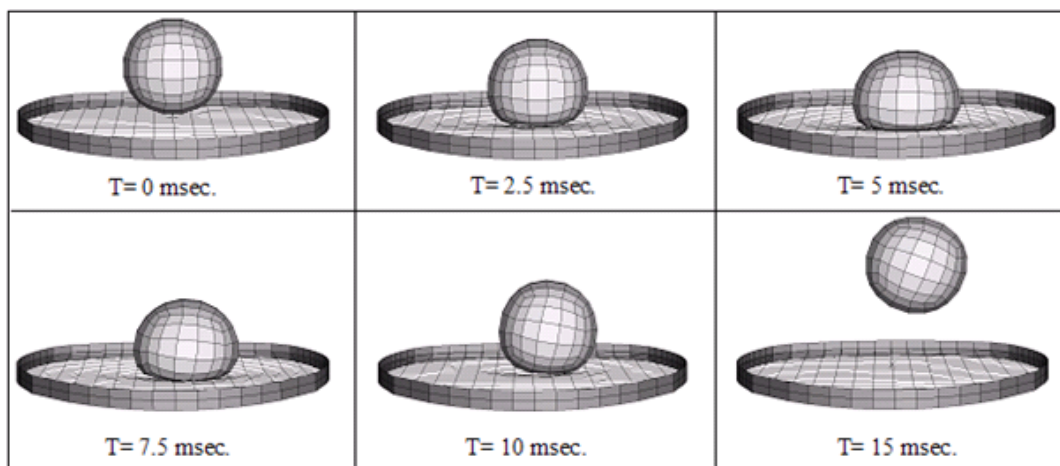


FIG. 3.17 – Différentes étapes de l'analyse

L'historique de la réaction de la raquette en fonction du temps est montré sur la figure 3.18. La valeur maximale de la réaction est atteinte au bout de 5 millisecondes. On peut remarquer que cette réponse commence à se stabiliser après une durée de 10 millisecondes. L'oscillation de la réaction pourrait être expliquée par une conséquence du taux élevé d'énergie cinétique se produisant par la vibration des cordes.

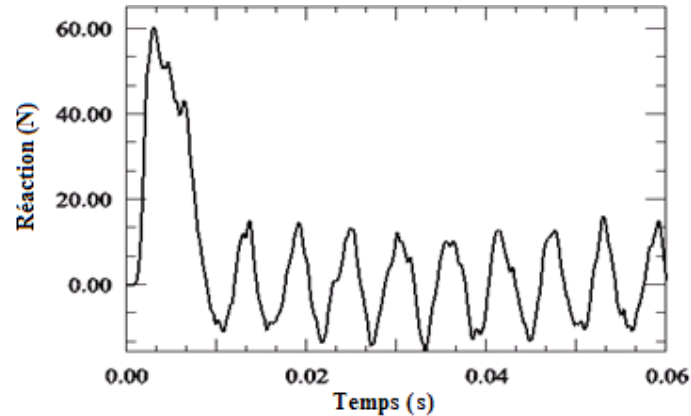


FIG. 3.18 – Réaction de la raquette en fonction du temps

La valeur maximale de la déformation de la balle ainsi que celle de l'intensité de la réaction de la raquette, ont été enregistrées pour chaque combinaison possible des paramètres étudiés.

3.6.3 Modèle RNA

Le modèle RNA choisi possède l'architecture 3-7-2 représentée par la figure 3.19 :

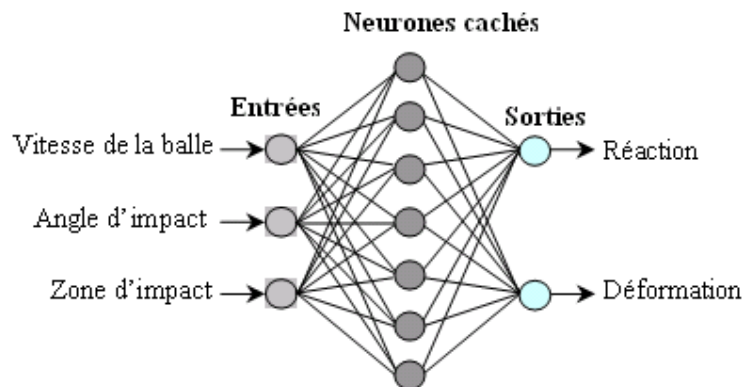


FIG. 3.19 – Structure du modèle RNA conçu

- 3 Neurones d'entrée : Vitesse de la balle, angle d'impact et zone d'impact.
- 7 Neurones cachés : Choisis après une phase de test.

- 2 Neurones de sorties : Déformation de la balle et la réaction de la raquette maximales.

Nous avons choisi aléatoirement 190 échantillons à partir de 260 pour la phase d'apprentissage de l'algorithme. Les 70 autres échantillons restants sont employés pour la phase de vérification du modèle conçu avec une possibilité de sa généralisation à d'autres applications.

3.6.4 Exécution de l'algorithme

Après préparation des différentes étapes pour chaque modèle MEF et RNA, l'algorithme sera exploité dans un calcul direct avec une exécution automatique. Ceci conduit à une prédiction des réponses pour les 70 autres combinaisons avec une éventuelle validation. Pour ce faire, les étapes suivantes doivent être suivies :

- Etape 1. Choix des paramètres opératoires. Ce sont les paramètres intervenant dans le processus. Ici nous nous sommes intéressés aux paramètres suivants :
 - V_i : Vitesse de la balle.
 - α_i : Angle d'impact de la balle.
 - Z_i : Zone d'impact de la balle.
 - RF_i : Intensité de réaction de la raquette.
 - ϵ_i : Déformation maximale de la balle.
- Etape 2. Choix de l'intervalle de variation propre à chaque variable : Dans cette étape, nous n'avons pas trop considéré la signification physique des variations des paramètres. Mais plutôt, nous étions intéressés à la partie validation de l'algorithme. Un choix arbitraire a été comme suit :
 - V_i : 5, 15, 25, 35, 45 et 60 m/sec.
 - α_i : 0°, 5°, 10°, 15° et 20°.
 - Z_i : {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 et 13}.
 - RF_i : Variable de sortie à prédire.
 - ϵ_i : Variable de sortie à prédire.
- Etape 3. Choix des paramètres du réseau de neurones artificiels
 - Nombre d'exemples pour l'apprentissage = 190
 - Nombre d'exemples pour la vérification = 70
 - Nombre d'Epochs = 4000
 - Momentum $\alpha = 0.19$
 - Taux d'apprentissage $\eta = 0.01$
 - Nombre de nœuds de la couche cachée NHidden = 7
- Etape 4. Exécution de l'algorithme

3.6.5 Résultats et interprétations

La figure 3.20 représente l'évolution de l'erreur quadratique moyenne en fonction de nombre d'epochs. Comme il peut être noté, l'apprentissage du réseau a entraîné une réduction significative de l'erreur commise entre les valeurs de sortie cibles et celles qui sont prédites. Initialement, l'erreur décroît d'une façon rapide jusqu'à 1000 epochs, au-delà elle termine par se stabiliser. D'après la même figure, une concordance bonne ou même assez bonne est obtenue entre les réponses calculées par le modèle EF et les valeurs de sorties prédites. Le maximum d'erreur pour toutes les combinaisons des paramètres reste inférieur à 1%.

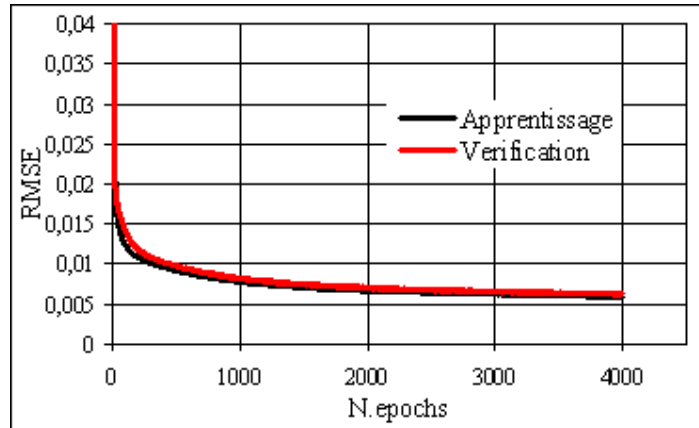


FIG. 3.20 – La moyenne de l’erreur quadratique RMSE en fonction de nombre d’epochs

La performance et la robustesse de l’algorithme développé résident dans sa rapidité et sa précision pour la prédiction des réponses pour d’autres valeurs des paramètres d’entrées. En effet, un temps nécessaire de calcul par le modèle RNA est de l’ordre de 0.01s alors que le temps enregistré pour la prévision par EF est égal à 180min.

3.7 Conclusion

Nous avons présenté dans ce chapitre la méthodologie suivie pour la construction d’une procédure automatique permettant d’être exploitée dans diverses applications. Cette procédure comporte essentiellement les sous programmes Script écrit en langage de programmation Python et un programme à base d’un réseau de neurones artificiels multicouches. Le programme Script comprend une partie pour la déclaration des paramètres opératoires à varier ensuite l’exécution du calcul par EF sur abaqus et enfin il permet de lancer le programme RNA à travers un exécutable. Des tests de validation et de vérification sont menés afin de voir les performances de l’algorithme. La bonne concordance issue de cette validation montre que cet algorithme fonctionne convenablement et il peut nous servir comme outil puissant pour résoudre certains problèmes complexes. Le couplage entre un calcul EF et un calcul RNA constitue une méthode de modélisation et d’optimisation très rapide que les méthodes classiques. Nous avons mis en application cette méthode sur deux exemples de mise en forme. Ceci permet d’une part, de s’affranchir de la lourdeur d’un calcul par les méthodes classiques et d’autre part d’obtenir une bonne précision sur les paramètres à optimiser.

Chapitre 4

Application des Réseaux de Neurones Artificiels à la modélisation et l'optimisation du procédé d'emboutissage

4.1 Introduction

En emboutissage des produits minces, l'étude de quelques paramètres comme le contour du flan, la géométrie des outils, les propriétés mécaniques de la tôle, la vitesse d'emboutissage et les efforts de serrage, est fondamentale pour la faisabilité d'une nouvelle pièce et de la fiabilité du procédé. L'objectif de la simulation numérique d'un tel procédé de mise en forme, est principalement l'amélioration de la qualité de la pièce finale et de la fiabilité du procédé par la maîtrise et l'optimisation de ses paramètres. L'optimisation des paramètres d'emboutissage a fait l'objet de plusieurs développements au cours de la dernière décennie. Les fonctions objectifs et les contraintes de limitation sont choisies essentiellement dans le but de réduire ou de contrôler la variation de l'épaisseur, les risques de striction et de plissement, le retour élastique, . . . Les variables de conception peuvent être de nature géométrique comme la forme du flan et la géométrie des paramètres du procédé ou bien de nature matérielle comme les coefficients d'anisotropie, l'exposant d'écrouissage et la rigidité du matériau. Pour connaître la zone de rupture, la localisation de l'amincissement minimale dans l'embouti est nécessaire.

Dans ce chapitre, nous nous intéressons à la modélisation numérique, la prédiction et l'optimisation du procédé de mise en forme par emboutissage à partir de quelques paramètres géométriques et matériels qui sont fournis antérieurement. L'algorithme développé dans le chapitre précédent est utilisé dans cette étude en se basant sur des essais numériques pour la prédiction des différentes réponses géométriques. Une comparaison entre les résultats obtenus par les deux approches éléments finis et réseau de neurones, a été établie. Trois applications sont envisagées :

1. Un calcul direct permet de déterminer l'épaisseur minimale dans un embouti en fonction de quelques paramètres matériels et du procédé.
2. Un calcul inverse pour l'identification du comportement du matériau constituant la pièce suite à une variation des paramètres matériels et du procédé.

3. Un calcul d'optimisation des paramètres géométriques pour réduire le phénomène du retour élastique.

Pour ce faire une gamme des matériaux a été définie par variation des leurs propriétés intrinsèques. Egalement, des variables géométriques et technologiques sont variées comme le frottement entre outils, le rayon de congé de la matrice, le rayon de congé du poinçon et le jeu entre les outillages. Dans la première et la deuxième application, nous avons fait varier les paramètres matériels pour une géométrie donnée et la réponse enregistrée, représente l'épaisseur minimale dans l'embouti. Quant à la troisième application, une variation des paramètres géométriques du procédé a été réalisée pour un matériau donnée et ceci mène à une prédiction de la géométrie de la pièce en relevant le retour élastique. Pour chaque combinaison de ces paramètres, le calcul des variables de sorties s'est fait par la méthode des éléments finis (MEF). Pour tester la validité du modèle RNA construit, nous avons procédé à d'autres simulations avec d'autres combinaisons de ces paramètres différentes à celles utilisées dans la phase d'apprentissage du réseau. A partir de la même base de données, des modèles RNA inverses sont construits permettant d'identifier les paramètres du matériau pour une réponse cible.

Dans la suite de ce chapitre, nous présenterons l'essai ainsi que les paramètres opératoires du procédé. Ensuite, les résultats d'une simulation numérique par EF seront présentés. Puis, la démarche suivie pour le calcul de l'amincissement maximal dans la pièce ainsi que l'identification des paramètres du matériau correspondants au moyen du modèle RNA, seront illustrés. A la fin de chapitre, une démarche d'optimisation visant la minimisation du retour élastique pour une variation de quelques paramètres géométriques du procédé sera exposée.

4.2 Opération d'emboutissage : Modélisation numérique

Considérons une opération d'emboutissage d'une pièce cylindrique de révolution à fond plat. Un flan circulaire de diamètre D_0 et d'épaisseur e_0 est préalablement posé sur une matrice aux dimensions extérieures de la pièce (D_m) avec un rayon d'entrée matrice R_m . Le flan est maintenu en place à l'aide d'un serre flan sur lequel un effort F_s est appliqué. Un poinçon aux dimensions intérieures de la pièce (D_p) avec un rayon de poinçon R_p vient alors enfoncer la tôle à l'intérieur de la matrice sous l'action d'un effort F_p .

Les principaux paramètres technologiques sont donc : Le rapport $\frac{D_0}{D_p}$, le jeu entre le poinçon et la matrice J , les rayons R_m et R_p , l'effort du serre flan F_s , le coefficient de frottement entre la tôle et les outils μ et la vitesse du poinçon V_p . La loi de comportement du matériau est supposée élastoplastique en considérant un écoulement isotrope. Le critère de plasticité répond à la formulation suivante :

$$f(\sigma, \alpha) = \sigma_c(\sigma) - \sigma_s(\alpha) \quad (4.1)$$

Avec : $\sigma_s(\alpha)$ est la fonction d'écrouissage donnée par la loi de Ludwick

$$\sigma_s(\alpha) = \sigma_0 + k(\epsilon_p)^n \quad (4.2)$$

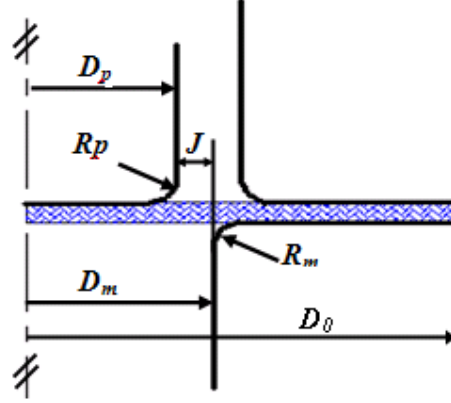


FIG. 4.1 – Principe d'emboutissage : Géométrie de la maquette numérique

Où n représente le coefficient d'écroutissage, k est la constante de rigidité et σ_0 est le seuil de plasticité. $\sigma_c(\sigma)$ est la contrainte équivalente au sens de HILL :

$$\sigma_c = (G + H)\sigma_{11}^2 - 2H\sigma_{11}\sigma_{22} + (F + H)\sigma_{22}^2 + 2N\sigma_{12}^2 \quad (4.3)$$

Avec σ_{ij} désigne les composantes du tenseur de contrainte de Cauchy dans les axes d'orthotropie. F , G , H et N sont les constantes qui définissent les coefficients d'anisotropie.

Les coefficients de Lankford sont exprimés en fonction des coefficients d'anisotropie comme suit :

$$r_0 = \frac{H}{G}, \quad r_{45} = \frac{2N - (F + G)}{2(F + G)}, \quad r_{90} = \frac{H}{F} \quad (4.4)$$

La loi d'écoulement plastique est donnée par :

$$\dot{\epsilon}^p = \dot{\alpha} \frac{\partial \sigma_c}{\partial \sigma}; \quad \dot{\alpha} > 0; \quad \dot{\alpha} f = 0; \quad \dot{\alpha} \dot{f} = 0 \quad (4.5)$$

Les réponses que nous cherchons à prédire par EF ou par RNA sont dépendantes des paramètres matériels ainsi que des lois qui les décrivent présentés ci-dessus. Généralement, l'écroutissage et l'anisotropie de la matière sont les paramètres matériels les plus influents sur le comportement de la tôle au cours de sa mise en forme par emboutissage.

- L'anisotropie est définie par les coefficients de Lankford noté par r_α . L'indice α désigne l'angle entre la direction de traction et la direction de laminage de la tôle. Plus le coefficient d'anisotropie r_α est élevé, plus la réduction d'épaisseur est faible par rapport au rétrécissement en largeur. Et par conséquent, une tôle qui n'a pas tendance à s'amincir, elle résistera mieux et le risque de rupture devient faible. D'où il sera préférable de travailler avec des matériaux possédant des coefficients d'anisotropie élevés. Ce qui permet d'améliorer la performance mécanique des pièces après leurs mise en forme. Dans le cas isotrope, r_α devient égal à 1 indépendamment de l'angle. Pour un emboutissage quelconque, où l'on ne connaît pas à priori l'orientation de la direction de laminage, on utilise la valeur moyenne :

$$\bar{r} = \frac{r_0 + 2r_{45} + r_{90}}{4} \quad (4.6)$$

- L'érouissage traduit l'aptitude d'un matériau à résister à la localisation de la déformation. Par exemple, dans le cas des matériaux satisfaisants la loi de Ludwick $\sigma_s(\alpha) = \sigma_0 + k(\epsilon_p)^n$ un coefficient d'érouissage n élevé traduit un comportement favorable du matériau. Par convention, la valeur du coefficient d'érouissage moyen d'une tôle doit être déterminée par la mesure des coefficients d'érouissage d'éprouvettes prélevées à 0° , 45° et 90° par rapport au sens de laminage de la tôle. Généralement, nous utiliserons la même formule que celle appliquée pour le calcul du coefficient moyen de Lankford. Elle est donnée par :

$$\bar{n} = \frac{n_0 + 2n_{45} + n_{90}}{4} \quad (4.7)$$

Dans ce qui suit, les principaux paramètres du matériau traduits par le coefficient de Lankford r_α caractérisant l'anisotropie et le coefficient d'érouissage n sont maintenus comme variables de l'étude de l'opération d'emboutissage.

4.3 Prédiction de l'épaisseur minimale dans l'embouti

4.3.1 Choix des paramètres opératoires

La figure 4.2 représente la maquette numérique faisant l'objet de cette étude pour la simulation de l'opération d'emboutissage associée à une géométrie donnée des outils [Yoo2000]. Tous les paramètres géométriques sont maintenus à des valeurs constantes. Dans cette étude, le coefficient de frottement est considéré comme variable puisqu'on suppose que son ajustement dépend de la nature de lubrifiant utilisé. Les différents facteurs opératoires seront donc les suivants :

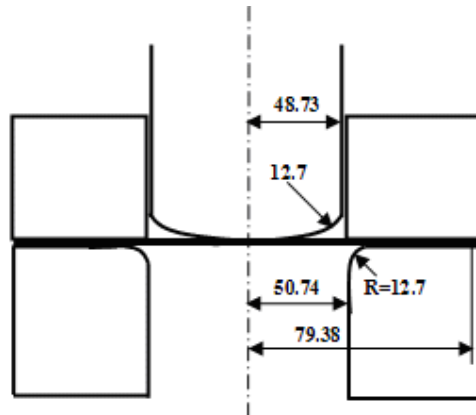


FIG. 4.2 – Géométrie des outillages

- Coefficient d'érouissage (n).
- Constante de rigidité (k).
- Coefficients de Lankford (r_0, r_{45}, r_{90}).
- Coefficient de frottement entre les outils et le flan (μ).

Concernant le choix des intervalles de variation des paramètres, chaque variable a été discrétisée suivant trois niveaux. Le tableau 4.1 récapitule les valeurs considérées pour chacun de ces paramètres. Un nombre total d'essai constituant une base de données est de 729 points de conception qui forment, ainsi un plan d'expériences envisagé pour la simulation numérique.

Niveaux	Paramètres					
	n	k	r_0	r_{45}	r_{90}	μ
1	0.1	400	0.5	0.5	0.5	0.05
2	0.2	700	1.4	1.4	1.4	0.1
3	0.4	1000	2.5	2.5	2.5	0.2

TAB. 4.1 – Les valeurs considérées des paramètres

4.3.2 Modèle éléments finis

Dans cette partie de travail un modèle numérique par éléments finis, a été développé sous ABAQUS simulant l'opération d'emboutissage pour tous les points de conception définis précédemment. Il permettra de prédire l'évolution de l'épaisseur dans toute la pièce après sa mise en forme ainsi que les zones de forte concentration de contraintes. Pour cela nous considérons les hypothèses suivantes :

- Pour des raisons de symétrie, nous avons considéré que seulement un quart du modèle complet. Le flan est modélisé dans l'hypothèse d'un corps déformable avec des éléments finis de type Shell. Le maillage de la pièce a été fait en deux domaines : La partie centrale de la pièce est maillée par des éléments triangulaires à 3 nœuds (S3R) tandis que la partie restante est faite en adoptant des éléments quadrangulaires à 4 nœuds (S4R) (figure 4.3).

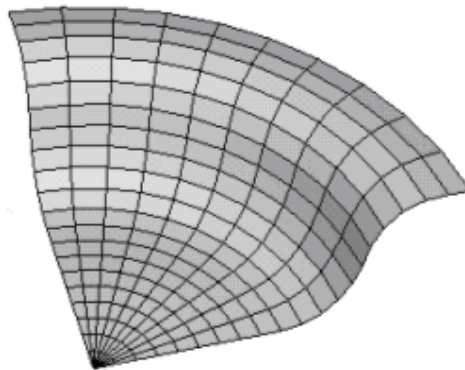


FIG. 4.3 – Maillage du flan

- Le comportement du matériau est donné par un modèle élastoplastique en grandes déformations et qui est implémenté dans le code ABAQUS utilisant le critère de Hill pour la prise en compte de l'anisotropie.

- Les conditions aux limites appliquées consistent en une force de serre-flan de 10kN, un déplacement imposé au niveau du poinçon de 65mm et des conditions de symétrie.
- Le poinçon, la matrice et le serre-flan sont modélisés dans l'hypothèse des corps rigides.

