



HAL
open science

Modélisation de la fiabilité des macro-composants mécanique à partir de modèles mathématiques et de réseaux de neurones, comparaisons des méthodes

Patrick Lyonnet, Rosario Toscano, P. Maisonneuve, Brice Lanternier

► To cite this version:

Patrick Lyonnet, Rosario Toscano, P. Maisonneuve, Brice Lanternier. Modélisation de la fiabilité des macro-composants mécanique à partir de modèles mathématiques et de réseaux de neurones, comparaisons des méthodes. 24. Machinery vibration seminar, Oct 2006, Montréal, Canada. pp.372-385. ineris-00973247

HAL Id: ineris-00973247

<https://ineris.hal.science/ineris-00973247v1>

Submitted on 4 Apr 2014

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.

MODELISATION DE LA FIABILITE DES MACRO-COMPOSANTS MECANIQUE A PARTIR DE MODELES MATHEMATiques ET DE RESEAUX DE NEURONES, COMPARAISONS DES METHODES

*P. Lyonnet*¹ *R. Toscano*¹ *P.L. Maisonneuve*^{1,2} *B. Lanternier*^{1,2}

¹ENISE/LTDS UMR 5513, Saint-Etienne, France

²INERIS, Saint-Etienne, France

RESUME

La fiabilité mécanique est un sujet qui intéresse la plupart des concepteurs et utilisateurs de systèmes techniques. Une des difficultés rencontrées est l'absence ou la pauvreté des modèles permettant de faire des études prévisionnelles correctes. Le taux de défaillances est souvent considéré comme constant, ce qui est manifestement faux en mécanique, d'où l'intérêt de l'élaboration d'outils, de modèles ou de méthodes plus adaptés. Notre présentation propose deux approches successives pour résoudre cette problématique, un modèle à taux de défaillances proportionnel et non constant, compléter par des coefficients tenant compte de l'environnement, du processus et de facteurs de conception, une modélisation à partir de réseaux de neurones. La comparaison des méthodes et leurs adéquations aux données de retour d'expériences s'appuie sur une application technique du domaine des industries chimiques (soupapes de sûreté).

1. INTRODUCTION A LA PROBLEMATIQUE DE LA MODELISATION FIABILITE DES DES MACRO-COMPOSANTS EN MECANQUES

La problématique de la modélisation des taux de défaillances des macro composants mécaniques, tel que des pompes, des soupapes, des moteurs pas à pas est ancienne. L'intérêt d'avoir un modèle pour la réalisation d'études prévisionnelles en phase de conception est évident. A l'instar des composants électronique, où nous trouvons des bases de données très élaborées, tel que la banque UTEC 80 810, la Mil HDBK 217, nous proposons une première approche s'appuyant sur des modèles statistiques régressif, régression linéaire à l'aide du modèle à hasard proportionnel, et régression non linéaire à l'aide de réseaux de neurones.

2. LOI DE FIABILITE POUR COMPOSANTS MECANQUES

Pour les composants mécaniques le taux de défaillances constant ne convient pas. Les modes de dégradations provoquent un taux croissant (usure, fatigue, corrosion). Le modèle de Weibull permet cette représentation. Il est d'une pratique aisé et universellement admise. Il couvre à lui seul l'ensemble des phénomènes de défaillance, c'est ce qui a motivé notre choix, sa forme est rappelée ci-après.

$$\lambda(t) = \frac{\beta}{\eta} \left(\frac{t - \gamma}{\eta} \right)^{\beta-1}$$

« β » est le paramètre de forme du modèle. Nous constatons que si $\beta < 1$ le taux de défaillances est décroissant, nous avons donc des pannes de jeunesse ; si $\beta = 1$ le taux est constant et si $\beta > 1$ le taux est croissant, panne de vieillesse ou de maturité en mécanique. « η » est le paramètre d'échelle et indique l'ordre de grandeur de la durée de vie moyenne.

« γ » est le paramètre de décalage, souvent il est égal à 0. Le modèle de Weibull ne peut à lui seul représenter l'ensemble des co-facteurs influents sur la fiabilité du macro composant, l'adjonction d'un modèle à hasard proportionnel sous forme de régression apporte une réponse qui devrait être plus adaptée.