Les paramètres utilisés dans une première simulation sont présentés dans le tableau 4.2. Le nombre total des éléments est de 200. Comme résultat que nous cherchons à appréhender

n	k	r_0	r_{45}	r_{90}	μ
0.1	400	1.8	1.2	0.5	0.2

TAB. 4.2 – Les valeurs utilisées

ici, est la répartition de l'épaisseur dans l'embouti final. L'utilisation des éléments Shell nous permettra d'accéder directement à la variation de ce paramètre à travers la commande STH définie dans le code Abaqus [Aba2001]. La valeur minimale de cette épaisseur, sera relevé à la fin de résolution.

Les résultats issus de cette simulation montrent qu'il y'a un amincissement au niveau du rayon de congé du poinçon. Les figure 4.4-a et figure 4.4-b montrent, respectivement les iso-valeurs et la variation de l'épaisseur le long d'un méridien de l'embouti. D'après la figure 4.4-b, la valeur minimale de l'épaisseur de la forme emboutie pour ce jeu de paramètre est de l'ordre de 1.53mm. En pratique, pour prédire un tel paramètre dans ce procédé industriel, nous sommes amenés à chaque fois à lancer une simulation numérique par EF et de consommer un temps important de calcul. Chacune de ces simulations a durée approximativement une demi heure sur une machine équipée d'un processeur Pentium 4 à 1.6GHz et une mémoire vive 256 Mégaoctets.

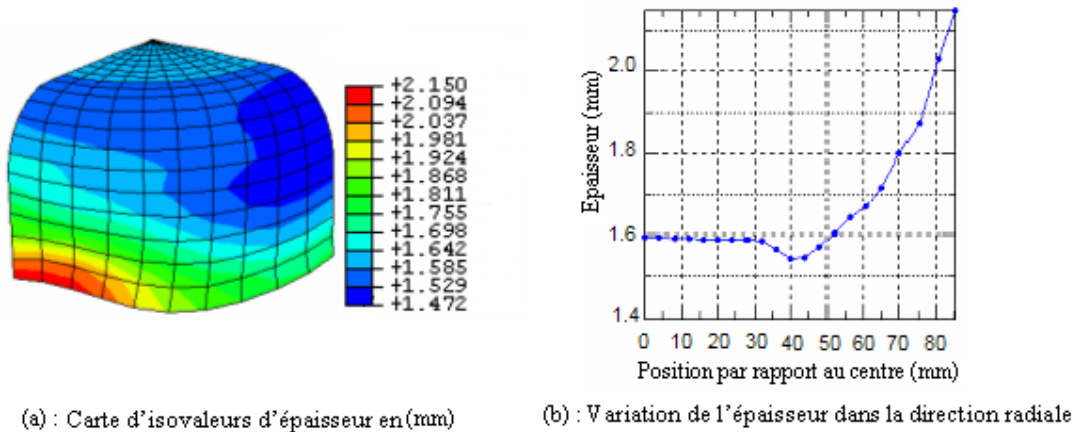


FIG. 4.4 – Distribution de l'épaisseur dans l'embouti en mm

4.3.3 Description générale de l'algorithme de calcul

Le schéma général de calcul, est donné par l'algorithme présenté par la figure 4.5. Il permet de calculer l'épaisseur minimale de la tôle et l'enregistré dans une base de données. Cette dernière va servir comme plan d'expériences pour l'apprentissage d'un modèle basé sur l'utilisation des réseaux de neurones artificiels. Donc, une matrice d'expérience de trois niveaux pour chacun des six paramètres a été construite après simulation de chaque point de conception.

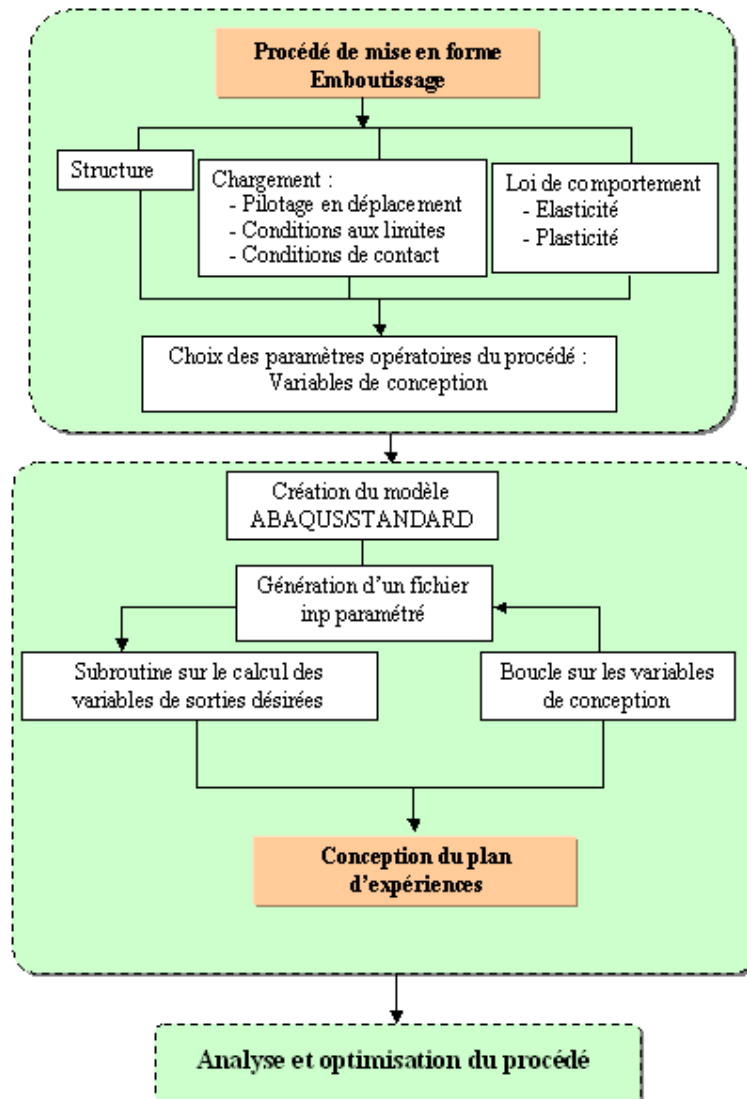


FIG. 4.5 – Schéma général de l'algorithme de calcul

4.3.4 Modèle RNA

Au début, l'architecture du modèle RNA est constituée de six neurones dans la couche d'entrée, deux neurones dans la couche cachée et un neurone dans la couche de sortie. L'apprentissage du modèle est habituellement lancé avec un nombre restreint de nœuds. Des nœuds additionnels sont alors ajoutés si le niveau d'erreur n'est pas réduit dans un nombre indiqué d'itérations. La méthode de retro-propagation est employée pour déterminer les poids de connexions par la minimisation de l'erreur. Des nœuds dans la couche cachée sont simultanément ajoutés, et les nouveaux poids de connexions sont actualisés en fonction du nombre d'itérations d'apprentissage. La fonction erreur est définie dans ce cas par l'équation (4.8) ci-dessous.

$$E = \frac{1}{2} \sum_{i=1}^P \sum_{k=1}^N (d_{ik} - o_{ik})^2 \quad (4.8)$$

Où d_{ik} est la sortie désirée (valeur numérique), o_{ik} est la sortie actuelle du modèle RNA, P représente le nombre total d'expériences numériques pour l'apprentissage et N représente le nombre total des neurones de sortie.

La figure 4.6 schématise la structure du modèle RNA proposé conduisant à un niveau d'erreur acceptable. Il se compose d'une couche d'entrée, d'une couche cachée avec 17 neurones et d'une couche de sortie. Les variables d'entrées qui sont employées dans ce cas sont le coefficient d'érouissage, la constante de rigidité, les coefficients d'anisotropie selon les axes d'orthotropie et le coefficient de frottement. La sortie du modèle RNA est l'épaisseur minimale dans l'embouti notée t_{RNA} .

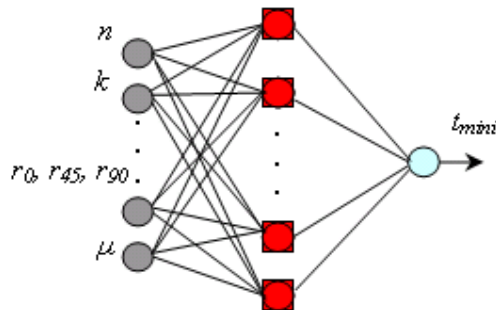


FIG. 4.6 – Structure du réseau de neurone employé

Le réseau est entraîné par un grand ensemble de paires de vecteurs d'entrées et de sorties (729) qui sont produites à partir du modèle EF. Les poids de connexions sont modifiés pendant l'apprentissage jusqu'à ce que le modèle RNA rapporte une réponse conforme d'une manière satisfaisante aux résultats obtenus par EF. En effet, le choix de l'ensemble des données d'apprentissage est très important. Comme dans n'importe quelle approximation, il n'y a pas de méthode claire pour produire une évaluation a priori complète de l'ensemble de

données d'apprentissage exigées avant de connaître les résultats et la convergence du processus d'apprentissage. Pour ce dernier, plusieurs stratégies avec différentes données doivent être initialement appliquées. Les données doivent contenir les connaissances nécessaires qui décrivent le comportement de l'embouti pour un ensemble de couples (entrée, sortie) finis. Une fois le modèle RNA a bien appris, il interpole les données selon une fonction implicite de la forme suivante :

$$t_{RNA} = g(r_0, r_{45}, r_{90}, n, k, \mu)$$

Cette fonction sera utilisée par la suite pour prédire l'épaisseur minimale dans l'embouti pour d'autres paramètres opératoires. Pendant l'opération de l'entraînement du réseau, les différentes valeurs associées aux paramètres du modèle neuronal sont décrites ci-dessous :

- Nombre total d'exemples d'apprentissage = 729
- Nombre total d'itérations = 1000
- Coefficient de moment $\alpha = 0.1$
- Taux d'apprentissage $\eta = 0.01$
- Nombre de nœuds dans la couche cachée NHidden=17

La courbe qui caractérise la réduction de l'erreur quadratique moyenne RMSE en fonction des itérations est représentée sur la figure 4.7. La valeur minimale de cette erreur est atteinte après 1000 itérations et qui correspond à 1.09%.

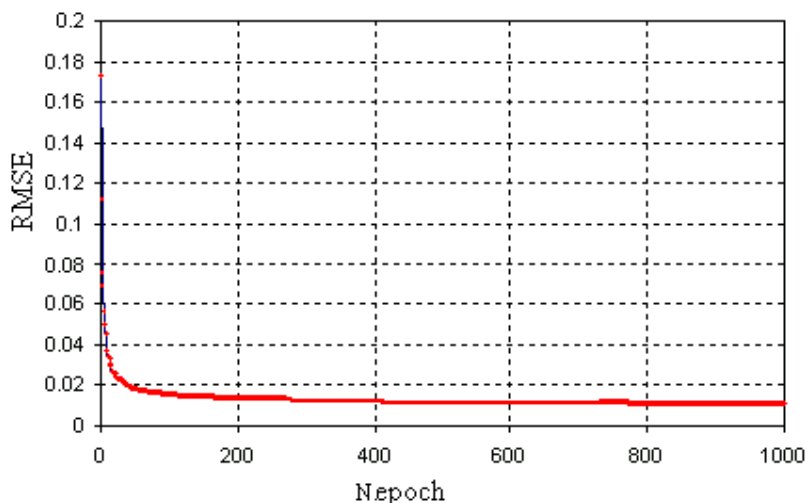


FIG. 4.7 – Erreur RMSE en fonction de nombre d'épochs

4.3.5 Résultats et interprétations

Une fois le modèle RNA satisfait l'erreur souhaitée, il sera ensuite examiné pour un ensemble de données non employées dans la phase d'apprentissage. Dans ce cas, nous avons utilisé 37 autres points de conception qui n'ont pas été servis dans l'ensemble des données d'apprentissage du modèle. Pour étudier la fiabilité de l'algorithme, en terme des sorties prédites nous avons exécuté 37 calculs EF correspondant aux mêmes points de conception. Les

réponses représentant la valeur minimale de l'épaisseur dans la pièce à la fin de l'opération d'emboutissage sont relevées. Une superposition des deux résultats numériques calculés par EF et prédites par le réseau de neurone en fonction de nombre des points de conception est présentée sur la figure 4.8.

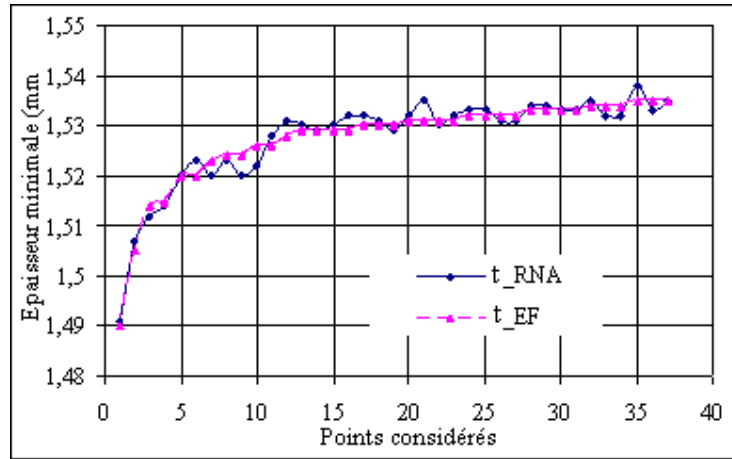


FIG. 4.8 – Variation de l'épaisseur minimale en fonction des points considérés

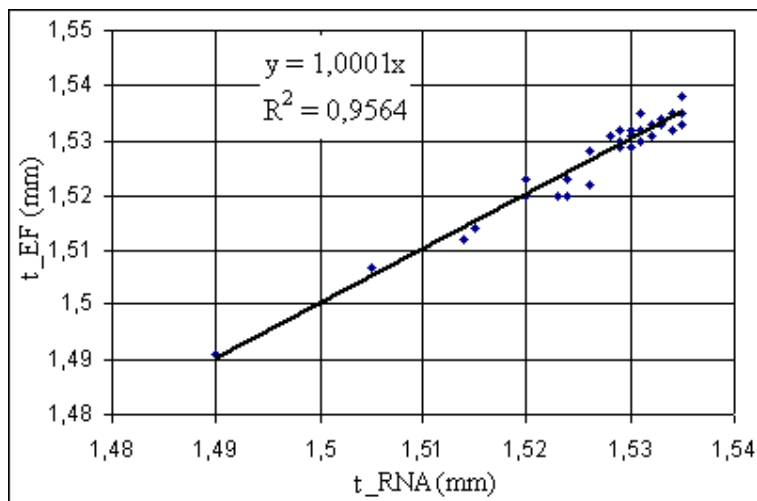


FIG. 4.9 – Epaisseur minimale calculée par le modèle EF en fonction de celle prédite par le modèle RNA

L'erreur relative maximale entre les deux résultats par rapport à celui calculé par EF reste inférieure à 1%. Ceci montre clairement une bonne concordance entre la réponse obtenue par le modèle RNA et celle obtenue par le modèle EF. Une autre façon de présenter cette comparaison entre les deux modèles est illustrée par la figure 4.9. Une régression linéaire

entre toutes les valeurs de l'épaisseur minimale permet de constater que tous les points sont dispersés autour d'une droite de pente proche de 1. Ceci pourrait être expliqué par la robustesse et l'efficacité de l'algorithme neuronal à la prédiction des résultats très proches à ceux calculés par EF.

D'après les figures 4.8 et 4.9, on peut dire que le modèle RNA construit est bien appris et qu'il est capable de générer d'autres réponses pour d'autres paramètres d'entrées différents de ceux qui sont appliqués pour la phase d'apprentissage. En terme de coût et de temps de calcul, on peut noter qu'un calcul par l'algorithme développé est plus rapide qu'une simulation par EF pour prédire une telle réponse. Si le temps nécessaire pour la réalisation d'un calcul EF dure 30 minutes alors que par application du modèle RNA, le temps de prédiction des réponses est de l'ordre de quelques secondes. Ceci montre que notre algorithme est un bon estimateur, capable de prévoir les résultats obtenus par EF dans un temps de calcul très court.

4.4 Identification du matériau par RNA

L'identification des lois de comportement est très utile pour les procédés de mise en forme par emboutissage. Contrairement aux méthodes classiques d'identification appelées méthodes paramétrées, on applique ici une méthode d'identification dite non paramétrée se basant sur l'algorithme présenté dans le troisième chapitre. Cette méthode ne s'intéresse plus à l'aspect physique des paramètres mis à l'identification. C'est un outil bien adapté pour la résolution des problèmes inverses complexes.

En optimisation, il arrive dans certains cas qu'il ne pas possible de définir une loi mathématique pour chercher les paramètres optimaux en fonction des paramètres d'excitation ou bien la fonction coût décrivant exactement le processus. Pour cela, on peut envisager d'obtenir un modèle RNA représentatif basé sur des données historiques entrées-sorties et/ou des expériences conçues.

Dans ce paragraphe, la procédure d'identification inverse présentée pour la détermination des paramètres du matériau, est illustrée dans la figure 4.10. Elle consiste à construire un RNA inverse permettant de fournir quelques paramètres d'entrées à partir des certaines réponses. La problématique est la suivante : Connaissant certaines sorties préalablement définies, peut-on identifier les valeurs des paramètres géométriques et/ou matérielles correspondantes d'une manière rapide et fiable ? Pour ce faire, trois cas seront étudiés dont chacun on a proposé un modèle RNA inverse donné par la figure 4.10.

1. Identification globale de tous les paramètres opératoires,
2. Identification des coefficients d'anisotropie,
3. Identification de la fonction d'érouissage.

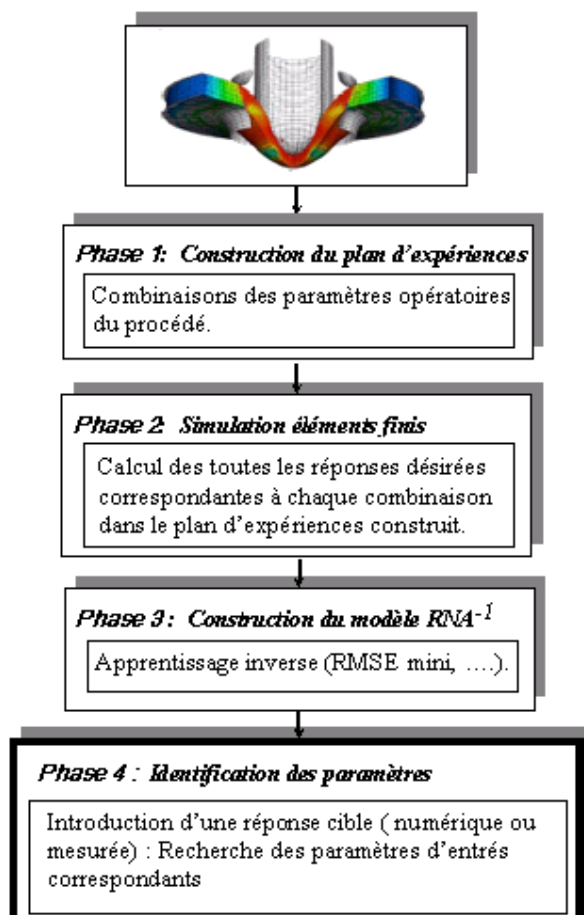


FIG. 4.10 – Stratégie d'identification par $(RNA)^{-1}$

4.4.1 Identification globale de tous les paramètres opératoires

Dans ce cas, on cherche à identifier tous les paramètres du procédé présentés dans le tableau 4.1 en connaissant l'épaisseur minimale dans l'embouti. Le modèle RNA proposé est présenté dans la figure 4.11.

Une fois les paramètres sont identifiés, ils seront tester à travers un autre calcul EF. Les deux réponses numériques identifiées et calculées seront enregistrées. Les résultats de comparaison sont présentés dans la figure 4.12. On peut remarquer que les paramètres identifiés à partir de ce modèle ne donnent pas une bonne estimation de l'épaisseur par comparaison avec l'épaisseur désirée. Nous pensons que cela est dû à la structure du modèle RNA elle-même. En effet, il est clair que ce modèle ne peut pas identifier correctement six paramètres à partir d'une seule information reçue à l'entrée du réseau. Pour cela, nous avons identifié dans le deuxième cas les paramètres liés à l'anisotropie qui sont au nombre de trois, connaissant les coefficients d'écroutissage et le coefficient de frottement.

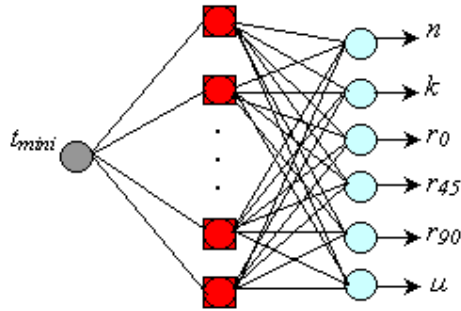


FIG. 4.11 – Modèle RNA proposé

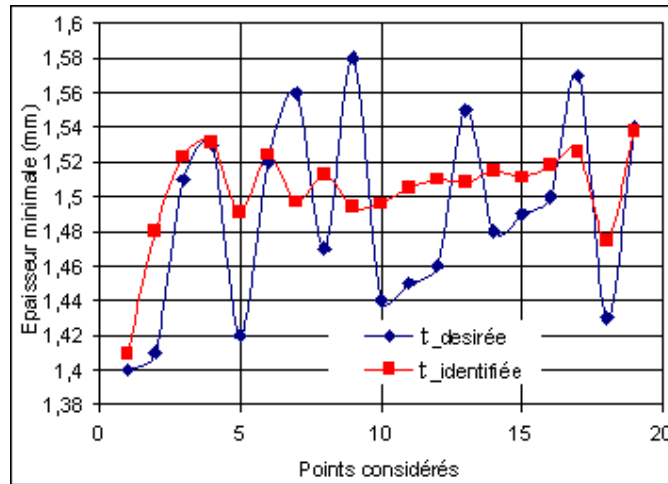


FIG. 4.12 – Résultats du premier cas d'identification

4.4.2 Identification des coefficients d'anisotropie

Dans ce cas, nous supposons que nous connaissons à priori les valeurs de k et n (courbe d'écoulement) ainsi que le coefficient de frottement entre outils qui est supposé égal à 0.2. L'épaisseur minimale dans l'embouti est déjà déterminée. Le modèle RNA correspondant à cette simulation est donné par la figure 4.13. Il est de type 4-17-3.

La courbe qui traduit la décroissance de l'erreur quadratique moyenne au cours des itérations est illustrée sur la figure 4.14. Pour les paramètres ajustés du modèle RNA inverse l'erreur était 5.3% obtenue après 5000 epochs d'apprentissage entre les sorties désirées et prédites. Dans un deuxième temps, le modèle employé pour identifier les coefficients de Lankford pour d'autres combinaisons des paramètres d'entrées. Pour valider ce modèle d'identification les paramètres identifiés sont introduits dans un calcul direct par EF. En comparant toujours l'épaisseur minimale obtenue par les paramètres prédits avec l'épaisseur désirée nous constatons d'après la figure 4.15, que dans ce cas les résultats obtenus sont beaucoup mieux

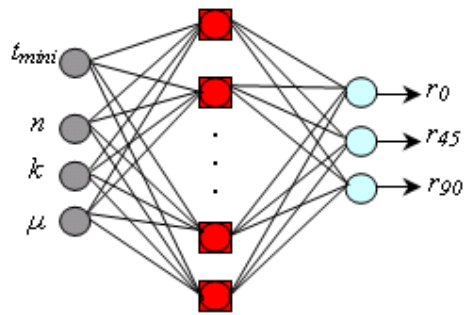


FIG. 4.13 – Modèle RNA proposé

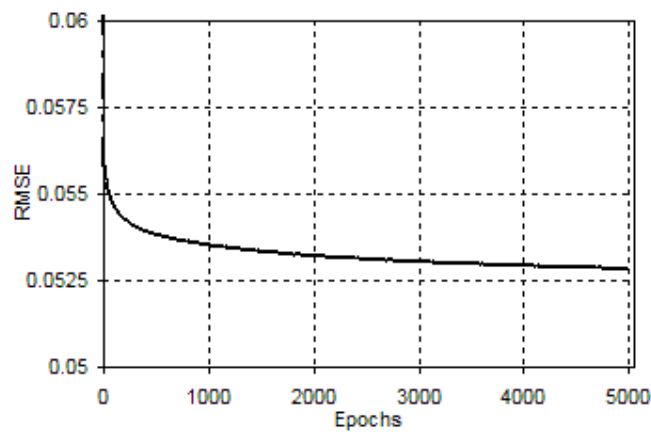


FIG. 4.14 – Résultats de l'évolution de l'erreur quadratique moyenne au cours des itérations

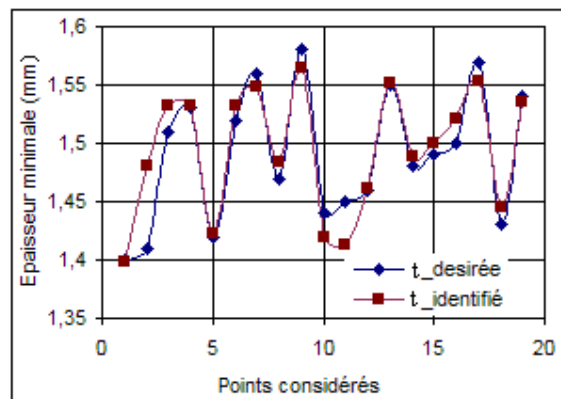


FIG. 4.15 – Résultats du deuxième cas d'identification

que ceux obtenus dans le premier cas. Il ressort de la même figure que le modèle RNA inverse est capable de prévoir les coefficients d'anisotropie pour une gamme des paramètres illustrés dans le tableau 4.1. L'erreur relative entre la réponse EF et la réponse RNA n'a pas excédé le 4%. La différence qui existe entre les deux réponses peut être remédiée par l'augmentation de nombre d'épochs pendant la phase d'apprentissage ou bien d'ajouter d'autres couches cachées.

4.4.3 Identification de la fonction d'écroutissage

La proposition d'un troisième cas d'identification est directement liée à la prédiction des paramètres d'écroutissage k et n . La structure du modèle correspondant est présentée par la figure 4.16. Le modèle proposé possède comme entrées les valeurs de t_{mini} , r_0 , r_{45} , r_{90} et μ et comme sorties k et n .

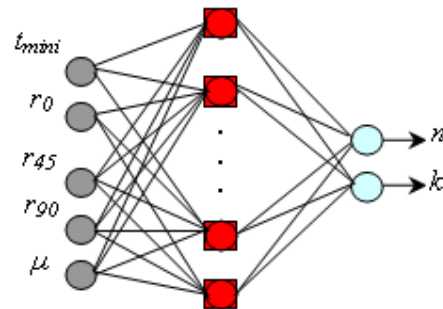


FIG. 4.16 – Modèle RNA proposé

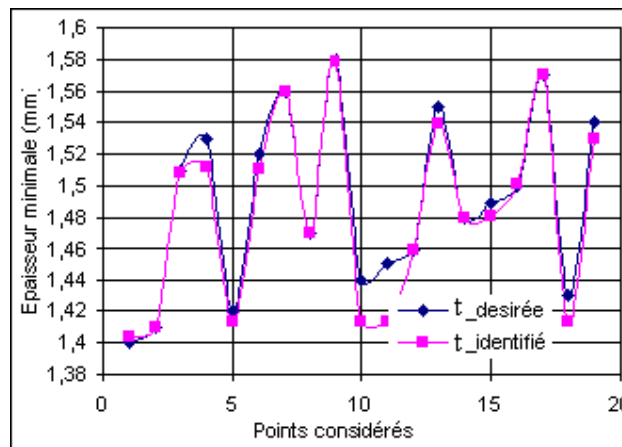


FIG. 4.17 – Résultats du troisième cas d'identification

La figure 4.17 montre bien que les résultats fournis, par cette technique d'identification inverse sont beaucoup plus meilleurs que ceux obtenus dans le premier et le deuxième cas.

Ceci peut être expliqué par le fait que le nombre des données d'entrées est plus important que le nombre des sorties cherchées. Un bon accord a été obtenu entre les réponses associées aux paramètres d'érouissage identifiés et celles souhaitées.

4.5 Optimisation du procédé d'emboutissage cylindrique

4.5.1 Introduction

Dans le cadre de l'optimisation, il est nécessaire de disposer de méthodes robustes de paramétrisation et d'incrémentation automatique des facteurs géométriques liés à la forme de la pièce qui dans notre étude est un embouti cylindrique. C'est dans cette optique qu'a été développé un programme sous Python qui permet de coupler le code de calcul par éléments finis ABAQUS et le programme de calcul par RNA. L'objectif de cette application est d'étudier l'influence de quelques paramètres géométriques liés au procédé par RNA, de prédire l'angle du retour élastique et de chercher les paramètres optimaux afin de minimiser le retour élastique. Les résultats obtenus seront comparés avec les résultats obtenus par la méthode des éléments finis et analysés par la méthode de surface de réponse.

4.5.2 Problématique liée au phénomène du retour élastique

La formabilité du matériau a toujours été considérée comme l'un des paramètres qui affecte la qualité du produit fini. En effet, les contraintes résiduelles introduites par laminage sont indésirables et elles jouent un rôle dans la géométrie après mise en forme. Le retour élastique est l'un des phénomènes les plus étudiés dans les procédés de mise en forme, particulièrement, l'emboutissage. Ce phénomène est dû à la redistribution des contraintes dans le matériau après recul des outils [Akr1984], [Bah2006-b]. Après l'opération de retrait des outillages, il y a donc des risques que la pièce après retour élastique ne soit pas dans les limites admissibles de tolérance et par conséquent inutilisable. Le niveau de retour élastique dépend de plusieurs paramètres citons à titre d'exemples : Les propriétés du matériau, la géométrie des outils, la géométrie de pièce et les paramètres du procédé comme le coefficient de frottement et le jeu entre les outils [Bru2005].

Donc, la maîtrise de la géométrie des pièces finies a toujours posé des problèmes dans le milieu industriel. Le phénomène de retour élastique est associé dans bien des cas aux propriétés du matériau. Cependant, le contrôle de procédé est plus aisé en agissant sur les paramètres de celui-ci que sur les propriétés du matériau [Mka2003]. Pour cela, dans la suite seulement des paramètres géométriques du procédé comme les rayons de matrice et du poinçon ainsi que le jeu entre les outils seront pris en compte.

4.5.3 Modélisation numérique

Dans cette étude le même exemple d'emboutissage présenté par la figure 4.2 a été considéré. Les valeurs numériques des paramètres du matériau sont fixées à des valeurs constantes et qui sont définies dans le tableau 4.3. Les variables de l'étude sont les paramètres géométriques du procédé reliés aux rayons du poinçon et de la matrice et du jeu résiduel. Elles sont présentées

par la figure 4.18. L'effort de serrage appliqué par l'intermédiaire du serre flan est maintenu à une valeur constante durant la simulation et qui vaut 10KN. Il permet d'une part d'établir le serrage de la pièce contre la matrice et le serre-flan et d'assurer le contact d'autre part, il ne favorise pas l'apparition des plis dans l'embouti.

Dans toute l'analyse numérique de ce procédé d'emboutissage, nous avons pris en compte que seulement un quart du modèle (Figure 4.18). Les outillages sont modélisés comme étant des corps analytiques rigides alors que le flan représente un corps déformable. La pièce à déformer a été maillée avec des éléments coques "Shell elements" de type S3R et S4R. Le choix de ce type d'élément s'est fait pour plusieurs raisons : Il convient pour des problèmes présentant une grande déformation. Il est aussi bien adapté pour la prédiction du retour élastique étant donnée qu'il n'est pas affecté par la variation de l'épaisseur du matériau [Aba2001]. Les analyses de chargement et de déchargement sont effectuées avec un pilotage en déplacement. Les simulations numériques ont été effectuées en statique implicite sur ABAQUS/STANDARD en négligeant tous les effets dynamiques. Le frottement a été modélisé suivant la loi de Coulomb avec un coefficient de 0.1 aux interfaces flan-outils.

E	μ	σ_0 (Mpa)	k (Mpa)	n
193000	0.3	238	1172	0.4

TAB. 4.3 – Paramètres utilisés

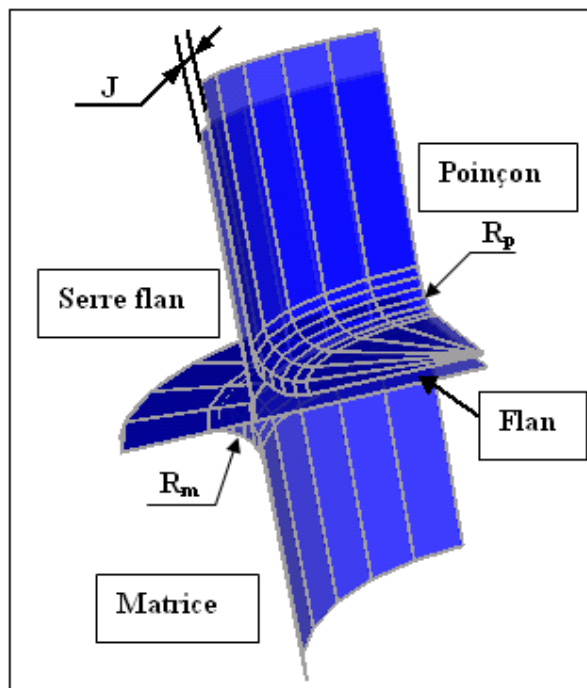


FIG. 4.18 – Modèle numérique du procédé

4.5.4 Plan d'expériences

Bien que la mise en forme des métaux est un procédé qui met en considération plusieurs paramètres, nous nous sommes restreint à l'étude des paramètres géométriques en ne considérant que l'influence des paramètres les plus significatifs du procédé d'emboutissage et qui sont : Le rayon de poinçon R_p , le rayon de la matrice R_m et le jeu résiduel entre le poinçon et la matrice J [Bru2005], [Bah2006-b].

La fonction objectif que nous considérons ici, est l'angle de déviation après retour élastique de la pièce. Cet angle sera prédit par éléments finis en vue de calculer la réponse de chaque combinaison des paramètres considérés. Le tableau 4.4 récapitule les résultats du plan d'expériences obtenus pour chaque jeu des paramètres de rayon de la matrice et de rayon du poinçon avec un jeu nul. Le tableau 4.5, quant à lui il représente une matrice d'expériences obtenue pour chaque combinaison de rayon de la matrice et de jeu résiduel entre le poinçon et le flan. Le rayon poinçon est fixé à une valeur de 8mm.

R_m (mm)	R_p (mm)					
	8	10	12	14	16	18
8	3.263	4.143	5.173	6.721	8.669	10.995
10	4.274	5.350	6.750	8.630	10.663	12.784
12	5.838	7.405	9.016	10.896	13.095	15.437
14	8.086	9.669	11.491	13.589	15.899	18.396
16	10.536	12.386	14.409	16.544	18.808	21.366
18	13.312	15.245	17.289	19.442	21.827	24.139

TAB. 4.4 – Plan d'expériences du retour élastique en (°) pour différents rayons de matrice et de poinçon (J=0mm)

R_m (mm)	J(mm)				
	0	0.2	0.4	0.6	0.8
8	3.309	3.515	3.703	3.789	3.971
10	4.245	4.485	4.695	4.951	5.196
12	5.724	5.991	6.297	6.540	6.834
14	8.091	8.411	8.731	9.083	9.390
16	10.497	10.874	11.243	11.585	11.959

TAB. 4.5 – Plan d'expériences du retour élastique en (°) pour différents jeu et rayons de matrice $R_p=8$ mm

4.5.5 Modélisation neuronale

L'objectif ici est de montrer la potentialité des modèles RNA à prédire le retour élastique pendant l'opération d'emboutissage d'une façon rapide et fiable. Dans un premier temps,

nous avons fait un calcul direct pour prédire le retour élastique dans l'embouti en fonction des paramètres du procédé : Rayons de poinçon et de la matrice. Suite à cette étude, un modèle RNA a été établi comprenant deux neurones dans la couche d'entrée, cinq neurones dans la couche cachée et un neurone dans la couche de sortie. Les résultats obtenus seront comparés à ceux obtenus par une analyse EF. Le Modèle numérique a été entraîné avec les résultats présentés dans le tableau 4.4 excepte les combinaisons correspondant aux valeurs de $R_p=14\text{mm}$ et $R_m=12\text{mm}$. Ces derniers sont utilisés pour une phase de validation et de prédiction. Les courbes qui représentent l'évolution de l'erreur quadratique moyenne au cours des itérations sont présentées dans la figure 4.19.

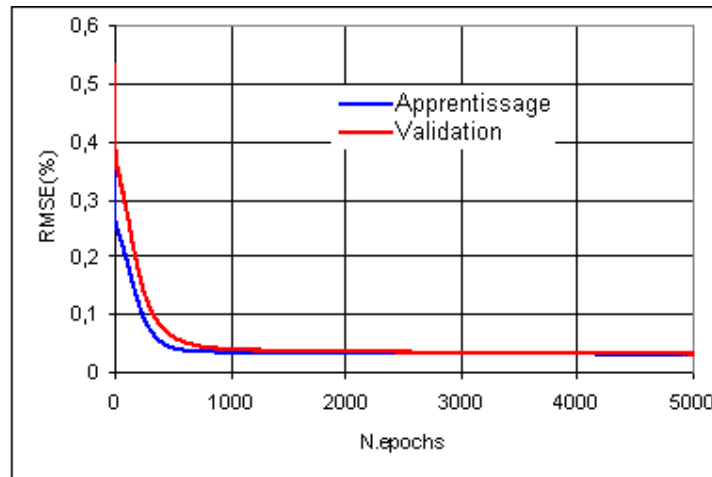


FIG. 4.19 – Erreur quadratique au cours de la phase d'apprentissage

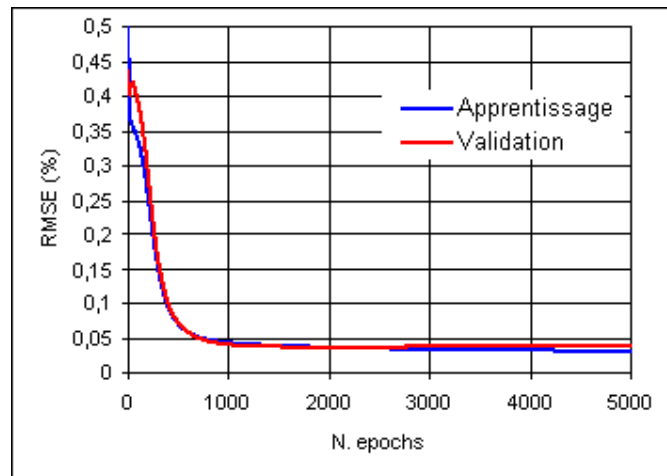


FIG. 4.20 – Evolution de l'erreur quadratique moyenne au cours des itérations

Dans un second temps, nous avons prédit l'angle du retour élastique en fonction du rayon de matrice et du jeu poinçon-pièce. Dans cette application, la même architecture de réseau que celle de l'étude précédente est employée. La figure 4.20 présente la réduction de l'erreur quadratique moyenne au cours de la phase d'apprentissage en fonction des itérations. Une fois, validé le modèle RNA sera utilisé pour prédire le retour élastique pour n'importe quelles valeurs des paramètres.

4.5.6 Influence du rayon de la matrice

Plusieurs expériences numériques par RNA et EF ont été conduites en faisant varier les paramètres du procédé. La courbe présentée dans la figure 4.21 traduit la variation de l'angle du retour élastique en fonction du rayon de la matrice pour des valeurs constantes de rayon de poinçon et du jeu et qui sont respectivement égales à 14mm et 0mm. Comme il peut être constaté, que les valeurs de cette erreur angulaire calculées par EF et prédites sont d'autant plus importantes que le rayon de la matrice augmente. Une comparaison entre les réponses numériques a permis de montrer que les deux courbes sont presque confondues, d'où un bon accord entre les deux approches.

L'erreur relative entre les deux modèles éléments finis et par réseau de neurones est exprimée par la relation ci-dessous. La valeur maximale de cette erreur n'a pas excédé le 1%.

$$Err\% = \frac{\theta^{EF} - \theta^{RNA}}{\theta^{EF}} 100$$

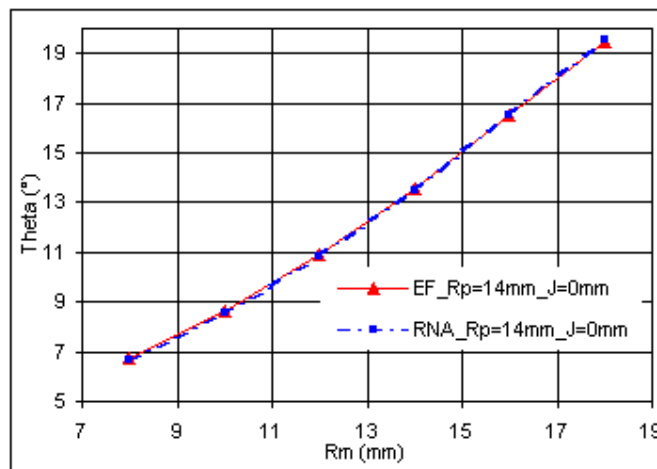


FIG. 4.21 – Influence du rayon de la matrice sur l'angle du retour élastique

4.5.7 Influence du rayon du poinçon

L'influence de rayon du poinçon sur le retour élastique a été également étudiée pour des valeurs constantes du rayon de la matrice et du jeu. La figure 4.22 traduit l'erreur géométrique

du retour élastique en fonction de rayon du poinçon obtenu par les deux approches. Sur la figure 4.22, nous pouvons voir qu'une augmentation du rayon de poinçon favorise l'augmentation de l'angle de retour élastique. L'analyse de l'évolution des deux courbes permet de constater qu'ils ont la même tendance et que globalement la réponse numérique prédite est en cohérence avec celle obtenue par un calcul EF.

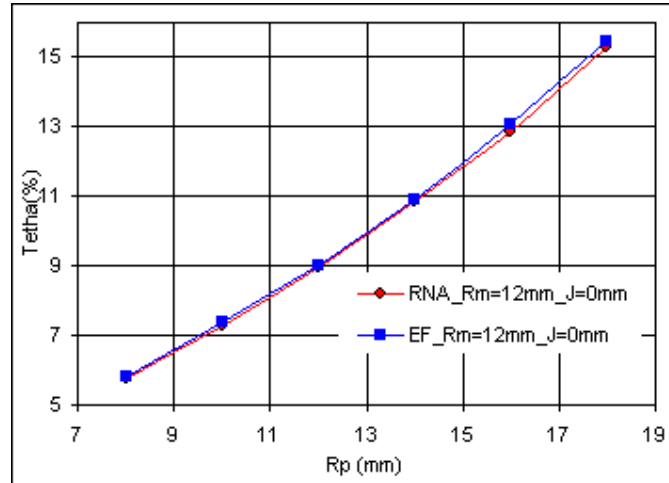


FIG. 4.22 – Influence du rayon de poinçon sur l'angle du retour élastique

4.5.8 Influence du jeu

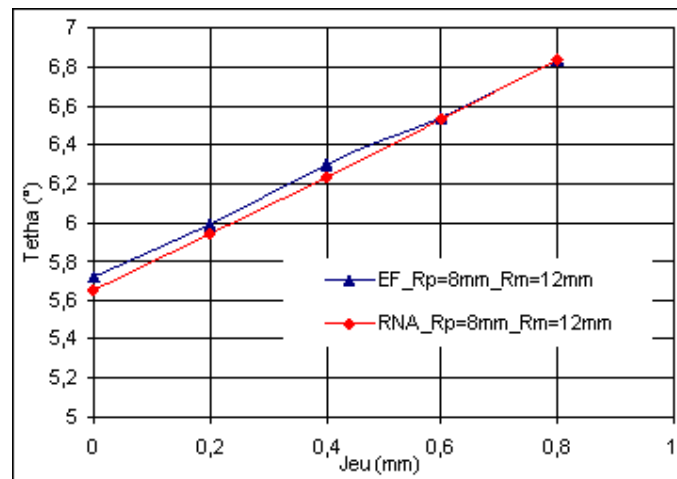


FIG. 4.23 – Influence du jeu sur le retour élastique

D'une façon similaire, Nous avons étudié par le modèle RNA l'influence du jeu entre la tôle et le poinçon sur la prédiction de l'angle de retour élastique. Les évolutions de deux

réponses, prédite et calculée sont illustrées sur la figure 4.23. La comparaison des résultats montre un bon accord entre les deux. L'erreur maximale relative déduite par rapport aux résultats éléments finis est de l'ordre de 1.5%.

4.5.9 Optimisation du retour élastique

Dans cette partie, on veut minimiser le retour élastique à la fin de l'opération de l'emboutissage en fonction du rayon de la matrice du jeu et du rayon de poinçon. Pour bien analyser et comparer les résultats d'optimisation en vue de déterminer les facteurs significatifs du problème, on choisit de fixer à chaque fois l'un de ces trois paramètres à une valeur constante et nous traçons la surface de réponse en fonction des variations des deux autres. Les résultats obtenus sont construits à partir des réponses prédites par le modèle réseau de neurones. Pour éviter une trop grande influence de l'un d'eux, les différents paramètres seront normalisés. La fonction objectif est encore rapportée à l'angle désirée de l'embouti ($\theta=90^\circ$).

$$\bar{R}_m = \frac{R_m}{t}, \quad \bar{R}_p = \frac{R_p}{t}, \quad \bar{J} = \frac{J}{t}$$

$$\bar{\theta} = \frac{\theta}{90} 100$$

Avec t représente l'épaisseur de la tôle.

Pour générer la surface de réponse, nous avons choisi une approximation polynômiale quadratique.