2.1 Régression linéaire à l'aide du modèle à hasard proportionnel.

La représentation d'un taux de défaillances proportionnel permet d'introduire des facteurs cofacteurs représentant l'influence de l'environnement, du type de conception, du processus industriel. Le modèle proposé par D.R. Cox en 1970 pour le traitement de données binaires nous semble pertinent. Les premières applications de ce modèle ont eu lieu en bio-statistique, plus de 10 ans après. Il fut utilisé pour mesurer l'effet de traitements thérapeutiques sur des populations humaines. On trouve également quelques utilisations dans le cas des matériaux et des applications électriques. En fiabilité, ce modèle permet de prendre en compte plusieurs variables pour expliquer la durée de vie des mécanismes testés, sans donner de forme précise aux fonctions de fiabilité. D'autre part il faut noter que le paramétrage est entièrement statistique, il n'y a pas l'obligation de connaître et d'expliquer les phénomènes physiques. Il suffit de déterminer les coefficients b_i pour paramétrer le modèle. On utilise en générale la méthode du maximum de vraisemblance partielle due à DR Cox (méthode classique), cependant une alternative est donnée à partir de la régression multiple, que nous développons ci-après, car plus aisée à appliquer pour des exemples démonstratifs.

$$\lambda(t) = \frac{\beta}{\eta} \left(\frac{t - \gamma}{\eta} \right)^{\beta-1} \cdot \psi_c(z_c) \cdot \psi_f(z_f) \cdot \psi_e(z_e)$$

Les fonctions $\psi(z)$ représentent l'influence des trois facteurs complémentaires (conception, fabrication, environnement). Le modèle de $\psi(z)$ est de la forme :

$$\psi(z) = e^{b_1 z_1 + b_2 z_2 + b_3 z_3 + \dots + b_i z_i} = e^{b^T z}$$

L'évaluation des paramètres se fait à partir des échantillons provenant des essais, du retour d'expériences, d'une base de données interne. Les relations présentées précédemment conduisent aux modèles de fiabilité.

2.2 Estimation par la méthode de la régression multiple

$$\lambda(t, z) = \lambda(t, 0) e^{b_1 z_1 + b_2 z_2 + b_3 z_3 + \dots + b_i z_i + \dots + b_{ij} z_i z_j \dots}$$

$\ln(\lambda(t, z)) = \ln(\lambda(t, 0)) + b_1 z_1 + b_2 z_2 + \dots + b_{ij} z_i z_j$, qui conduit à une régression multiple :

$$Y = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_{ij} x_i x_j$$

$$Y = XB \text{ (Résultat classique pour la régression multiple)}$$

D'où la solution

$$\hat{B} = (X^T X)^{-1} X^T Y$$

3. REGRESSION NON LINEAIRE A L'AIDE DE RESEAUX DE NEURONES

Il est fréquent que les relations régissant les taux de défaillances des composants macro mécaniques soient complexes, et que les modèles de régressions linéaires ne conviennent pas ou soient trop imprécis. Un réseau de neurones définissant implicitement une fonction non linéaire multi variables paramétrables devrait apporter une réponse plus adéquate, c'est ce que nous développons ici.

$$y(t) = h(u(t), W)$$

Où y est la variable de sortie (i.e. la l'estimation de la durée de vie), u le vecteur d'entrée et W la matrice des paramètres. Nous verrons que par le choix d'une architecture de réseau adapté, et à condition de disposer d'une base de données suffisamment représentative des diverses conditions de fonctionnement du composant, il sera possible de paramétrer la fonction précédente, de façon a obtenir une bonne estimation (au sens d'un critère qui sera précisé) du taux de défaillances du composant considéré. Avant cela, il est nécessaire d'introduire différents concepts de base relatifs aux réseaux de neurones.

3.1 Définitions et propriétés des réseaux de neurones

Bien que d'inspiration biologique, le terme de neurone doit être compris ici comme un opérateur mathématique, et n'a que peu de liens avec le fonctionnement d'une cellule nerveuse réelle.