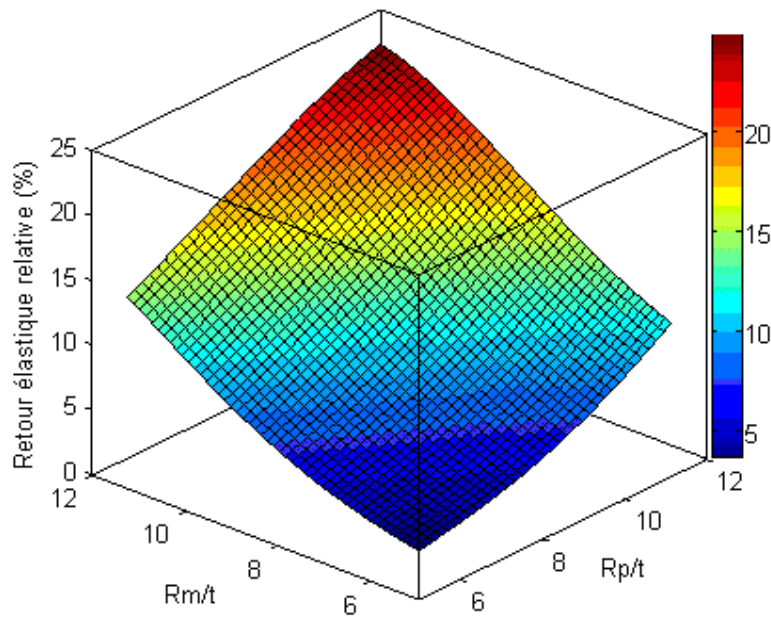


FIG. 4.24 – Surface de réponse obtenue dans le cas où $J=0$ mm

La figure 4.24 représente la surface de réponse dans le cas où le jeu entre le poinçon et la pièce est égal à 0mm. Dans ce cas, nous constatons que le retour élastique évolue avec l'augmentation de rayon de poinçon et celui de la matrice. Ceci paraît physiquement logique puisqu'il y'aura moins d'écroutissage et par conséquent moins de déformation plastique quand on utilise des grandes valeurs de ces paramètres. Les courbes de niveaux présentés dans

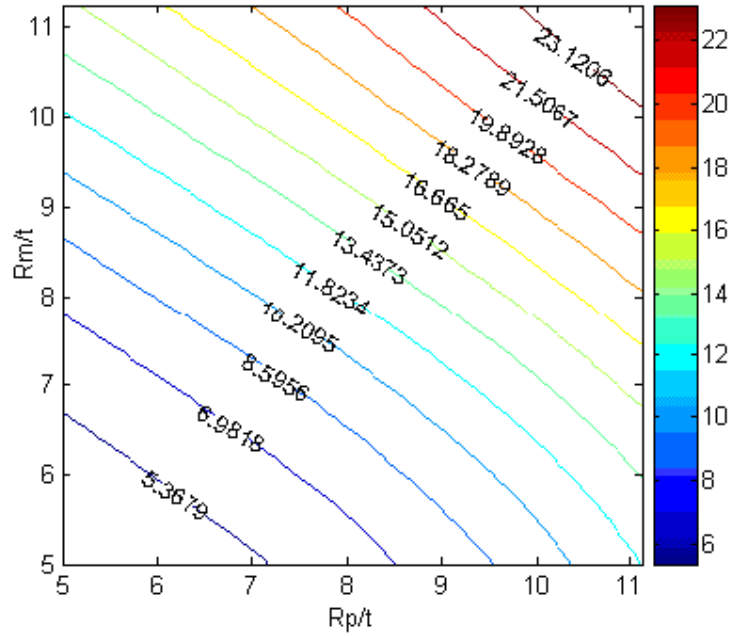


FIG. 4.25 – Courbes de niveaux obtenues dans le cas où $J=0$ mm

la figure 4.25 permettent de juger sur la sensibilité de retour élastique aux variations des deux paramètres géométriques considérés. D'après cette figure, on remarque que les deux paramètres ont presque la même influence sur la variation de cette réponse. Le tableau 4.6 récapitule les résultats de minimisation. Eclairé de l'étude précédente les deux paramètres de

Valeurs optimales	$(\frac{R_p}{t})_{opt}$	$(\frac{R_m}{t})_{opt}$	$\theta_{las.Relative}$
RNA	5	5	3.75

TAB. 4.6 – Optimisation du retour élastique pour $J=0$ mm

procédé présentent la même influence. Nous avons choisit de représenter la deuxième fonction objectif en fonction des valeurs relatives du jeu et de rayon de la matrice en conservant une valeur constante du rayon de poinçon correspondante à sa valeur optimale ($R_p=8$ mm).

Nous remarquons d'après la figure 4.26, que la surface de réponse du retour élastique présente deux allures totalement différentes. Elle varie de façon non linéaire avec le rayon relatif de la matrice. Le retour élastique évolue légèrement avec le jeu relatif entre le poinçon

et la tôle, et son comportement avec ce paramètre est sensiblement linéaire. Ce qui montre que le rayon de la matrice est le facteur le plus prépondérant sur la mise en forme. Les valeurs maximales de retour élastiques sont obtenues pour les grandes valeurs des paramètres du procédé. En revanche, les valeurs minimales se trouvent à des faibles valeurs de rayon de la matrice et du jeu poinçon-tôle. Ceci pourrait être expliqué par un effet très important d'écroutissage qui se manifestent lors de variation des ces paramètres.

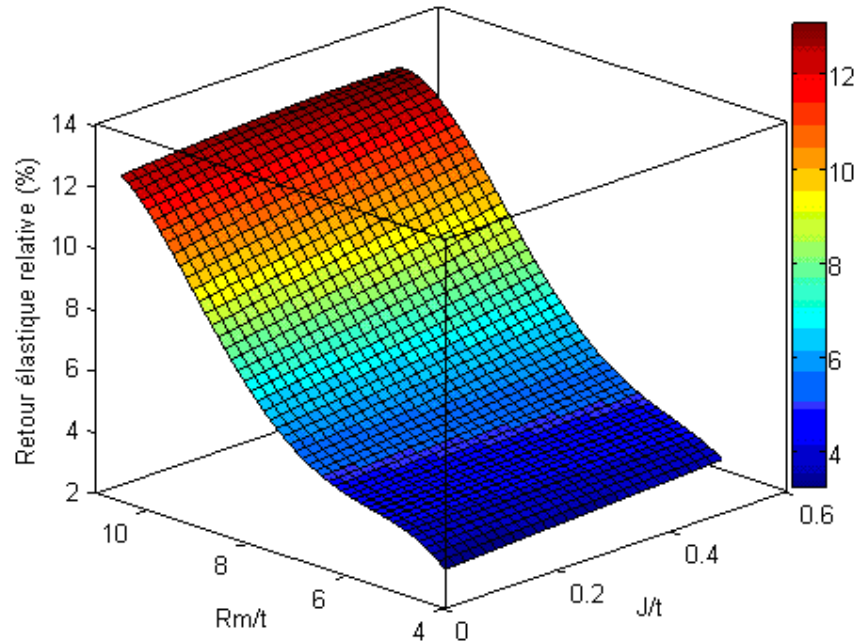


FIG. 4.26 – Evolution RNA du retour élastique en fonction des paramètres du procédé sous forme de surface de réponse

L'analyse de surface de réponse présentée dans la figure 4.26 permet de constater que le modèle RNA adopté permet une bonne prédiction de l'angle du retour élastique. Les paramètres optimaux conduisant à une minimisation de cette réponse sont récapitulés dans le tableau 4.7.

Valeurs optimales	$(\frac{J}{t})_{opt}$	$(\frac{Rm}{t})_{opt}$	$\theta_{las.Relative}$
RNA	0	5	3.67

TAB. 4.7 – Optimisation du retour élastique pour $R_p=8\text{mm}$

Le retour élastique est ensuite représenté sous forme des courbes de niveau (figure 4.27). Ils permettent de bien cerner la zone optimale contenant la valeur minimale de cette réponse dans le domaine de conception. D'après cette figure il peut être aussi conclu que le jeu a peu

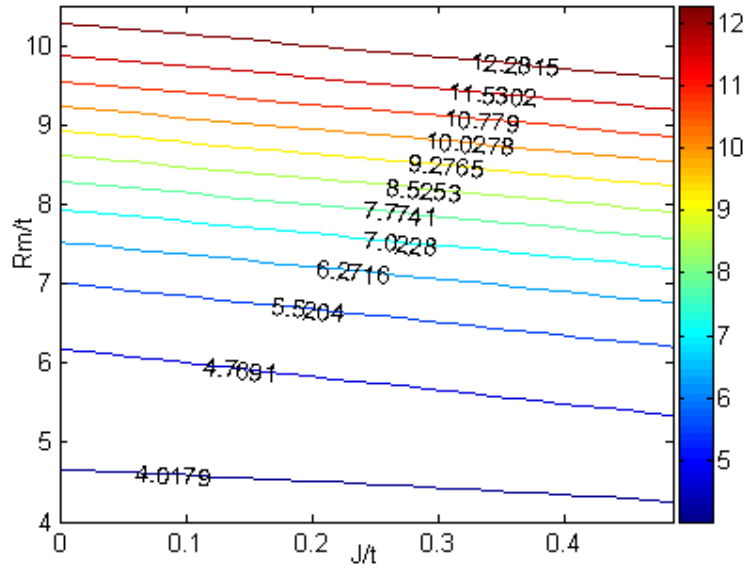


FIG. 4.27 – Evolution RNA du retour élastique en fonction des paramètres du procédé sous forme des courbes de niveau

d'influence sur le retour élastique par comparaison avec l'effet de rayon de la matrice. Dans ce cas le retour élastique suit une variation presque linéaire avec ce paramètre.

4.6 Conclusion

Dans ce chapitre, nous avons mis en application les réseaux de neurones artificiels pour la modélisation et l'optimisation du procédé d'emboutissage. Différentes applications ont été abordées afin de valider le programme développé ainsi que les routines de couplage EF et RNA. Les études menées ont concerné la modélisation du procédé d'emboutissage. Ceci a été fait à partir de la conception d'un réseau pour un calcul direct des différentes nuances de matériau. Ensuite, un calcul inverse a été établi à partir du même modèle conçu dans le but d'identifier les paramètres du matériau à partir des quelques données antérieurement connues. Enfin des techniques d'optimisation directes basées l'application des plans d'expériences et des surfaces de réponse dont les résultats sont issus d'un modèle RNA ont été utilisées. Cela permet une représentation globale de retour élastique en fonction des paramètres du procédé. Pour les trois applications, nous avons abouti à des bons résultats. Ces derniers permettent de conclure que les RNA sont bien adéquats pour la prédiction du comportement des pièces mises en forme et l'optimisation de leur procédé.

Chapitre 5

Identification des paramètres du matériau AISI 304 par RNA à partir du procédé d'hydroformage

5.1 Introduction

Dans ce chapitre nous présentons l'application des techniques des RNA pour l'identification des paramètres du modèle de comportement élastoplastique avec le critère quadratique de Hill. Ce problème mathématiquement appelé problème inverse peut être considéré comme un problème d'optimisation dont la fonction objectif est une norme de l'écart entre les résultats expérimentaux et les résultats numériques. Les variables d'optimisation sont les paramètres du matériau intégrés dans le modèle de comportement. Le programme développé dans le troisième chapitre sera exploité pour cette application. Le procédé que nous mettons en application pour l'identification des paramètres matériels est l'hydroformage. Ce dernier est représenté par un essai de gonflement hydraulique d'un flan circulaire. L'avantage de cet essai réside dans sa réponse qui ne fait intervenir que les propriétés intrinsèques du matériau en s'affranchissant presque totalement des problèmes de contact et de frottement. Il sera exploité pour établir à la fois une courbe d'érouissage et déterminer les coefficients de HILL dans notre application.

Comparé à l'essai de traction habituellement utilisé pour la caractérisation de l'érouissage d'un matériau, l'essai de gonflement hydraulique permet d'atteindre des déformations plus larges. Ce qui permet de palier au problème d'extrapolation des courbes de traction aux grandes déformations qui peut s'éloigner du comportement physique du matériau et compromettre ainsi des erreurs sur la prédiction. Bien que l'essai de traction reste pour plusieurs matériaux une référence, il n'est plus suffisant pour identifier le comportement des modèles de plus en plus raffinés. Donc, des informations additionnelles doivent être fournies à partir d'autres essais [For2002],[Ben2005].

Etant donnée l'écart obtenu entre la réponse expérimentale et numérique, nous proposons une stratégie inverse d'identification basée sur deux techniques différentes : Une méthode inverse classique basée sur le calcul du gradient de l'erreur et une méthode inverse basée

sur les RNA. Nous mettons en valeur la méthode inverse par RNA pour sa rapidité et son efficacité comparée aux méthodes d'identification classiques. Pour exhiber l'apport de cette technique et sa fiabilité nous avons procédé comme suit :

Dans un deuxième paragraphe, l'essai d'hydroformage ainsi que les résultats expérimentaux obtenus seront présentés. Dans le troisième paragraphe, nous présentons les résultats d'une première simulation par éléments finis à partir de laquelle nous mettons en considération l'écart enregistré entre les résultats expérimentaux et celles numériques. Le quatrième paragraphe traduit une démarche d'identification inverse par une méthode classique en vue de caler les deux réponses déterminées par les deux approches. Au cinquième paragraphe, l'algorithme développé basée sur les RNA a été appliqué pour identifier les paramètres du modèle. Une comparaison entre les deux techniques est présentée dans un sixième paragraphe. Enfin ce chapitre est achevé par des conclusions.

5.2 Essai d'hydroformage

5.2.1 Principe

C'est un essai de mise en forme réalisé au sein de laboratoire LGM-ENIM (figure 5.1). Il est utilisé pour la caractérisation des matériaux par expansion bi-axiale. Il permet de déterminer le comportement mécanique d'un matériau en grandes déformations sous un état bi-axial de déformation. Le dispositif est composé d'une matrice sur laquelle repose l'éprouvette, d'un serre-flan et des deux capteurs de déplacement et de pression. Le serre-flan est maintenu serré sur la platine par l'intermédiaire de plusieurs vis à couple constant. Un jonc est ajouté sur toute la périphérie du serre-flan de façon à supprimer l'écoulement de la matière et réaliser ainsi un encastrement.

Les éprouvettes utilisées sont des flans circulaires de diamètre 133 mm. Le gonflement de la tôle est obtenu par l'application d'une pression d'huile exercée par une pompe manuelle et mesurée par un capteur de pression. Le suivi de la forme du flan au cours de l'essai est obtenu par un capteur de déplacement qui est placé au centre de l'éprouvette.

Dans cet essai les déformations en expansion sont prédominantes. L'essai met en évidence l'importance des propriétés du matériau pour ces déformations, et la faible importance des paramètres géométriques.

Notations

a : Rayon intérieur de la matrice (45.5mm),
 r_0 : Position initiale d'un point matériel (sur une méridienne),
 u : Hauteur au pôle,
 e : Epaisseur en un point à l'état déformé,
 (r, z) : Position à l'état déformé d'un point matériel,
 p : Pression exercée,
 ρ : Rayon de courbure de la méridienne en un point,

σ_m, σ_c : Contraintes méridienne et circonférentielle,
 $\epsilon_m, \epsilon_c, \epsilon_n$: Déformation plastiques méridienne, circonférentielle et normale,
 $D_0 = 91 \text{ mm}$: Diamètre extérieur de la matrice (jusqu'au jonc de retenue),
 $R = 6 \text{ mm}$: Rayon de congé de la matrice,
 $e_0 = 0.8 \text{ mm}$: Epaisseur initiale de l'éprouvette,
 F_c : Force de serrage.

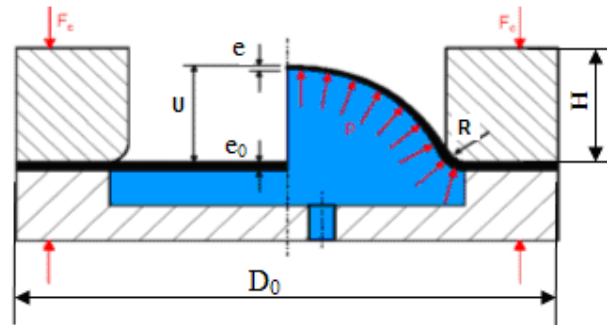


FIG. 5.1 – Principe de l'essai de gonflement hydraulique

5.2.2 Préparation des éprouvettes

La préparation des éprouvettes à partir d'une tôle vierge exige des précautions répondant aux spécificités de l'essai et à la forme géométrique du dispositif de chargement. Les étapes de préparation sont présentées sur la figure 5.2.

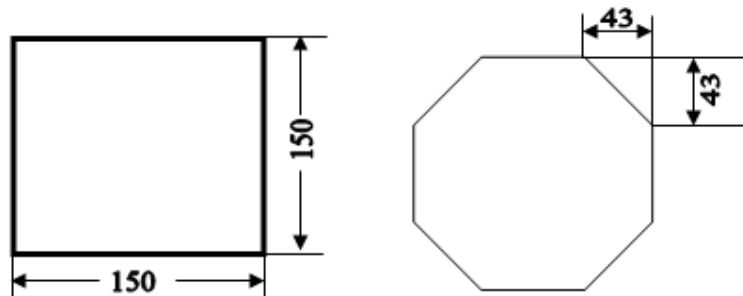


FIG. 5.2 – Découpage de la tôle en forme d'éprouvette

Une fois découpées, les éprouvettes sont transférées au poste d'électrolyse pour effectuer le dépôt de grilles électrolytiques. Le système de dépôt électrolytique de grilles est composé d'un générateur de courant, un rouleau, de grilles, un feutre, un électrolyte et un neutraliseur. Les grilles sont des motifs réguliers formés de cercles adjacents ou entrelacés déposées sur les éprouvettes avant chargement et permettent après la phase de chargement d'évaluer les

déformations subies par l'échantillon. La figure 5.3 présente le principe de mise en place des grilles.

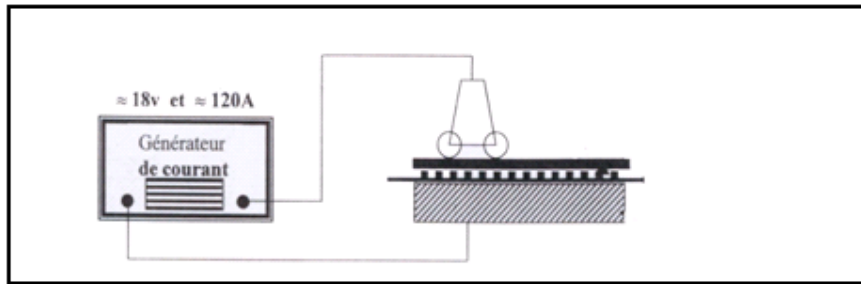


FIG. 5.3 – Dispositif de mise en place des grilles

La fixation des éprouvettes dans la cellule de gonflement hydraulique se fait par l'intermédiaire d'une matrice. Pour assurer une meilleure étanchéité entre l'éprouvette et la matrice, il est nécessaire d'imprimer sur l'éprouvette un jonc de retenu d'environ 5mm de profondeur. La figure 5.4 illustre le dispositif d'emboutissage des joncs de retenu.

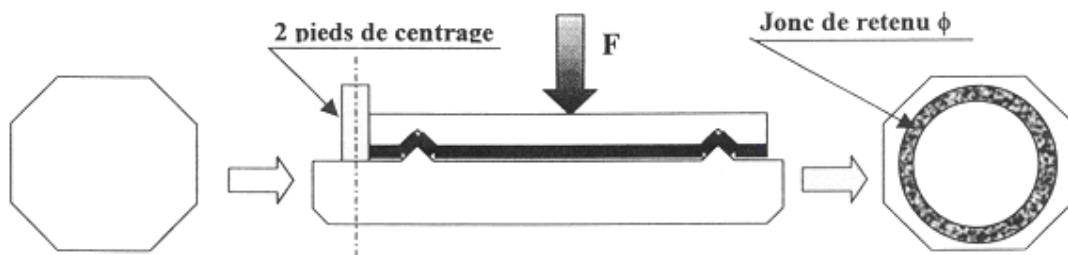


FIG. 5.4 – Outil d'emboutissage des joncs de retenu

5.2.3 Résultats expérimentaux

Le domaine de chargement envisagé est une fonction de la nature du matériau et des caractéristiques dimensionnelles des éprouvettes. Le dispositif permet en effet d'atteindre des pressions conduisant à l'apparition des striction localisées. Cette instabilité est souvent suivie par l'apparition des fissures. Par ailleurs, pour une pression donnée appliquée sur l'éprouvette, il sera possible de récupérer en un point donné les mesures suivantes :

- La hauteur,
- Les déformations,
- L'épaisseur.

Le type de sollicitation est essentiellement conditionné par la forme de la matrice et de l'anisotropie du matériau. Pour tenir compte de ces deux paramètres nous avons utilisé deux types de matrice, circulaire et elliptique (figure 5.5).

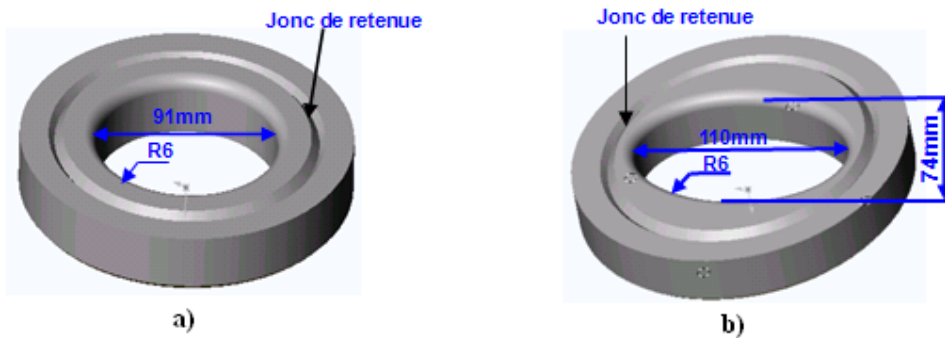


FIG. 5.5 – a- Matrice circulaire, b- Matrice elliptique

Les essais sont réalisés sur une cellule de gonflement hydraulique instrumentée. Elle est équipée de deux capteurs : L'un pour la mesure de pression et l'autre pour la mesure de déplacement au pôle (figure 5.6).

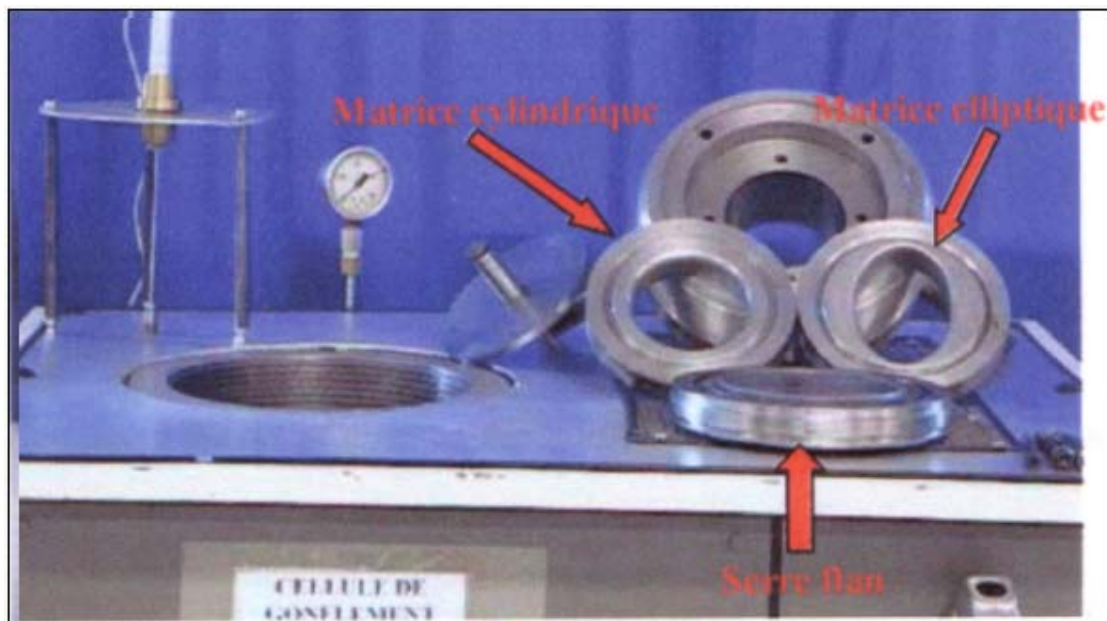


FIG. 5.6 – Cellule de gonflement hydraulique munie de tous les équipements

L'étude expérimentale menée au laboratoire LGM-ENIM nous fournit les mesures suivantes [Gah2006] :

- La hauteur au pôle en fonction de la pression : La mesure de la hauteur au pôle est effectuée par un comparateur de déplacement. Le niveau zéro de la mesure correspond à l'état non chargé de l'éprouvette. Dans cette configuration, la surface active est assimilée à un plan. La hauteur au pôle est relevée au centre de la partie active pour une pression donnée. La mesure est effectuée en même temps que l'éprouvette est chargée. Les courbes représentatives de la variation de la pression en fonction de la hauteur au pôle pour les deux matrices circulaire et elliptique sont données par les figures 5.7.

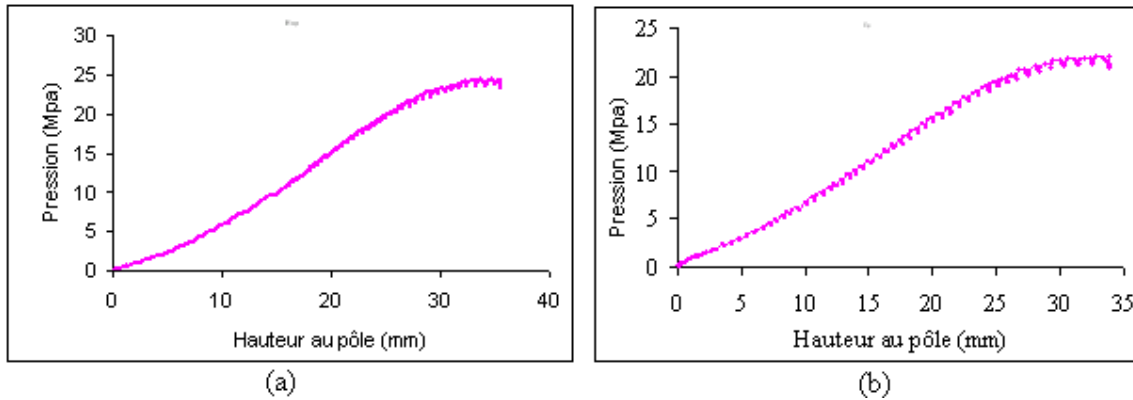


FIG. 5.7 – Pression en fonction de la hauteur au pôle : (a) Cas d'une matrice circulaire, (b) Cas d'une matrice elliptique

- Présentation du profil de la déformée (Hauteur en fonction de la position radiale) pour une pression de 160 bars (Figure 5.8).

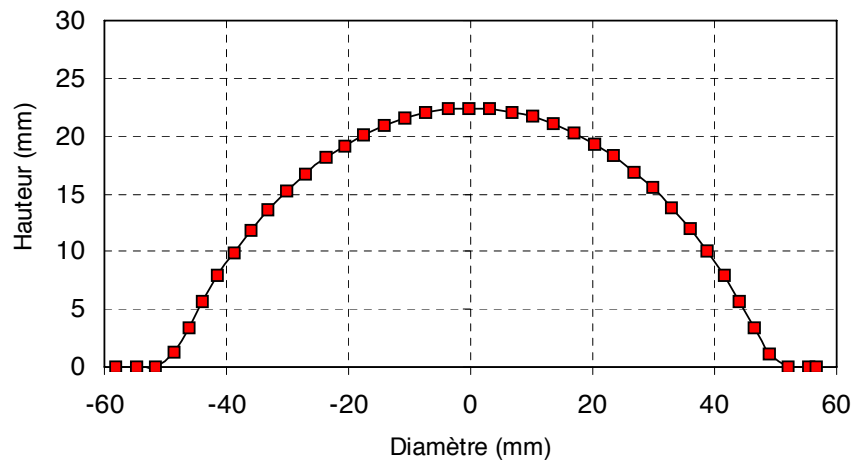


FIG. 5.8 – Profil de la pièce après mise en forme pour une pression de 160 bars : Cas d'une matrice circulaire

- La variation de l'épaisseur suivant le profil de l'éprouvette pour une pression de 160 bars : L'évolution de l'épaisseur correspond à une pression donnée, n'est réalisée qu'après démontage de l'éprouvette. Cette procédure expérimentale est exigée par le dispositif de visualisation des grilles déformées utilisant un projecteur de profil. La figure suivante nous montre l'évolution de l'épaisseur le long d'un méridien de l'éprouvette.

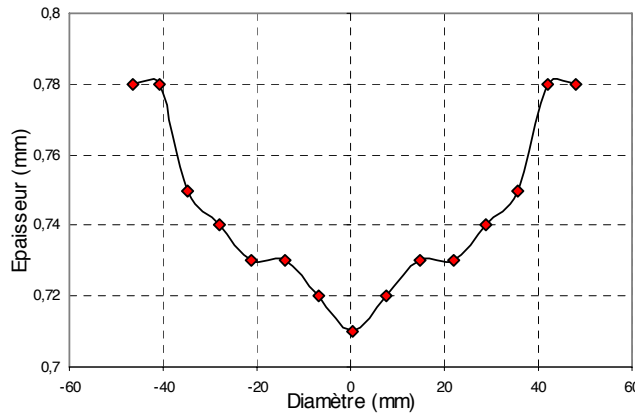


FIG. 5.9 – Epaisseur le long d'un méridien pour une pression de 160 bars : Cas d'une matrice circulaire

5.3 Analyse par éléments finis de l'essai

5.3.1 Modélisation et simulation

On présente dans ce paragraphe la simulation numérique de l'essai. Les résultats issus de cette simulation seront comparés aux résultats expérimentaux. Les données nécessaires pour effectuer les calculs numériques sont de plusieurs types : La géométries et les conditions aux limites, les caractéristiques mécaniques du matériau et la définition des chargements appliquées.

Les différents paramètres pris en compte dans nos calculs seront explicités dans les paragraphes qui suivent. Pour simuler l'essai d'hydroformage, nous avons utilisé la méthode des éléments finis moyennant le code EF ABAQUS/STANDARD. Pour cela, les hypothèses suivantes ont été considérées :

- Le modèle géométrique adopté est une plaque mince circulaire de diamètre 133mm. Elle est maillée par des éléments de type Shell quadrangulaires à 4 nœuds (S4R) en deux domaines : Une attention particulière a été faite au niveau de la partie centrale qui a été maillée en adoptant un maillage fin tandis que la partie restante nous avons utilisé globalement un maillage grossier (figure 5.10).
- La loi de comportement suit un modèle de type élastoplastique formulé en grandes déformations. Il est basé sur le critère quadratique de Hill.

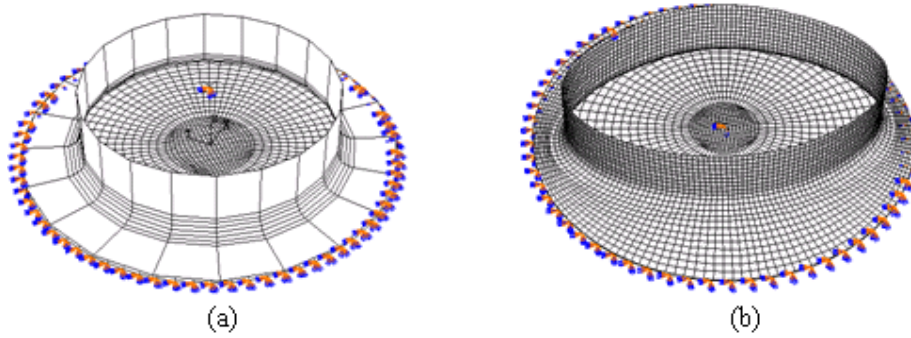


FIG. 5.10 – Modèle géométrique EF de l’essai : (a) Cas d’une matrice circulaire , (b) Cas d’une matrice elliptique

- Les conditions aux limites consistent en un encastrement imposé sur la matrice et une pression de profil linéaire exercée sur le flan selon le tableau 5.1.

Temps (s)	Pression (MPa)
0	0
100	30

TAB. 5.1 – Profil de la pression

- Les matrices circulaire et elliptique sont considérées comme des corps rigides pour toutes les simulations.

Nous simulons l’essai avec l’acier AISI 304 dont les caractéristiques sont listées ci-dessous :

- Le module d’Young : $E = 193000$ MPa
- Le coefficient de poisson : $\nu = 0.3$
- Les coefficients de Lankford expérimentaux sont illustrés dans le tableau 5.2.

r_0	r_{45}	r_{90}	\bar{r}
0.93	1.07	0.87	0.94

TAB. 5.2 – Coefficients de Lankford de l’acier inox

- La courbe d’érouissage (figure 5.11) : Elle est issue de l’essai de traction uniaxiale effectué sur des éprouvettes de dimensions normalisées.

On présente par la suite la déformée (Figure 5.12) ainsi que les isovaleurs de quelques variables pour une pression maximale de 24MPa (Figures 5.13, 5.14 et 5.15). Les figure 5.16-a et 5.16-b représentent la variation de la pression en fonction de la hauteur au pôle. Elles sont déterminées par les deux approches numérique et expérimentale. La confrontation de deux réponses pour les deux types de matrices (circulaire et elliptique) a montré un écart significatif. Ce même essai a été traité par [Ben1992] qui a montré cependant une différence en terme de réponse globale pression-hauteur au pôle entre l’expérience et les prévisions. Cette différence peut être expliquée par le fait que la loi constitutive implémentée dans ABAQUS ne décrit pas correctement le comportement de la tôle ou que l’identification des

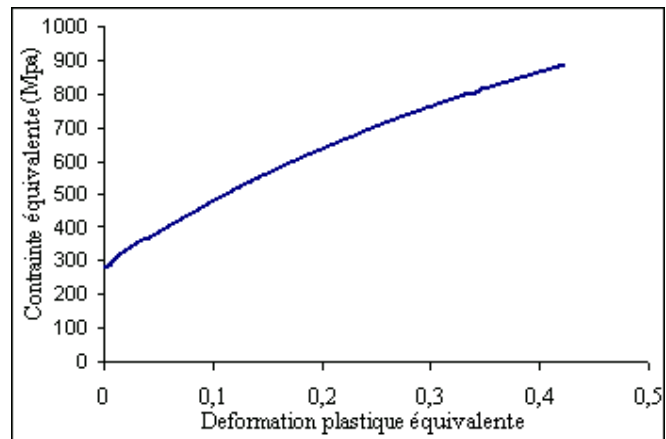


FIG. 5.11 – Courbe d'écrouissage de l'acier inox

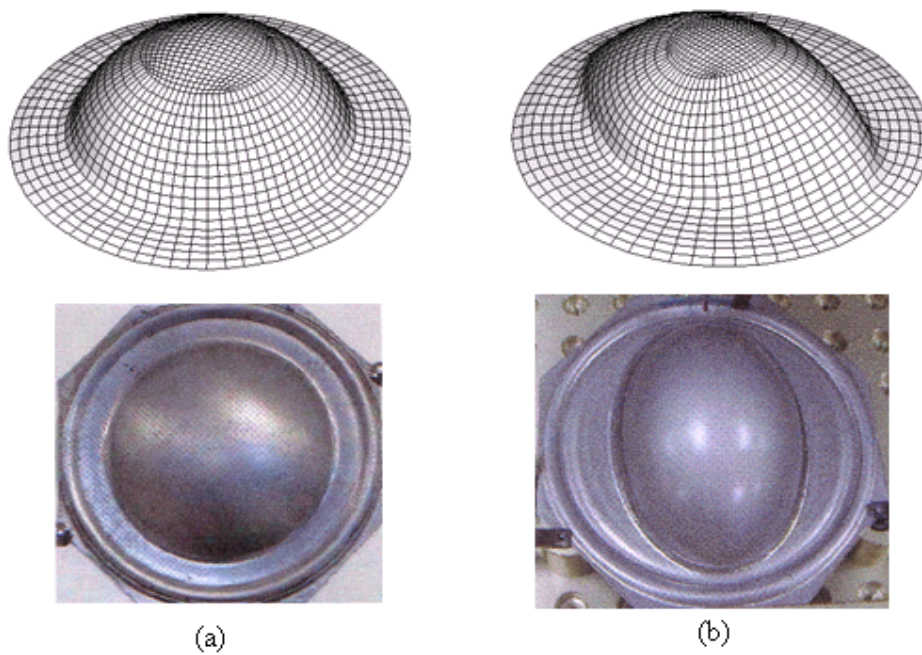


FIG. 5.12 – Représentation de la forme déformée : (a) Cas d'une matrice circulaire, (b) Cas d'une matrice elliptique

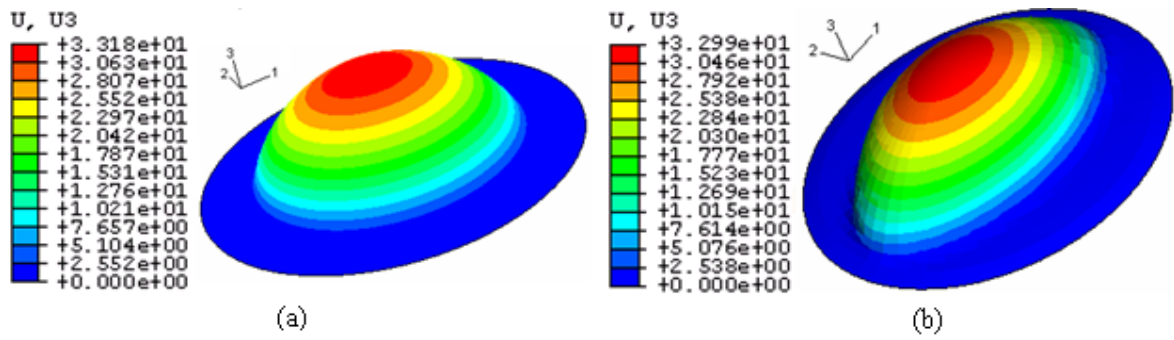


FIG. 5.13 – Isovaleurs de déplacement à l'increment final (P=24MPa) : (a) Cas d'une matrice circulaire, (b) Cas d'une matrice elliptique

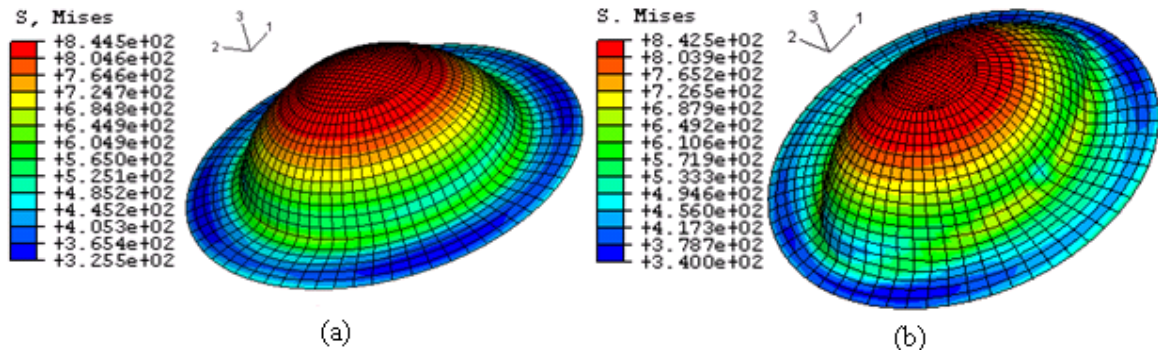


FIG. 5.14 – Contrainte équivalente de Von Mises : (a) Cas d'une matrice circulaire, (b) Cas d'une matrice elliptique

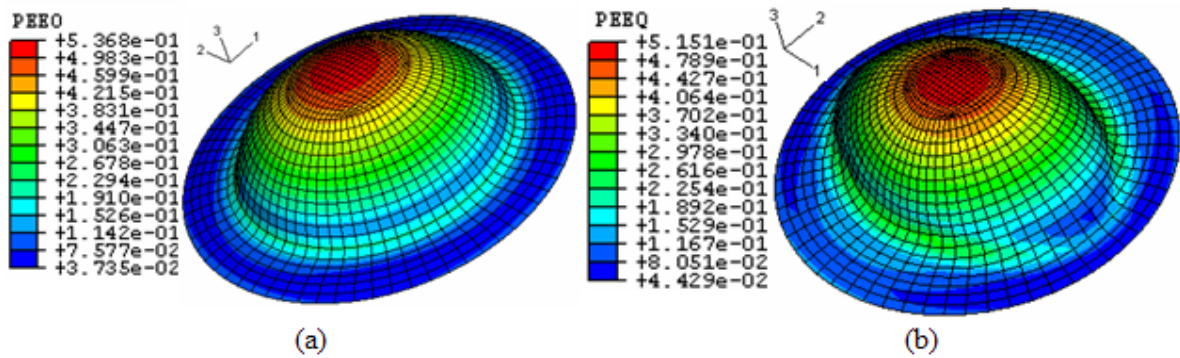


FIG. 5.15 – Déformation plastique équivalente : (a) Cas d'une matrice circulaire, (b) Cas d'une matrice elliptique

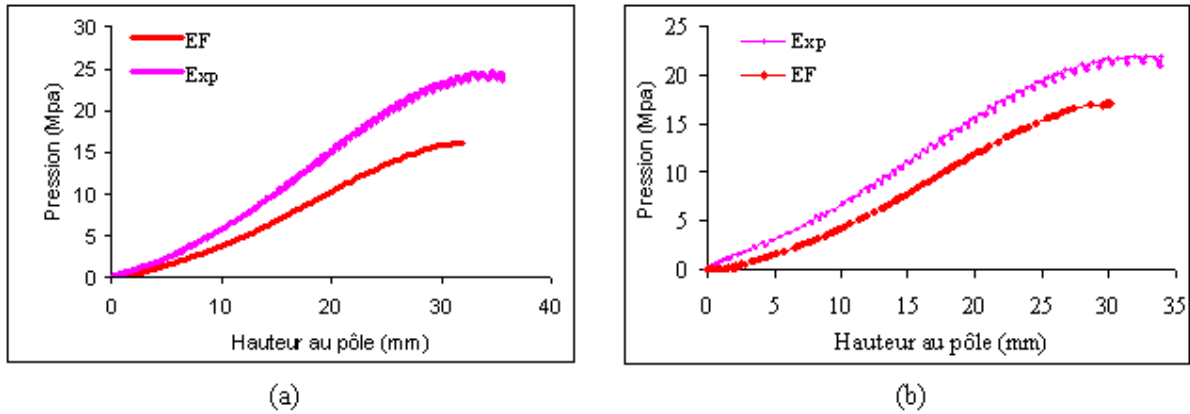


FIG. 5.16 – Pression en fonction de la hauteur au pôle : (a) Cas d’une matrice circulaire, (b) Cas d’une matrice elliptique

constantes du modèle à partir de l’essai de traction simple est insuffisant. Nous avons supposé que cet écart entre les réponses numériques et expérimentales provient de la caractérisation du matériau à partir d’un essai de traction simple qui aurait du peut être remplacer par un essai équibi-axiale. Pour cela on a proposé d’identifier la loi de comportement à partir de l’essai de gonflement hydraulique lui-même. Pour ce faire nous avons fait d’abord une analyse de contrainte et de déformation au pôle et à quelques nœuds situés loin du pôle. Peut-on donc, passer à travers un modèle analytique pour identifier la courbe d’écrouissage à partir de l’essai lui même? Ceci sera l’objet du paragraphe suivant.

5.3.2 Analyse numérique de l’essai

Dans cet essai, on s’intéresse à l’état de contrainte et de déformation, uniquement aux nœuds représentés sur la figure 5.17. Deux cas de matériaux sont étudiés : Un avec écrouissage isotrope et l’autre avec écrouissage anisotrope. L’essai est simulé avec matrice circulaire. D’après les figures 5.18, 5.19, 5.20 et 5.21 qui correspondent au cas d’un matériau isotrope, nous constatons que :

- Le rapport en contrainte et en déformation au pôle est égale à un. Cela veut dire que l’essai au pôle peut être interprété comme étant équibiaxiale en contrainte et en déformation dans le cas de l’hypothèse d’un matériau isotrope. Ce rapport reste proche de un et presque constant pour les nœuds adjacents du pôle.
- L’état de contrainte est plane au pôle puisque toutes les composantes du tenseur de contrainte selon la normale à la tôle tendent vers zéro.

Pour les figures décrites précédemment correspondantes au matériau anisotrope, nous pouvons adopter les mêmes interprétations puisque le rapport en contrainte et en déformation varie autour de un. Ceci peut être expliqué par le fait que le matériau étudié est faiblement anisotrope. Il s’ensuit qu’à partir de cette analyse, il sera possible d’adopter l’hypothèse de la déformée en calotte sphérique pour identifier par l’intermédiaire d’une approche analytique

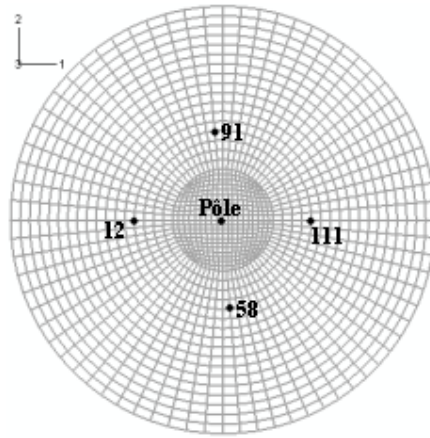


FIG. 5.17 – Nœuds choisis sur la partie utile du flan.