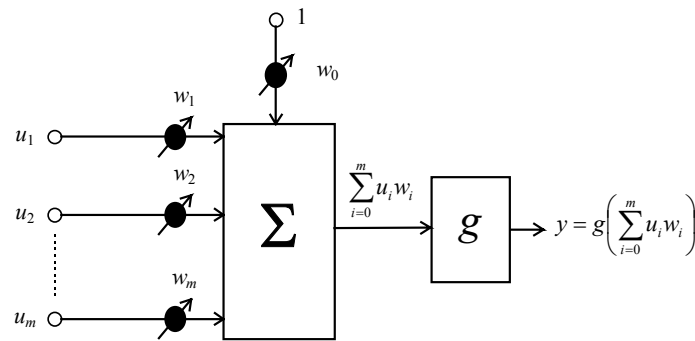
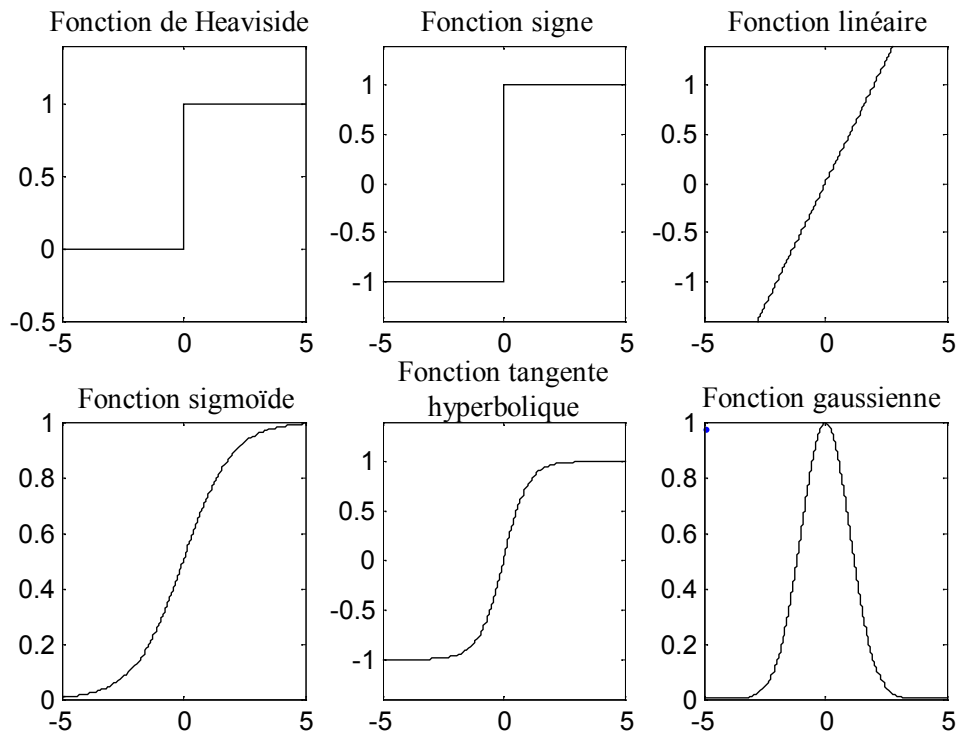


Figure 1. Neurone.

Un neurone (figure 1) réalise une fonction g , généralement non linéaire, de la somme pondérée de ses entrées u_i ($i=1\dots m$) :

$$\begin{cases} y = g(w^T u) \\ w^T = [w_0 \dots w_m], \quad x^T = [1 \ u_1 \dots u_m] \end{cases}$$

Les coefficients de pondération w_i , ou poids, sont les paramètres ajustables du neurone. La pondération w_0 , appelée biais, qui est connectée à une entrée constante et égale à un, permet d'introduire un décalage de la somme pondérée des entrées u_i .



Heaviside

$$g(\xi) = \begin{cases} 1 & \text{si } \xi \geq 0 \\ 0 & \text{si } \xi < 0 \end{cases}$$

Signe

$$g(\xi) = \begin{cases} 1 & \text{si } \xi \geq 0 \\ -1 & \text{si } \xi < 0 \end{cases}$$

Linéaire

$$g(\xi) = \xi = \sum_{i=0}^m u_i w_i$$

Sigmoïde

$$g(\xi) = \frac{1}{1 + e^{-\xi}}$$

Tangente hyperbolique

$$g(\xi) = \frac{e^{\xi} - e^{-\xi}}{e^{\xi} + e^{-\xi}}$$

Gaussienne

$$g(\xi) = \exp\left(-\frac{1}{2} \left(\frac{\xi - m}{\sigma}\right)^2\right)$$

Figure 2. Principales fonctions d'activation.

La figure 2 présente les principales fonctions utilisées pour g , encore appelée fonction d'activation du neurone.

Un neurone réalise une fonction non linéaire de la somme pondérée de ses entrées. L'intérêt des neurones réside dans les propriétés qui résultent de leur association en réseau. D'une manière générale, on appelle réseau l'interconnexion de plusieurs neurones formels. Un tel réseau permet alors la réalisation d'une fonction non linéaire de ses entrées par composition des fonctions réalisées par chacun de ses neurones. La manière dont sont connectés les neurones entre eux définit l'architecture du réseau. On distingue deux types d'architecture : les réseaux non bouclés et les réseaux bouclés.

3.1.1 Réseaux de neurones non bouclés

Un réseau de neurones non bouclé définit une fonction statique non linéaire. C'est-à-dire que les sorties du réseau à un instant « t » donné ne dépendent que des entrées

appliquées au même instant. Le comportement d'un réseau de neurones statique peut alors être décrit par la représentation suivante :

$$y(t) = h(u(t), W)$$

Où $u(t)$ est le vecteur des entrées, $y(t)$ le vecteur de sortie, h représente la fonction non linéaire réalisée par le réseau et W est la matrice des poids des connexions. La figure 3 présente un réseau de structure particulière, très couramment utilisé, appelé réseau en couches ou perceptron multicouche. On trouve les entrées du réseau, une ou plusieurs couches de neurones intermédiaires, dits neurones cachés et des neurones de sortie. Les neurones de la couche cachée ne sont pas connectés entre eux, ils rendent possible la résolution de problèmes non séparables linéairement. La fonction d'activation des cellules de la couche de sortie est généralement linéaire.

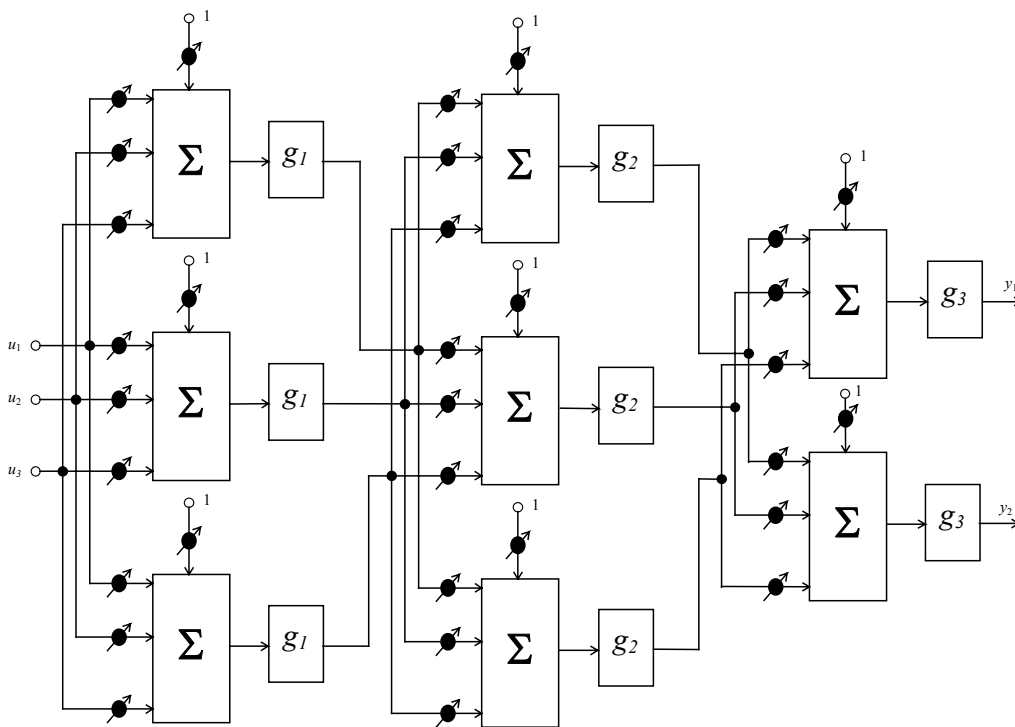


Figure 3. Réseau de neurones multicouche.

Les réseaux non bouclés sont généralement utilisés pour l'approximation de fonctions non linéaires, c'est le cas de notre application en fiabilité.