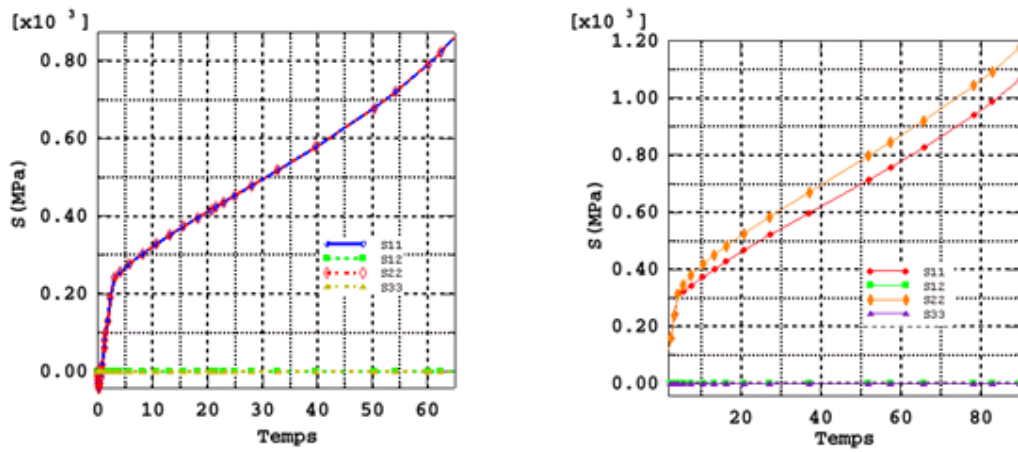


FIG. 5.18 – Contraintes au pôle : A droite cas anisotrope et à gauche cas isotrope

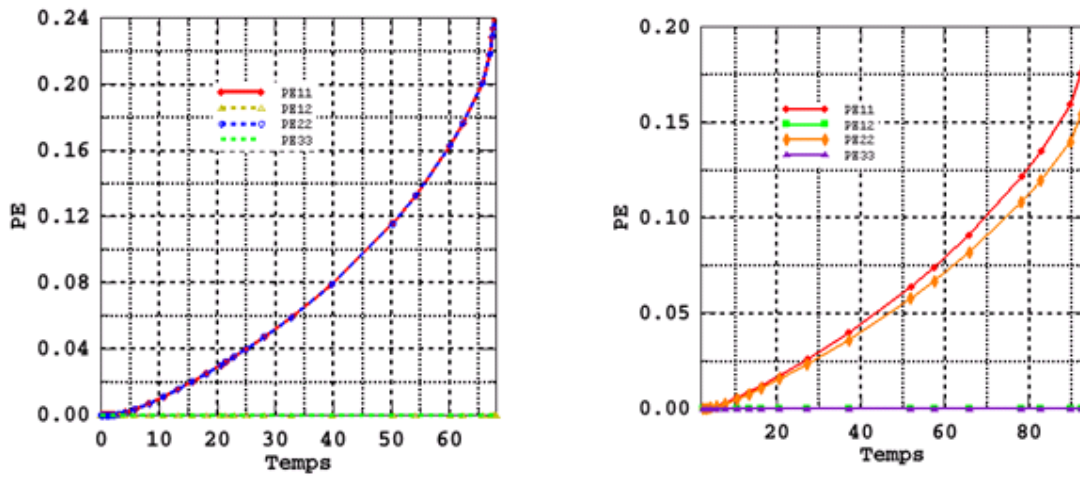


FIG. 5.19 – Déformations plastiques au pôle : A droite cas anisotrope et à gauche cas isotrope

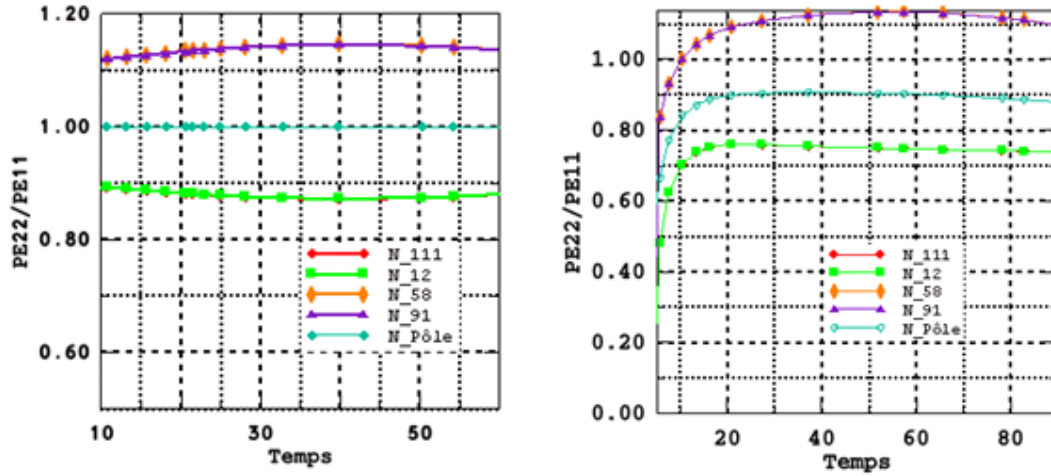


FIG. 5.20 – Rapport des déformations au pôle : A droite cas anisotrope et à gauche cas isotrope

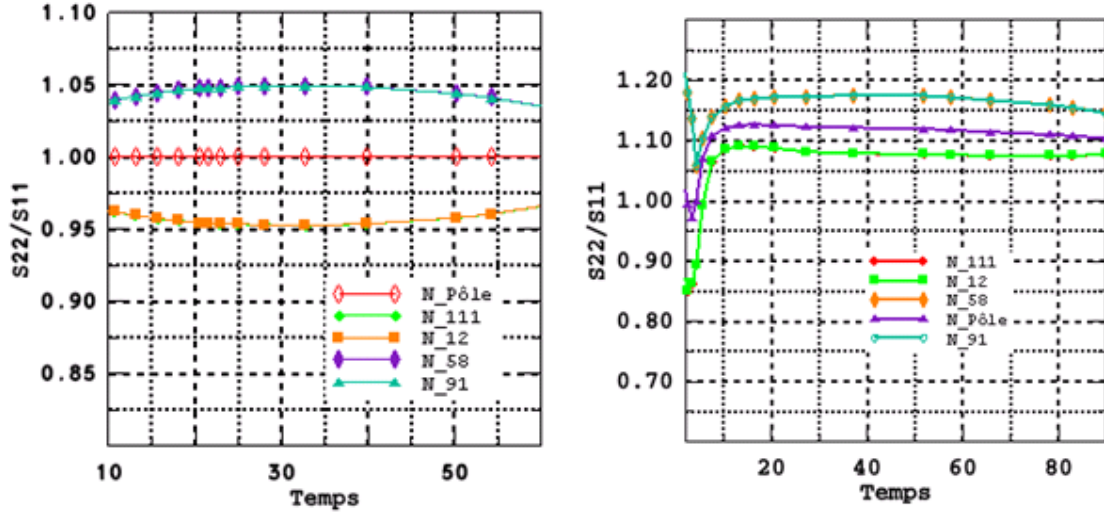


FIG. 5.21 – Rapport des contraintes au pôle : A droite cas anisotrope et à gauche cas isotrope

la courbe d'érouissage à partir de l'essai d'hydroformage. Ceci fait l'objet du paragraphe suivant :

5.4 Identification de la courbe d'érouissage par une interprétation bi-axiale

5.4.1 Hypothèses

Notre modélisation préconise les hypothèses suivantes :

- Le matériau est isotrope,
- L'éprouvette se déforme en gardant une symétrie de révolution,
- Les directions normales et tangentielles selon une méridienne sont supposées des directions principales des contraintes et des déformations,
- La déformation plastique est sans variation de volume.

5.4.2 Détermination de la courbe d'érouissage

Suivant les hypothèses citées ci-dessus on considère l'équilibre d'un tronçon de la calotte sphérique soumis à une pression p (figure 5.22).

$$p = \frac{2e\sigma_p}{\rho} \quad (5.1)$$

$$\sigma_c = \sigma_m = \sigma_p \quad (5.2)$$

$$\epsilon_c = \epsilon_m = \epsilon_p = \text{Log}\left(\frac{2\rho}{2\rho - u}\right) \quad (5.3)$$

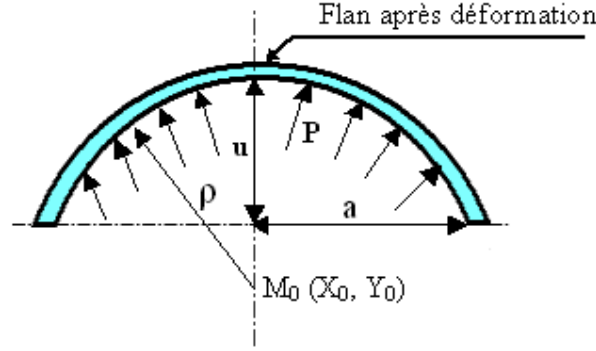


FIG. 5.22 – Equilibre du tronçon de la calotte sphérique

A partir d'une estimation du rayon de courbure au pôle :

$$\rho = \frac{a^2 + u^2}{2u} \quad (5.4)$$

et de la courbe expérimentale pression en fonction de la hauteur au pôle, on peut écrire les relations suivantes :

$$\epsilon_c = \epsilon_m = \epsilon_p = \text{Log}\left(1 + \frac{u^2}{a^2}\right) \quad (5.5)$$

Sachant que :

$$e = e_0 \exp(\epsilon_{33}) \quad (5.6)$$

Avec :

$$\epsilon_{33} = -2\epsilon_{11} \quad (5.7)$$

et

$$\epsilon_{11} = \epsilon_{22} = \text{Log}\left(1 + \frac{u^2}{a^2}\right) \quad (5.8)$$

ce qui donne :

$$e = e_0 \left(1 + \frac{u^2}{a^2}\right)^{-2} \quad (5.9)$$

Finalement on trouve l'expression de la contrainte en fonction de la pression et de la hauteur au pôle soit :

$$\sigma = \frac{p(a^2 + u^2)^3}{4e_0 u a^4} \quad (5.10)$$

Par suite nous avons pu reconstituer la courbe d'écroissage à partir de la courbe pression en fonction de la hauteur au pôle (figure 5.23) :

Les résultats de la simulation utilisant la courbe d'écroissage identifiée à partir de l'essai de gonflement hydraulique par une approche analytique sont présentés dans la figure 5.24. D'après cette figure on constate que la réponse numérique pression-hauteur au pôle, surestime la réponse expérimentale dans le cas d'une courbe d'écroissage issue d'un essai

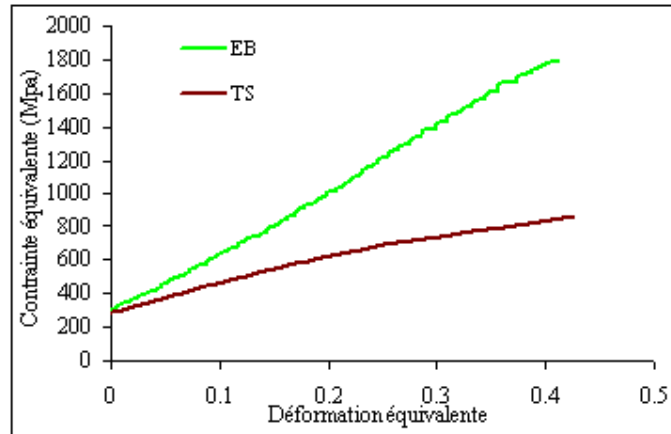


FIG. 5.23 – Courbes d’écrouissage issues de la traction simple et de l’essai equibiaxiale

d’expansion équi-biaxiale, et le sous-estime dans le cas d’une courbe d’écrouissage relevée d’un essai de traction simple. Nous pensons alors, ajuster les paramètres du matériau. Pour ce faire, l’algorithme d’optimisation développé sera utilisé pour l’identification des paramètres du modèle de comportement. Les résultats issus de cette identification seront comparés aux résultats obtenus par une stratégie d’identification basée sur une méthode d’optimisation classique. Les paragraphes qui suivent présentent la démarche de ce travail.

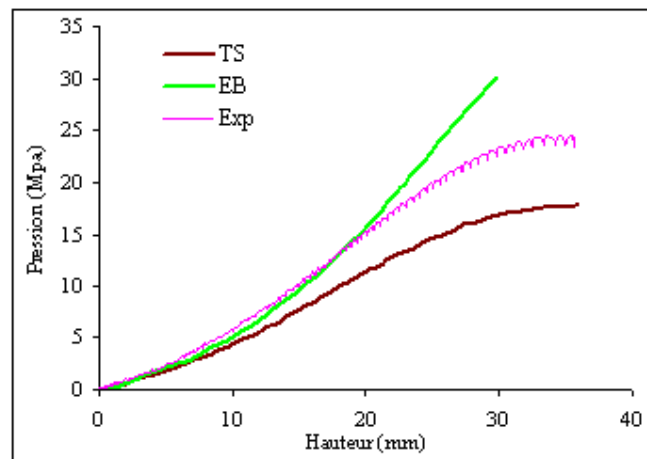


FIG. 5.24 – Comparaison entre les réponses pression-déplacement

5.5 Identification par méthode inverse

5.5.1 Méthode basée sur les procédures d'optimisation classique

5.5.1.1 Stratégie d'identification

L'objectif est l'identification de modèle orthotrope de Hill donnant une bonne concordance entre la réponse expérimentale et la réponse obtenue par EF. La démarche générale pour réaliser ceci est illustrée dans la figure 5.25.

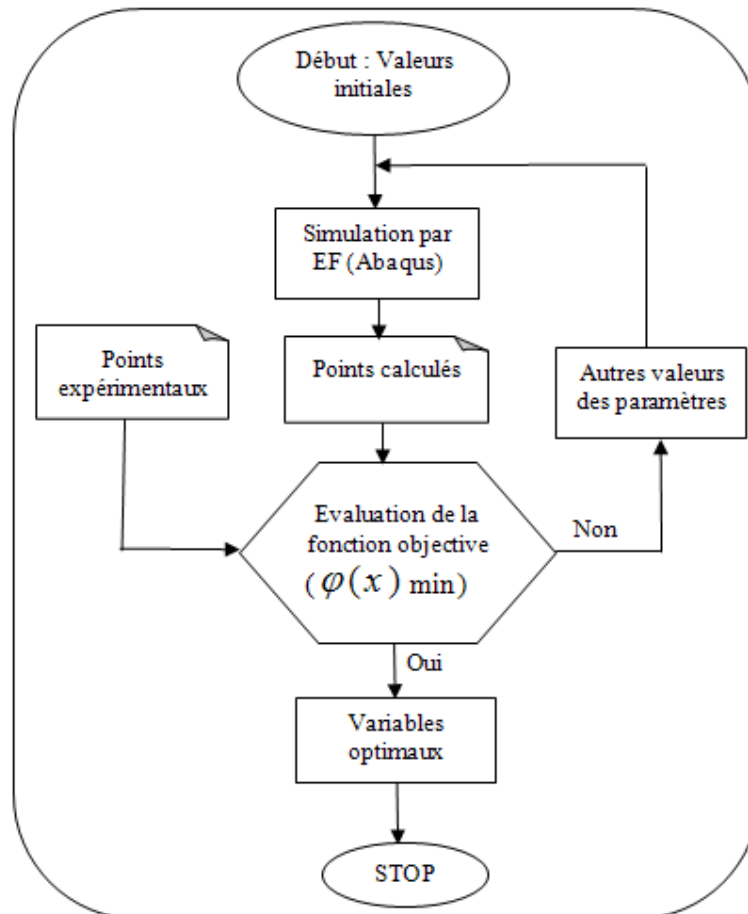


FIG. 5.25 – Procédure d'identification

Les données expérimentales se présentent sous la forme d'un échantillon de points où les mesures ont été réalisées (le hauteur u au pôle pour différentes valeurs de la pression $p = p_1, \dots, p_i, \dots, p_n$). Numériquement, on a défini la distance entre la courbe expérimentale et celle simulée comme la somme des carrés des différences d'ordonnées entre chaque point expérimental et le point de même abscisse sur la courbe obtenue par simulation. On définit

la notion d'écart relatif entre les deux courbes par la relation suivante :

$$\varphi(x) = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{p_i^{EF}(x) - p_i^{Exp}}{p_i^{Exp}} \right)^2} \quad (5.11)$$

Où x est le vecteur des valeurs des paramètres matériels, n est le nombre des points expérimentaux, p_i^{Exp} est l'ordonnée du i^{ieme} point obtenu expérimentalement, et p_i^{EF} est l'ordonnée du i^{ieme} point obtenu par simulation. On obtient ainsi la forme canonique du problème d'identification paramétrique envisagé.

Trouvez $x = (x_1, x_2, \dots, x_t)$ Tel que : $\varphi(x)$ soit minimale

Ce problème présente donc une forme classique de problème d'optimisation. Pour toutes les simulations les variables de conception sont :

- Les coefficients de Lankford : r_0 , r_{45} et r_{90}
- La fonction d'érouissage définie par : σ_0 , k et n

Ces paramètres sont identifiés selon les deux étapes suivantes :

Etape1 : Dans cette étape il s'agit de trouver σ_0 , k et n en utilisant :

- L'essai de gonflement avec une matrice circulaire.
- Les coefficients du Lankford expérimentaux donnés dans le tableau 5.2.

Etape2 : Dans cette étape il s'agit de trouver les coefficients de Lankford en utilisant :

- L'essai de gonflement avec une matrice elliptique.
- La courbe d'érouissage $\sigma_s(\epsilon^p)$ identifiée de la première étape.

Cette procédure d'identification a été programmée sous MATLAB et couplée au code de calcul par éléments finis ABAQUS (Figure 5.25).

5.5.1.2 Résultats et interprétation

Les paramètres identifiés à partir de la première étape sont illustrés dans le tableau 5.3 en comparaison avec les paramètres expérimentaux. On constate que les valeurs des paramètres présentent un écart entre eux. Une comparaison des résultats de cette étude en terme de pression-hauteur au pôle a été présentée dans la figure 5.26 pour le cas d'une matrice circulaire sous le nom Etape1. Un écart entre la courbe numérique et celle expérimentale à des pressions élevées, a été constaté.

Afin de calibrer les deux réponses et de raffiner encore le modèle, nous proposons d'identifier les coefficients de Lankford en utilisant la même fonction objectif. Mais les variables de conception dans ce cas sont les coefficients de Lankford. On adopte l'hypothèse que l'essai avec une matrice elliptique est plus sensible au coefficient de Lankford. Pour cela, nous suivons la démarche décrit dans la deuxième étape en vue d'identifier ces paramètres.

Une comparaison entre les valeurs des paramètres identifiés et les valeurs expérimentales est présentée dans le tableau 5.4. Ces valeurs sont proches de un. Ceci est logique puisque le matériau que nous étudions présente une faible anisotropie. Les réponses globales pression-hauteur au pôle, obtenues par un calcul EF utilisant ces valeurs sont représentées dans la

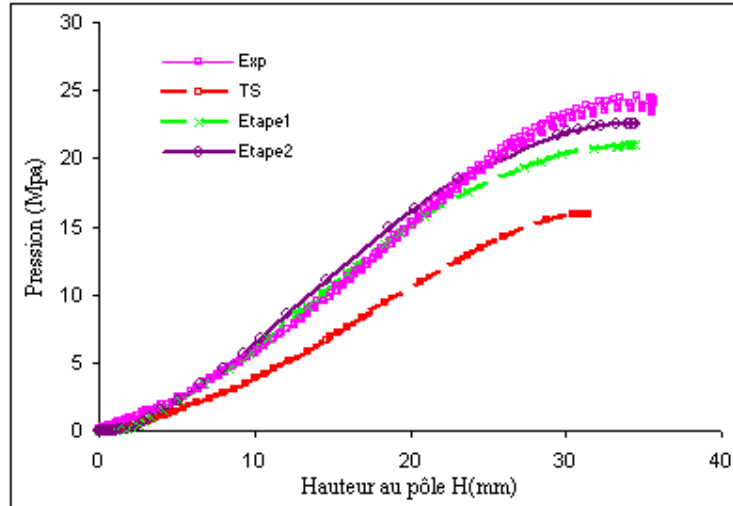


FIG. 5.26 – Pression en fonction de la hauteur au pôle : Matrice circulaire

figure 5.26 pour le cas d’une matrice circulaire et dans la figure 5.27 pour le cas d’une matrice elliptique, sous le nom Etape2. L’écart constaté entre les réponses expérimentales et simulées est plus faible que celui enregistré entre les mêmes réponses dans la première étape (Etape1).

Variable	Unité	Valeurs identifiées	Valeurs expérimentales
σ_0	MPa	283.3	238.2
k	MPa	1154	1172
n	-	0.4	0.67

TAB. 5.3 – Comparaison entre les valeurs identifiées et expérimentales

	r_0	r_{45}	r_{90}
Expérimentaux	0.93	1.07	0.87
Identifiés	1.08	1.26	1.1

TAB. 5.4 – Comparaison des coefficients de Lankford identifiés et expérimentaux

La mauvaise concordance entre quelques réponses peut être aussi expliquée par la non convergence de la méthode d’optimisation utilisée vers un minimum global. En effet, cette méthode d’optimisation classique fournit pour certains problèmes une mauvaise convergence voire même pas de convergence de tous dans d’autres cas [Pon2006]. Elle est aussi trop lente. Dans notre cas, le nombre d’itération minimum requis pour la recherche de l’optimum d’une variable donnée est de 20 itérations. Chaque itération consomme environ 15 minutes de calcul

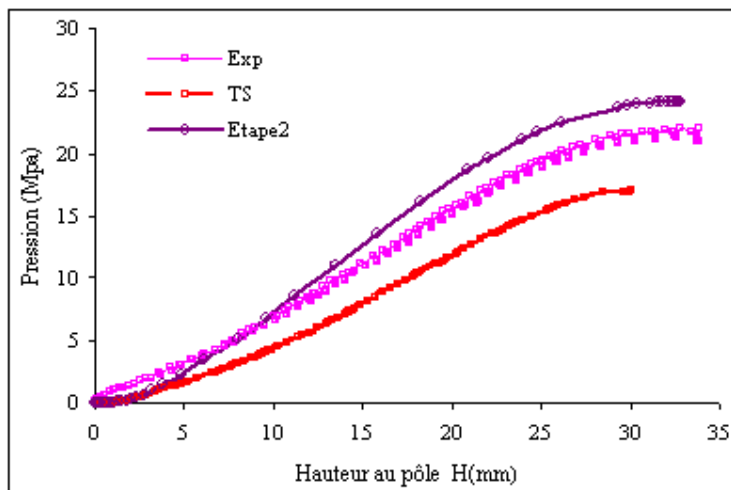


FIG. 5.27 – Pression en fonction de la hauteur au pôle : Matrice elliptique

sur un ordinateur muni d'un processeur à 1.6GHz et une mémoire vive de 256Méga-Octets. Cette méthode peut aboutir la plus part des cas à un optimum mais ce dernier peut être non significatif. Les paramètres optimaux peuvent être déterminés en dehors de leurs domaines admissibles. L'ajout des contraintes sur les valeurs des variables peut avoir une mauvaise conséquence sur le temps de calcul. C'est pourquoi Ponthot et al. [Pon2006] en 2006, ont proposé de combiner certaines des méthodes d'optimisation, avec passage automatique de l'une vers l'autre lorsque la première devient inefficace. Pour remédier à ces problèmes, nous proposons d'appliquer les techniques des RNA en vue d'atteindre les mêmes objectives. Ceci est présenté dans le paragraphe suivant.

5.5.2 Méthode basée sur les RNA

5.5.2.1 Procédure d'identification

L'identification des modèles de comportement consiste à trouver les paramètres matériels qui permettent de réduire l'écart (méthodes classiques ou méthodes stochastiques) entre la mesure expérimentale et la réponse calculée du modèle. Cette réponse du modèle est la réponse optimisée quand le processus d'identification converge. En utilisant l'algorithme présenté dans le troisième chapitre (Figure 5.28), la même démarche que celle du paragraphe précédent a été adoptée pour chercher l'optimum des paramètres du modèle de comportement.

Le processus d'identification dans ce cas n'est pas itératif. En effet, d'après la figure 5.28 on commence par des simulations par éléments finis (Partie gauche de la figure) en utilisant différentes valeurs des paramètres opératoires. Pour chaque simulation on enregistre la courbe pression en fonction de la hauteur au pôle de la tôle $P(h)$. Cette fonction est ensuite enregistrée sous forme d'un tableau (P_i, h_i) on considérant que :

- $P_1 = P(h = 0)$

$$- P_n = P(h_{max}) = 24MPa$$

Etant donnée que nous avons introduit la pression à des incréments fixes, de 0 à 24 bars par incréments de 2.4bars, il en résulte un tableau comportant 10 composantes d'un vecteur h_i correspondant à chaque combinaison du vecteur des paramètres opératoires Par_i . Les deux vecteurs sont ensuite enregistrés et normalisés suivant l'expression suivante :

$$X^{norm} = \frac{X_{old} - X_{min}}{X_{max} - X_{min}}(Y_{max} - Y_{min}) + Y_{min} \quad (5.12)$$

Avec X^{norm} la variable à normaliser, X_{old} est la valeur réelle de la variable avant normalisation, X_{max} , X_{min} sont la valeur maximale et minimale de la variable avant normalisation respectivement, et Y_{max} , Y_{min} sont la valeur maximale et la valeur minimale de la variable après normalisation respectivement.

Les deux vecteurs (Par_i^{norm} , h_i^{norm}) sont enregistrés dans une base de données qui sera utile pour l'apprentissage du modèle RNA. Avec Par_i^{norm} est le vecteur des paramètres et h_i^{norm} est le vecteur des hauteurs au pôle. Durant la phase d'apprentissage le modèle RNA génère une fonction implicite reliant la forme de la réponse globale aux paramètres opératoires du procédé. Cette fonction sera utilisée ensuite, pour identifier les paramètres du matériau à partir de la réponse pression-hauteur au pôle expérimentale (Partie droite de la figure 5.28). En résumé, cette procédure d'identification contient trois phases essentielles

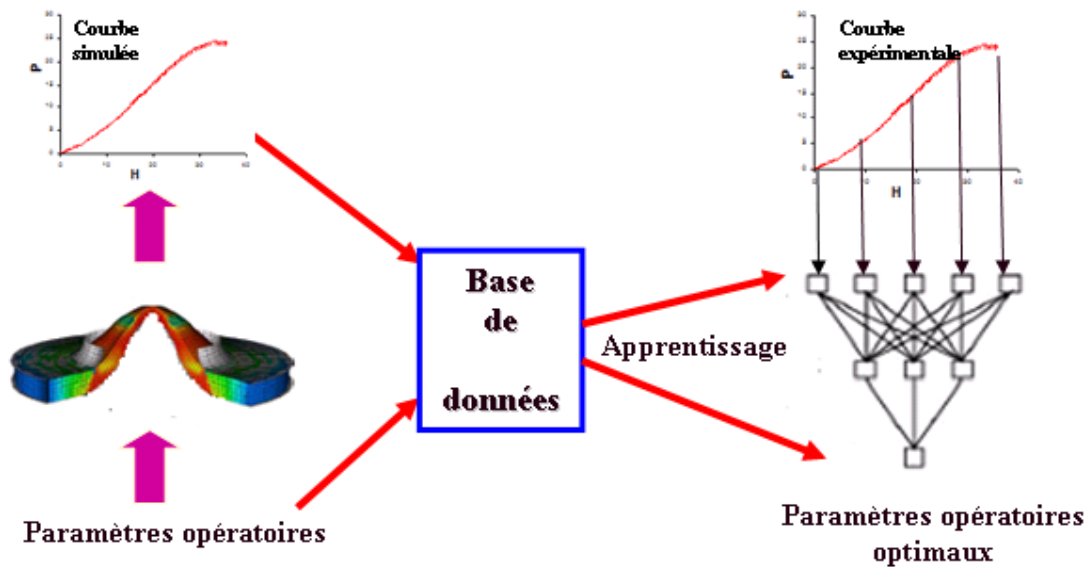


FIG. 5.28 – Procédure d'identification par RNA

pour la recherche des paramètres optimaux du modèle :

- Choix des paramètres opératoires,
- Construction et apprentissage d'un modèle RNA à partir des résultats obtenus par le modèle EF,
- Identification inverse à partir de la réponse expérimentale.

5.5.2.2 Choix des paramètres opératoires du procédé

Pour un procédé donné on distingue les paramètres liés à la géométrie et au procédé ainsi que les paramètres matériels. Dans le cas de l'hydroformage particulièrement l'essai que nous utilisons ici, les paramètres autres que ceux liés au matériau sont : Le rayon de la matrice et le coefficient de frottement entre la matrice et le flan. Des études précédentes montrent que ces paramètres n'ont pas d'influence sur la réponse globale pression en fonction de la hauteur au pôle [Cha2003]. Donc ces derniers seront négligés dans la suite de cette étude. Par suite la réponse globale pression en fonction de la hauteur au pôle n'est sensible qu'aux paramètres matériels. C'est la raison pour laquelle cet essai est jugé bon pour la caractérisation du matériau dans les cas d'un chargement complexe.

Etant donnée que nous cherchons à identifier les paramètres de l'acier AISI304, une gamme des matériaux a été définie par variation des paramètres opératoires du procédé. Chaque valeur de chaque variable est choisie à plus au moins 25% par rapport à la valeur expérimentale identifiée à partir de l'essai de traction simple. Ce qui donne le tableau 5.5 pour les paramètres qui sont liés à l'érouissage et le tableau 5.6 pour les paramètres qui sont liés à l'anisotropie. Puisque le matériau présente une faible anisotropie et en vue de réduire la base de données, nous avons considéré uniquement deux niveaux pour les coefficients de Lankford. Ces derniers sont pris à différentes angles par rapport à la directions de laminage ψ_0, ψ_{45} et ψ_{90} .

Niveaux	Paramètres		
	σ_0 (Mpa)	k (Mpa)	n
1	178.5	879	0.375
2	238	1172	0.5
3	297.5	1465	0.625

TAB. 5.5 – Niveaux et valeurs des paramètres liés à l'érouissage

Niveaux	Paramètres		
	r_0	r_{45}	r_{90}
1	0.7125	0.8025	0.6525
2	1.1875	1.3375	1.0875

TAB. 5.6 – Niveaux et valeurs des paramètres liés à l'anisotropie

Les propriétés élastiques du flan sont prises selon un modèle linéaire isotrope avec un module d'Young de 193000MPa et un coefficient de Poisson de 0.3.

Nous rappelons les étapes de calcul qui seront adoptées dans la suite de cette étude :
 Etape1 : Dans cette étape il s'agit de trouver σ_0 , k et n utilisant :

- L’essai de gonflement avec une matrice circulaire.
- Les coefficients de Lankford expérimentaux donnés dans le tableau 5.3.

Étape 2 : Dans cette étape il s’agit de trouver les coefficients de Lankford utilisant :

- L’essai de gonflement avec une matrice elliptique.
- La courbe d’écrouissage $\sigma_s(\epsilon^p)$ identifiée de la première étape.

5.5.2.3 Résultats et discussions

On procède, en premier lieu à l’identification de la courbe d’écrouissage à partir de l’essai avec une matrice circulaire. Dans ce cas les coefficients de Lankford déterminés expérimentalement à partir de l’essai de traction simple sont introduits. L’architecture du modèle RNA employé contient 11 neurones dans la couche d’entrée 3 neurones dans la couche cachée et 3 neurones dans la couche de sortie. Ce modèle a été entraîné par 27 courbes pression en fonction de la hauteur au pôle. Une erreur quadratique moyenne de l’ordre de 0.001 après 10800 epochs, est atteinte. Les valeurs identifiées des paramètres d’écrouissage sont présentées dans le tableau 5.7 pour une comparaison avec les valeurs expérimentales. Le temps de calcul n’a pas excédé 15 minutes sur un ordinateur muni d’un processeur à 1.6GHz et une mémoire vive de 256Méga-Octets.

VARIABLES	UNITÉ	VALEURS IDENTIFIÉES	VALEURS EXPÉRIMENTALES
σ_0	MPa	241.6	238.2
k	MPa	1422	1172
n	-	0.4	0.67

TAB. 5.7 – Comparaison entre valeurs identifiées et valeurs expérimentales

Les paramètres identifiés ont été ensuite utilisés, pour simuler l’essai par un calcul EF direct. La réponse pression en fonction de la hauteur au pôle enregistrée sous le nom Etape1, est comparée avec la réponse expérimentale. Pour le cas d’une matrice circulaire une bonne concordance a été constatée (figure 5.29). Mais pour le cas de validation par une matrice elliptique, un petit écart a été enregistré entre les deux courbes (figure 5.30). L’erreur relative maximale est de l’ordre de 7,52%.

Pour ajuster encore le modèle nous avons adopté la même démarche en passant à la deuxième étape. Il s’agit d’identifier les coefficients de Lankford à partir du même essai mais dans ce cas avec une matrice elliptique puisque cette dernière met en évidence l’importance des coefficients d’anisotropie lors de la déformation du flan. Les valeurs identifiées sont présentées dans le tableau 5.8. Ces paramètres sont identifiés en utilisant un modèle RNA contenant 11 neurones dans la couche d’entrée, 9 neurones dans la couche cachée et 3 neurones dans la couche de sortie. Le modèle a été entraîné par 24 courbes, pression en fonction de la hauteur au pôle, obtenues en simulant l’essai avec une matrice elliptique. Les valeurs sont ensuite utilisées dans un calcul direct par EF. La réponse globale de cette simulation est présentée dans la figure 5.29 pour le cas d’une matrice circulaire et la figure 5.30 pour

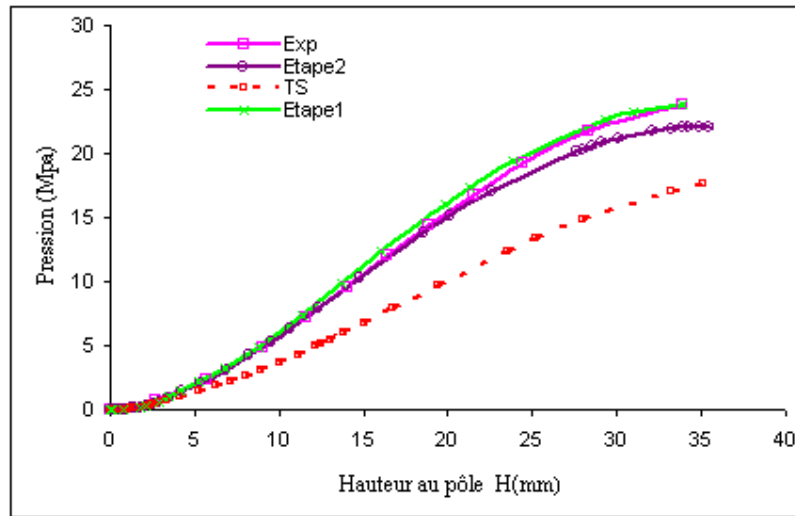


FIG. 5.29 – Pression en fonction de la hauteur au pôle : Cas d’une matrice circulaire

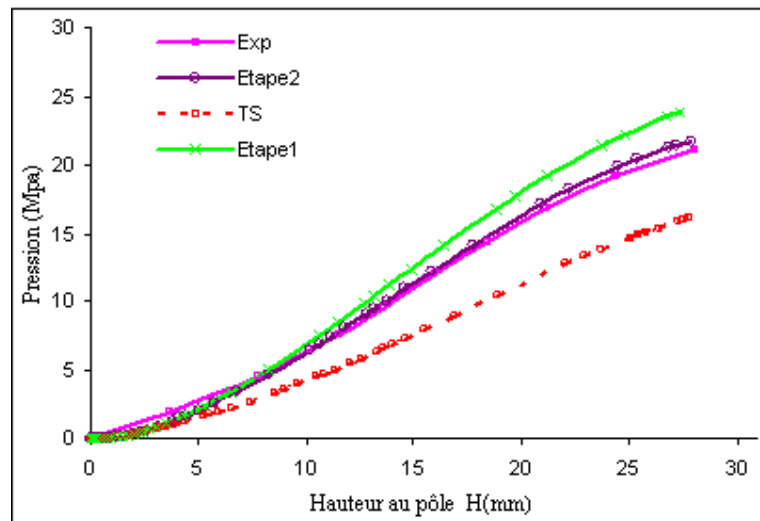


FIG. 5.30 – Pression en fonction de la hauteur au pôle : cas d’une matrice Elliptique à 0°

le cas d'une matrice elliptique sous le nom Etape2. Une bonne concordance avec la réponse expérimentale a été obtenue pour les deux cas.

Une étape de validation de cette identification consiste à simuler l'essai d'hydroformage avec les valeurs identifiées (Tableau 5.7 et 5.8). Nous exploitons dans ce cas l'essai avec matrice elliptique pour une direction de 45° par rapport à la direction de laminage. La réponse trouvée, a été tracée sous le nom Etape2 dans la figure 5.31. Elle présente une bonne cohérence avec la mesure déterminée par l'expérience.

En dépit des résultats trouvés, on remarque que les réseaux de neurones peuvent identifier d'une façon plus fiable et rapide les paramètres du matériau à partir d'une bonne prédiction de l'évolution de la pression en fonction de la hauteur au pôle dans le cas de ce procédé.

	r_0	r_{45}	r_{90}
Expérimentaux	0.93	1.07	0.87
Identifiés	1.19	0.8	0.7

TAB. 5.8 – Comparaison des coefficients de Lankford's

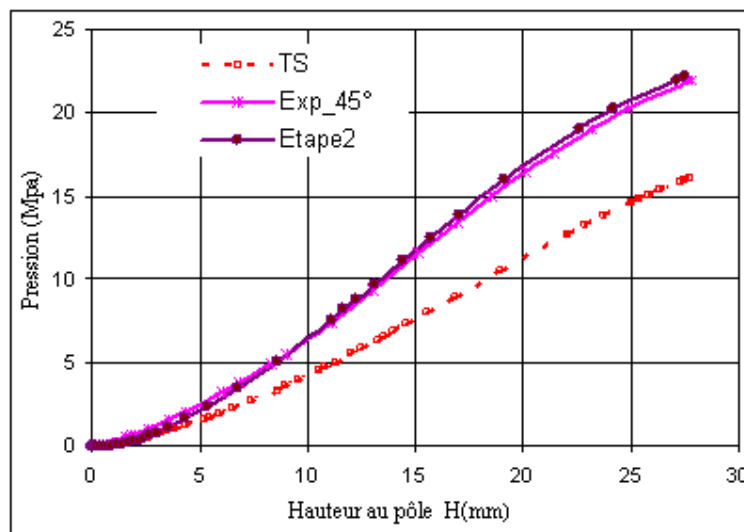


FIG. 5.31 – Pression en fonction de la hauteur au pôle : Cas d'une matrice Elliptique à 45°

5.6 Synthèse et comparaison des deux méthodes

On présente dans ce paragraphe la comparaison entre les deux méthodes d'optimisation, mises en application dans ce chapitre pour trouver la réponse optimisée du modèle de comportement, selon plusieurs points de vues :

Premièrement, nous comparons les valeurs des paramètres identifiés obtenues par l'approche expérimentale, l'approche inverse classique et l'approche neuronale. Nous constatons, d'après ces résultats qui figurent dans le tableau 5.9 que les valeurs obtenues des différents paramètres diffèrent entre eux, mais en restant toujours dans leurs domaines admissibles. La valeur identifiée du coefficient d'érouissage par l'approche expérimentale est égale à 0.67. Alors que, les deux autres approches aboutissent à une même valeur de ce paramètre qui est égale à 0.4. Ceci peut expliquer l'insuffisance de l'essai de traction simple à identifier tous les paramètres du matériau.

Variable	Unité	Expérimentale	Methode inverse classique	Méthode inverse RNA
σ_0	MPa	238.2	283.3	241.6
k	MPa	1172	1154	1422
n	-	0.67	0.4	0.4
r_0	-	0.93	1.08	1.19
r_{45}	-	1.07	1.26	0.80
r_{90}	-	0.87	1.1	0.70

TAB. 5.9 – Comparaison des valeurs identifiées avec les valeurs expérimentales

Deuxièmement, nous comparons la réponse globale (Pression en fonction de la hauteur au pôle) obtenue avec les paramètres identifiés par les deux méthodes sous le nom Etape2 dans les figures 5.26, 5.29, 5.30 et 5.31 par rapport à la réponse expérimentale. D'après ces figures nous constatons que les réponses obtenues par l'approche neuronale sont plus proches que celles obtenues par l'approche inverse classique.

Troisièmement, nous comparons les deux méthodes du point de vue temps de calcul. Il est constaté que la méthode inverse classique consomme beaucoup de temps pour aboutir à l'optimum. En effet, chaque itération a pris environ 15 minutes de calcul sur un ordinateur muni d'un processeur à 1.6GHz et une mémoire vive de 256Méga-Octets. Le nombre d'itérations pour aboutir à l'optimum peut atteindre parfois plus de 500 itérations. Alors que dans le cas de la méthode basée sur les RNA le temps de calcul est divisé en trois :

- Un temps pour la génération de la base de données : Celui ci est le plus long. Mais il reste inférieur à celui consommé par la première méthode. Il peut être encore réduit par la réduction de la taille de la base de données. Ceci nécessite une étude préliminaire de sensibilité des paramètres intervenant dans le procédé. Ce point constituera l'un des perspectives de ce travail. En effet, il est possible d'exploiter le programme développé pour ajouter une phase d'étude de sensibilité en utilisant les réseaux de neurones artificiels.
- Un temps consommé lors de la phase d'apprentissage : C'est le temps nécessaire pour que le modèle ajuste ses paramètres internes (Poids synaptiques, nombre d'epochs,...). Ce temps peut être aussi réduit. Mais dans toutes nos applications il n'a jamais dépassé un quart d'heure. Et même s'il prenait pour certains autres processus de fabrication un temps plus long, il sera réduit par l'aspect de parallélisme dont bénéficie les techniques des RNA. En effet, lors de la phase d'apprentissage on peut lancer cette dernière sur plusieurs processeurs en même temps.

- Un temps de calcul : C'est le temps nécessaire pour que le modèle fournit la réponse à une excitation donnée. Ce dernier pour nos applications est de l'ordre de quelques secondes.

5.7 Conclusion

Dans ce chapitre, une démarche utilisant le procédé d'hydroformage a été proposé pour l'identification des paramètres du matériau AISI304. L'algorithme d'optimisation développé et qui est basé sur les réseaux de neurones artificiels a été exploité dans cette application. Les résultats issus de cet algorithme ont été comparés avec ceux obtenus par une méthode d'optimisation classique utilisant la même démarche. Pour valider cette procédure, des essais expérimentaux ont été exploités. Une comparaison entre les différentes approches (Inverse classique et neuronale) en terme de pression-hauteur au pôle montre que les résultats issus d'une approche neuronale sont meilleurs, comparés à ceux obtenus par une approche inverse classique, de plusieurs points de vue :

- Temps de calcul très court si on ne tient pas compte de la génération de la base de données,
- Les réponses obtenues sont en bonne concordance avec les réponses expérimentales,
- Les paramètres identifiés restent toujours dans le domaine admissible de la variable,
- Bien adapté pour les problèmes fortement non linéaires.

Conclusions et Perspectives

L'objectif de cette contribution portait sur l'utilisation des Réseaux de Neurons Artificiels pour l'optimisation des procédés de mise en forme des tôles. Les procédés que nous avons étudiés ici sont l'emboutissage et l'hydroformage.

Ce travail a débuté d'abord, par une étude bibliographique portant sur les procédés de mise en forme particulièrement, par emboutissage et hydroformage ainsi que les différentes méthodes qui permettent leur optimisation. Nous avons distingué entre les méthodes basées sur le calcul de gradient et celles qui sont de type stochastique. Les différentes notions et aspects théoriques qui ont servi pour la partie développement dans ce travail ont été exposées dans le deuxième chapitre. Un aperçu historique sur les principaux travaux utilisant les réseaux de neurones artificiels dans des applications mécaniques a été aussi présenté dans la fin de ce dernier. Ensuite nous avons développé dans un premier temps, un algorithme d'optimisation basé sur les RNA, baptisé PONN pour prédire certaines réponses matérielles et géométriques de ces procédés. Dans un second temps, une étude paramétrique pour l'ajustement des paramètres du modèle RNA a été faite et des tests de validations sont présentés à la fin du troisième chapitre.

Dans le quatrième chapitre, nous avons mis en application cet algorithme afin de modéliser certaines opérations de mise en forme. En effet, il a été appliqué pour la modélisation et l'optimisation d'un essai d'emboutissage profond. Nous avons calculé l'épaisseur minimale dans un embouti cylindrique par combinaison des paramètres liés au matériau et des paramètres technologiques tel que le coefficient de frottement. Dans une première partie, les réseaux de neurones sont utilisés pour substituer un calcul par éléments finis et de s'affranchir par suite de la lourdeur de ce dernier lors d'un calcul direct par simple variation de l'un des paramètres opératoires du procédé. Les résultats obtenus montre une bonne concordance avec les résultats obtenus par la méthode des éléments finis. L'erreur maximale était inférieure à 0.1% pour chaque point de conception. Cela prouve la potentialité de ces modèles à prédire des réponses lorsqu'on attribut d'autres valeurs aux paramètres du procédé.

Dans une deuxième partie de ce chapitre, nous avons utilisé cet algorithme pour l'optimisation des modèles de comportement utilisant le même essai. Une procédure d'identification des paramètres du matériau utilisant PONN a été présentée. L'avantage de cette méthode d'optimisation est qu'elle ne tient pas compte de l'aspect physique des paramètres qui sont mis à l'identification. Il a été constaté que cette méthode devient moins efficace lorsque le nombre des paramètres mis à l'identification serait supérieur au nombre des données à l'entrée du réseau.

Une troisième partie du quatrième chapitre est consacrée à l'optimisation des paramètres géométriques du procédé. Nous étions intéressés dans cette partie au phénomène du retour

élastique. L'objectif est de déterminer les valeurs optimales des rayons de congés de la matrice et du poinçon, et du jeu résiduel entre le poinçon et la matrice qui minimise le retour élastique dans l'embouti. Cette optimisation a été effectuée en se basant sur les résultats d'une prédiction par un modèle RNA en s'appuyant sur la technique des plans d'expériences. Le traçage des surfaces de réponse a permis de déterminer d'une façon globale le minimum du retour élastique et leurs paramètres optimaux correspondants.

La dernière partie de ce travail a été présentée dans le cinquième chapitre. Elle a été consacrée à l'identification des paramètres de l'acier AISI304 à partir d'un procédé d'hydroformage d'un flan circulaire par l'intermédiaire d'une matrice circulaire et une matrice elliptique. Dans un premier temps, nous avons présenté la partie expérimentale de l'essai d'hydroformage. Une analyse numérique par éléments finis de l'essai a été ensuite, effectuée en vue d'exposer la problématique de la suite du travail. Dans un second temps, l'algorithme présenté dans le troisième chapitre a été exploité dans cette étude pour l'identification. Mais pour mettre en valeur cette méthode, les mêmes paramètres ont été déterminés en utilisant une approche analytique et une autre basée sur une méthode d'optimisation classique. Les paramètres qui sont mis à l'identification concernent la courbe d'écrouissage ainsi que les coefficients d'anisotropie. Une étude comparative entre la réponse expérimentale (pression-hauteur au pôle) et celles obtenues par les valeurs identifiées issues des deux approches est présentée. Une bonne corrélation a été obtenue par les paramètres issus d'une approche neuronale par rapport à la réponse expérimentale. Nous avons pu apprécier, ainsi les avantages de celle-ci par rapport à la méthode d'optimisation classique :

- Une bonne concordance a été constatée,
- Un gain de temps calcul et une rapidité notable,
- Il n'y a pas de risque que les variables soient déterminées en dehors de leurs domaines admissibles.

A partir des résultats trouvés dans ce travail et la manipulation de cet outil tout au long de notre travail de thèse nous avons constaté que le véritable avantage de cette technique était dans leur faculté à représenter des rapports linéaires et non linéaires complexes, et dans leur capacité à apprendre ces rapports directement de la série de données modélisée. Cette propriété permet aux réseaux de neurones d'être un bon candidat pour prendre des entrées dans un nouvel ensemble et pour produire la sortie correspondante dans un temps de simulation très court et cela fait gagner aux industriels beaucoup de temps de calcul.

En conclusion on peut envisager plusieurs perspectives de ce travail :

- Il pourrait être envisagé de développer un modèle RNA permettant de générer un modèle de comportement bien déterminé. Cela permet en fait de passer à des modèles plus réalistes et plus fins puisqu'il nous offre la possibilité de prendre compte de plusieurs paramètres en même temps. Ceci peut aggraver le problème de temps de calcul déjà long. Pour cela il peut être envisagé de faire recours au parallélisme. En effet, les réseaux de neurones artificiels sont fortement parallélisables.
- Afin d'approfondir l'étude précédente, il sera intéressant aussi de voir la possibilité d'introduire cette technique (RNA) pour faire une étude de sensibilité des paramètres des

procédés. Cette étude permet de détecter les paramètres les plus influents dans un procédé sans faire des expériences complémentaires. Cela permet aussi de réduire la phase d'apprentissage du modèle RNA. Cette phase peut être développée en amont du modèle.

- L'optimisation multi-objectifs est un outil essentiel pour la prise en compte de plusieurs aspect qui sont parfois fortement dépendants. Les procédures fréquemment utilisées en optimisation multicritères nécessitent plusieurs ré-analyses qui sont indispensables pour calculer des fonctions performances ou coût généralisé. Mais la complexité des modèles et de la modélisation associée rend ces ré-analyses longues et parfois non envisageables. Pour surmonter ces difficultés, on pourra proposer d'utiliser les techniques des réseaux de neurones artificiels.

Références

- [Aba2001] Abaqus, Theory Manual, Standard User's Manuel et Explicit User's Manual, Version 6.2, Hibbit, Karson et Sorensen, Inc, 2001.
- [Ade1989] Adeli H., Yeh C., Perceptron learning in engineering design, Microcomput Civil Engng, vol.4, p.247-256, 1989.
- [Akr1984] Akrouit M.,Chaker C., Validation experimentale et numerique d'un modele pour l'etude de l'emboutissage de toles minces, CPI, Maroc, 2005.
- [And1998] Andersson K, Sellgren U., Modeling and simulation of physical behavior of complex products. Proceedings of Productmodeller, Linköping, 10 November, 1998.
- [Bah2005] Bahloul R., Optimisation du procédé de pliage sur presses de pièces en tôles à haute limite d'élasticité, Thèse de doctorat, Ecole Natioanle d'Arts et de Métiers d'Angers, France, 2005.
- [Bah2006-a] Bahloul R., Ben Eléchi S., Dal Santo Ph., Naceur H., Potiron A., Optimisation of bending process by means of response surface and moving least squares methods, Far East Journal of Applied Mathematics, Vol.22, p. 25-54, 2006.
- [Bah2006-b] Bahloul R., Ben-Elechi S., Potiron A., Optimisation of springback predicted by experimental and numerical approach by using response surface methodology, International Journal of Materials Processing Technology, Vol.173, p.101-110, 2006.
- [Bat1998] Batoz J.-L., Guo Y.Q, Mercier F., The inverse approach with simple triangular shell elements for large strain predictions of sheet metal forming parts, Engrg. Comput.15, Vol.7,P.864-892, 1998.
- [Ben1992] Benchouikh A., Formulation et identification de loi de comportement anisotropes pour tôles minces en Emboutissage, Thèse de doctorat, Ecole centrale de Lyon, France, 1992.
- [Ben2004] Ben Elechi S., Analyse et conception rapides de paramètres du procédé d'emboutissage pour le contrôle du retour élastique. Thèse de doctorat, université de Technologie de Compiègne, France, 2004.

- [Ben2005] Ben Tahar M, Contribution à l'étude et la simulation du procédé d'hydroformage, Thèse de doctorat, l'Ecole des Mines de Paris, Paris, 2005.
- [Bie2000] Bielser D., Gross M.H., Interactive simulation of surgical cuts, Proceedings of Pacific graphics, IEEE Computer Society Press, p. 116-125, 2000.
- [Bis1995] Bishop C., Neural Networks for Pattern Recognition, Oxford University Press, 1995.
- [Bou2001] Boumaza A., Louchet J., Dynamic flies : Using real-time evolution in robotics, EVOIASP Workshop, Artificial Evolution in Image Analysis and Signal Processing, 2001.
- [Bru2005] Bruni C., Forcelese A., Gabrielli F., Simoncini M., Pieralisi M., Spring back evaluation and prediction in deep drawing of autenitic manganese steels, ESAFORM, 2005.
- [Cha2003] Chamekh A., Simulation numérique de l'emboutissage des tôles minces, Mastère de recherche, Ecole Nationale d'Ingénieurs de Monastir, Tunisie, 2003.
- [Cha2004] Chamekh.A, Bel Hadj Salah H., Gahbich M. A., Ben Amara A., Dogui A., Identification of the hardening curve using a finite element simulation of the bulge test, Journal of Steel and Related Material, p.741-746, 2004.
- [Dav1993] Davalo E., Naim P., Des Réseaux de Neurones, EYROLLES, Deuxième édition, 1993.
- [Del2001] Delamizière A., Naceur H., Breitkoff P., Knof-Lenoir K., Batoz J.L., Villon P., Faisabilité en emboutissage, Optimisation du matériau par surface de réponse, Mécanique et Industries, Vol.3, p93-98, 2002.
- [Dob2004] Dobrzanski L.A., Sitek W., Krupinski M., Dobrzanski J., Computer aided method for evaluation of failure class of materials working in creep conditions, Journal of Materials Processing Technology, p.102-106, 2004.
- [Dre2002] Dreyfus G., Martinez J-M., Samuelides M., Gordon M.B., Badran F., Thiria SS., Hérault L, Réseaux de neurones : Méthodologie et Applications, Eyrolles, 2002.
- [Fle1975] R. Fletcher, An Ideal Penalty Function for Constrained Optimization, J. Inst. Maths Applics., Vol.15, p.319-342, 1975.
- [For2002] R. Forestier, E. Massoni, Y. Chastel, Estimation of constitutive parameters using an inverse method coupled to a 3D finite element software, Journal of Materials Processing Technology, Vol.125, p.594-601, 2002.
- [Fos2002] Foskey M., Otaduy M.A., Lin M.C., ArtNova : Touch-enabled 3D model design, Proceedings of the IEEE virtual reality, 2002.

- [Gah2006] Gahbich A., Caractérisation expérimentale des tôles d'emboutissage : Application à l'identification des lois de comportement, These de doctorat, L'Ecole Nationale d'Ingénieurs de Monastir, Tunisie, 2005.
- [Gav1995] Gavrus A., Massoni E., Chenot J.L., Constitutive parameter identification using a computer aided rheology approach, 5th International Conference on Numerical Methods in Industrial Forming Processes, Balkema, Rotterdam, p. 563-568, 1995.
- [Gav1996] Gavrus A., Massoni E., Chenot J.L., An inverse analysis using a finite element model for identification of rheological parameters, Journal of Materials Processing Technology, Vol.60, p.447-454, 1996.
- [Gel1995] Gelin J.C., Ghouati O., Une méthode d'identification inverse des paramètres matériels pour les comportements non linéaires, Revue Européenne des éléments Finis, Vol.4, p.463-485, 1995.
- [Gen2004] Genel K., Application of artificial neural network for predicting strain-life fatigue properties of steels on the basis of tensile tests, International Journal of Fatigue, Vol.26, p.1027-1035, 2004.
- [Gib1984] Gibson S., 3D ChaignMail : a fast algorithm for deforming volumetric objects, Proceedings of symposium on interactive 3D and graphics, Providence RI LISA, p. 149-54, 1997.
- [Gil1995] Gelin J.C., Ghouati O., Une méthode d'identification inverse des paramètres matériels pour les comportements non linéaires, Revue Européenne des éléments Finis, Vol.4, p.463-485, 1995.
- [Gol1989] Goldberg D., Genetic Algorithms in search, Optimisation and Machine Learning, Addison-Wesley, 1989.
- [Gou1998] Ghouati O., Gelin J.-C., Identification of material parameters directly from metal forming processes, Journal of Materials Processing Technology, Vol.80,p.560- 564, 1998.
- [Gou1999] Ghouati O., Gelin J.C., Baida M., Lenoir H., Simulation and control of hydroforming processes for tubes or flanges forming, in : J.A. Govas (Ed.), 2nd ESAFORM Conference on Material Forming, Guimaraes, Portugal,p. 473-476, 1999.
- [Gou1999] Goupy J., Plans d'expériences pour surfaces de réponse, Jacques Goupy, Dunod, 1999.
- [Gou2001] Goupy J., Introduction aux plans d'expériences, Dunod, Paris, 2001.
- [Ham2002] Hamda H., Jouve F., Lutton E., Schoenauer M., Sebag M., Compact unstructured representations in evolutionary topological optimum design, Applied Intelligence, Vol.16, 2002.
- [Ham2002] Hambli R., Prediction of burr height formation in blanking processes using neural network, International Journal of Mechanical Sciences, vol.44,p. 2089-2102, 2002.

- [Hor1989] Hornik K., Stinchcombe M., White H., Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, Vol.2, p.359-366, 1989
- [Hor1994] Hornik K., Stinchcombe M., White H., Auer P., Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives, *Neural Computation*, Vol.6, p. 1262-1275, 1994.
- [Hsu2002] Hsu C. W., Ulsoy A. G., Demeri M. Y., Development of process control in sheet metal forming, *Journal of materials Processing Technology*, Vol.127, p.361-368, 2002.
- [Hub2001] Huber N., Nix D. W., Gao H., Identification of elastic plastic material parameters from pyramidal indentation of thin films, *Proc. R. Soc. Lond. A*, vol.458, p. 1593-1620, 2002.
- [Jan2000] AJansson J, Horváth I, Vergeest JSM. Implementation and analysis of a mechanics simulation module for use in a conceptual design system, *Proceedings ASME Design Engineering Technical Conferences*, 2000.
- [Jan2002] Jansson J., Vergeest J.S.M., A discrete mechanics model for deformable bodies, *Comput Aided*, Vol.34, p. 913-928, 2002.
- [Jef2006] Jeffrey Kuo C.-F., Jeffrey · Wu Y., Application of a Taguchi-based neural network prediction design of the film coating process for polymer blends, *International Journal of Advanced Manufacturing Technology*, Vol.27, p.455-461, 2006.
- [Jin2000] Jin Q., Wu H., The parameter identification of superplastic constitutive model using optimisation, *Advances in Engineering Plasticity*, Vol.177-180, p.607-424, 2000.
- [Kan1996] Kanai S. and Takahashi H., Modeling and NC programming for free-form surfaces by haptic interfaces, *Proceedings of DETC/design for manufacturing*, ASME, Irvine, California, USA, p.509-18, 1996.
- [Kha2004] Khalfallah A., Identification des lois de comportement élastoplastiques par essais inhomogènes et simulations numériques, *Thèse de doctorat, Faculté des sciences de Tunis, Tunisie*, 2004.
- [Kle2000] Kleinermann J.P., Identification paramétrique et optimisation des procédés de mise a forme par problèmes inverse, *Thèse de Doctorat, Université de Liège, Belgique*, 2000.
- [Kle2003] Kleinermann J.P., J.P. Ponthot, Parameter identification and shape/process optimization in metal forming simulation, *Journal og Materials Processing Technology*, Vol.139, p.521-526, 2003.
- [Kuh1997] Kuhn D., Une approche neuronale pour l'asservissement visuel d'un robot manipulateur, *Thèse de doctorat, Université de Haute Alsace*, 1997.

- [Leg1984] Legorju-jago K., Bathias C., Fatigue initiation and propagation in natural and synthetic rubbers, Vol.24, Issues 2-4, p.85-497, 2002.
- [Ler1993] Leriche R., Haftka R.T., Optimization of laminate stacking sequence for buckling load maximization by genetic algorithm, AIAA Journal, Vol.31, p.951-969, 1993.
- [Lin2003] Lin J.C., Using FEM and neural network prediction on hydrodynamic deep drawing of T-piece maximum length, Finite Elements in Analysis and Design, Vol.39, p.445-456, 2003.
- [Mal2003] Malinov S., Sha W., Software products for modelling and simulation in materials science, Computational Materials Science, Vol.28, 179-198, 2003.
- [Mar1984] Mars W. V., Fatemi A., A literature survey on fatigue analysis approaches for rubber, Vol.24, Issue 9, p.913-1012, 2002.
- [Mar2003] Martin A., Kuna M., Determination of deformation and failure properties of ductile materials by means of the small punch test and neural networks, Computational Material Science, Vol.28, p.633-644, 2003.
- [Mer1998] Mercier F., Contribution à la modélisation de l'emboutissage de tôles minces par approche inverse, Thèse de doctorat, Université de Technologie de Compiègne, France, 1998.
- [Mka2003] Mkaddem A., Experimentation et simulation du pliage de tôles H.L.E. Prevision du comportement en service des pièces pliées, These de doctorat, L'ENSAM d'Angers, France, 2003.
- [Moz2000] Mozolin M., Thill J.-C., Lynn Usery E., Trip distribution forecasting with multilayer perceptron neural networks : A critical evaluation, Transportation Research Part B, Vol.34, p.53-73, 2000.
- [Mul2002] Muliana A., Steward R., Haj-Ali Rami M., Saxena A., Artificial Neural Network an Finite Element Modeling of Nanoindentation Tests, Metallurgical and Materials Transactions, Vol.33A, 2002.
- [Nac2004] Naceur H., Delaméziere A., Batoz J.L., GuoY.Q., Knopf-Lenoir C., Some improvements on the optimum process design in deep drawing using the inverse approach, Journal of Materials Processing Technology, Vol.146, p.250-262, 2004.
- [Ner1993] Nerrand O., Roussel-Ragot P., Personnaz L., Dreyfus G., Neural networks and nonlinear adaptive filtering : unifying concepts and new algorithms, Neural Computation, Vol.5, p.165-199, 1993.
- [Ogd1984] Ogden R.W., Non-linear elastic deformations, Ellis Horwood Ltd., Chichester, UK, 1984.

- [Par1999] Park S.H., Yoon J.W., Yang D.Y., Kim Y.H., Optimum blank design in sheet metal forming by the deformation path iteration method, *International Journal of Mechanical Sciences*, Vol.41, p.1217-1232, 1999.
- [Per1999] Pernot S., Lamarque C.-H., Application of neural networks to the modelling of some constitutive laws, *Neural Networks*, Vol.12, p.371-392, 1999.
- [Pon2005] Ponthot J.P., Kleinermann J.P., Optimization methods for initial/tool shape optimisation in metal forming processes, *International Journal of Vehicle*, Vol.39(1/2), p.14-24, 2005.
- [Pon2006] Ponthot J.P., Kleinermann J.P., A cascade optimization methodology for automatic parameter identification and shape/process optimization in metal forming simulation, *Computer Methods in Applied Mechanics and Engineering*, Vol.195, Issues 41-43, p.5472-5508, 2006.
- [Pop1999] Popov E., *Engineering mechanics of solids*, Prentice-Hall, 1999.
- [Pre1992] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P., *Numerical Recipes, Art of Computing*, Cambridge University Press, 1992.
- [Ros1958] Rosenblatt F., The Perceptron : a probabilistic model for information storage and organization in the brain, *Psychological Review*, Vol.65, p.386-408, 1958.
- [Rou2002] Roux W. J., Stander N., Haftka R. T., Response surface approximations for structure optimisation. *International Journal for Numerical Methods in Engineering*, Vol.42, p.517-534, 2002.
- [Rum1986] Rumelhart D., Hinton G., Williams R., *Parallel Distributed Processing*, MIT Press, Vol.1, Cambridge, 1986.
- [San2005] Sanjay C., Neema M.L., Chin C.W., Modeling of tool wear in drilling by statistical analysis and artificial neural network, *Journal of Materials Processing Technology*, Vol.170, p.494-500, 2005.
- [Sch1997] Schuler Hydroforming, *Moderne Blechteilefertigung-neue Entwicklungen für das Karosserieziehen*, Geistlinger, L., Adam Opel AG, UKD Darmstadt, 1997.
- [Shi2003] Shin H.S., Pande G.N., Identification of elastic constants for orthotropic materials from a structural test, *Computers and Geotechnics*, Vol.30, p.571-577, 2003.
- [Sin2005] Singh S. K., Kumar D. R., Application of a neural network to predict thickness strains and finite element simulation of hydro-mechanical deep drawing, *International Journal of Advanced Manufacturing Technology*, Vol.25, p.101-107, 2005.
- [Tan1998] Tanaka A., Hirota K., Kaneko T., Virtual cutting with force feedback, *Proceeding of VRAIS*, 1998.

- [Ter2003] Tercej M., Perus I., Turk R., Suitability of CAE neural networks and FEM for prediction of wear on die radii in hot forging, *Tribology International*, Vol.36, p.573-583, 2003.
- [Tom2004] Tomonari F., Mark H., Accurate cyclic plastic analysis using a neural network material model, *Engineering Analysis with Boundary Elements*, Vol.28, p.195-204, 2004.
- [Tou1992] Touzet C., *Les réseaux de neurones artificiels : Introduction au connexionnisme : Cours, Exercices et travaux pratiques*, 1992.
- [Tro1993] Trompette P., Marcelin J. L., Schmeling C., Optimal damping of viscoelastic constrained beams or plates by use of a genetic algorithm, In *IUTAM, Zakopane Pologne*, 1993.
- [Was2001] Waszczyszyn Z., ziemianski L., Neural networks in mechanics of structures and materials : New results and prospects of applications, *computers and structures*, vol.79, p. 2261-2276, 2001.
- [Wat1987] Watous R. L., Learning algorithms for connectionist networks : Applied gradient methods of non linear optimisation. IEEE, *First International Conference On Neural Networks*, San Diego, California, 1987.
- [Xio2005] Xiong Y.S., Withers P.J., An evaluation of recurrent neural network modelling for the prediction of damage evolution during forming, *Journal of Materials Processing Technology*, Vol.170, p.551-562, 2005.
- [Yoo2000] Yoon J.W., Barlat F., Chung K., Pourboghrat F., Yang D.Y., Earing predictions based on asymmetric non quadratic yield function, *International Journal Of Plasticity*, Vol.16, p.1075-1104, 2000.
- [Yua2004] Yuan S., Wang L., Peng G., Neural network method based on a new damage signature for structural health monitoring, *Thin-Walled Structures*, Vol.43, Issue 4, p.553-563, 2005.
- [Zha1994] Zhang W.H., Fleury C., Two-point based sequential convex approximations for structural optimization, in : M. Hogge, E. Dick (Eds.), *3ème Congrès National Belge de Mécanique Théorique et Appliquée*, Université de Liège, 1994.
- [Zha1997] Zhang W.H., Fleury C., A modification of convex approximations methods for structural optimization, *Computers and Structures*, Vol.64, Issues 1-4, p.89-95, 1997.
- [Zmu2005] Zmudzki A., Kusiak J., Neural networks based optimisation in inverse analysis, *Proceeding of the first invited COST 526 conference in automatic process Optimisation in Materials Technology*, Morschach-Switzerland, p.236-241, 2005.

Liste de publications et de communications

1. Publications

- (a) Chamekh.A, Bel Hadj Salah.H, Gahbiche.A, Ben Amara.A, Dogui.A, "Identification of the Hardening Curve Using a Finite Element Simulation of the Bulge test", Journal of Steel and Related Material, pp.741-746, 2004.
- (b) Ridha Hambli, Abdessalem Chamekh, Hédi Bel Hadj Salah, "Real-time deformation of structure using finite element and neural networks in virtual reality applications, Finite Elements in Analysis and Design, Vol.42, p.985-991, 2006.
- (c) Chamekh A., BelHadjSalah H., Hambli R., Gahbiche A., "Inverse Identification using the bulge test and Artificial Neural Networks", Journal of Materials Processing, Vol.177, p.307-310, 2006.
- (d) Chamekh A., BelHadjSalah H., Hambli R., "Inverse technique identification of material parameters using finite element method and neural network computation", International Journal of Advanced Manufacturing Technology.

2. Communications

- (a) Chamekh.A, Bel Hadj Salah.H, Gahbiche.A, Ben Amara.A, Dogui.A, "Etude expérimentale et numérique de l'essai de gonflement hydraulique", Ecole de l'Aviation de Borj El Amri-Tunisie, JS'2003, 21-22 Mai 2003.
- (b) Chamekh.A, Bel Hadj Salah.H, Hambli.R, "Modélisation d'un essai d'emboutissage profond par éléments finis et réseau de neurones artificiels", Hammamet-Tunisie, CM-SM'2005, 23-25 Mars 2005.
- (c) Chamekh.A, Bel Hadj Salah.H, Hambli.R, "Modélisation de l'emboutissage par réseaux de neurones Artificiels", Casablanca-Maroc, 7ième congrès de Mécanique 2005, 19-22 Avril 2005.