3.1.2 Réseaux de neurones bouclés

Un réseau de neurones bouclé à temps discret définit une, ou plusieurs, équations aux différences non linéaires, par composition des fonctions réalisées par chacun des neurones et des retards associés à chacune des connexions. En d'autres termes, les sorties du réseau à un instant donné dépendent non seulement des entrées appliquées au même instant mais aussi des entrées et des sorties passées. De tels réseaux sont alors être utilisés pour la modélisation de systèmes dynamiques non linéaires.

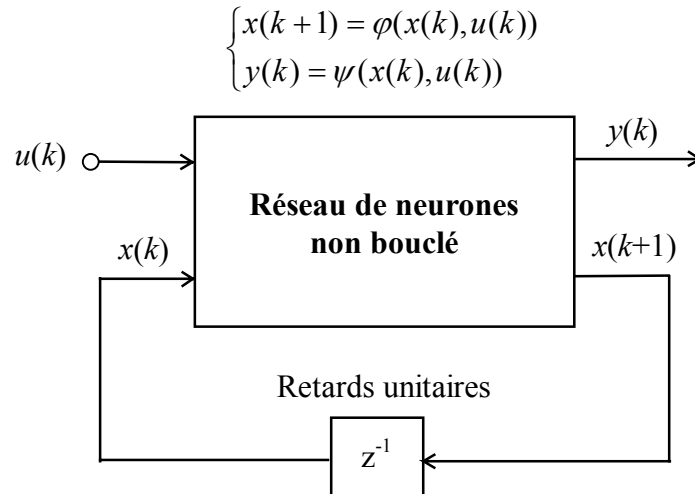


Figure 4. Réseau de neurones bouclé.

On montre que tout réseau de neurones bouclé, aussi complexe soit-il, peut toujours être mis sous la forme canonique de la figure 4, c'est à dire, comportant un réseau de neurones non bouclé, dont certaines sorties, appelées variables d'état, sont ramenées aux entrées par des bouclages unitaires.

3.2 Apprentissage des réseaux de neurones

D'un point de vue pratique, les réseaux de neurones sont utilisés pour réaliser l'approximation de fonctions non linéaires inconnues. Ceci est rendu possible en vertu du résultat suivant [8] :

Toute fonction bornée suffisamment régulière peut être approchée uniformément, 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.

Toutefois, cette propriété d'approximation universelle n'est pas propre aux seuls réseaux de neurones. C'est la propriété de parcimonie qui leur est particulière et fait tout leur intérêt. En effet, on montre [3] que, si l'approximation dépend des paramètres ajustables de manière non linéaire (ce qui est le cas des réseaux de neurones), alors elle est plus parcimonieuse que si elle dépend linéairement des paramètres (ce qui est le cas des polynômes). En d'autres termes, l'approximation par réseaux de neurones nécessite, à précision égale, moins de paramètres que tout autres approximateurs linéaires en les paramètres.

Le problème de l'apprentissage d'un réseau, se pose généralement de la façon suivante : étant donné un ensemble de mesures entrées/sorties réalisées sur un processus de nature quelconque, on suppose qu'il existe une relation déterministe entre les entrées/sorties mesurées. Le problème est alors de rechercher une forme mathématique de cette relation, valable dans le domaine où les mesures ont été réalisées. En d'autres termes, il s'agit de construire un modèle entrées/sorties du processus étudié, à partir des mesures disponibles. Ceci peut être obtenu en paramétrant la fonction non linéaire définie par un réseau de neurones de façon à ce qu'elle approche au mieux la relation inconnue. Ce paramétrage comprend deux parties distinctes : le réglage structurel et le réglage paramétrique.

3.2.1 Réglage structurel

Le réglage structurel consiste à établir la structure du réseau, c'est-à-dire : le nombre d'entrées, le nombre de couches, le nombre de neurones par couche et enfin le choix des fonctions neuronales. Il n'existe pas de méthode systématique quant au choix structurel du réseau, la structure est alors généralement établie par essais et corrections successifs. Signalons toutefois qu'une seule couche cachée est généralement adoptée, car il a été montré que les réseaux comportant au moins une couche cachée sont dotés de la propriété d'approximation universelle.

3.2.2 Réglage paramétrique

Il consiste à déterminer les poids du réseau de façon à approcher au mieux la relation non linéaire inconnue, c'est la phase d'apprentissage proprement dite.

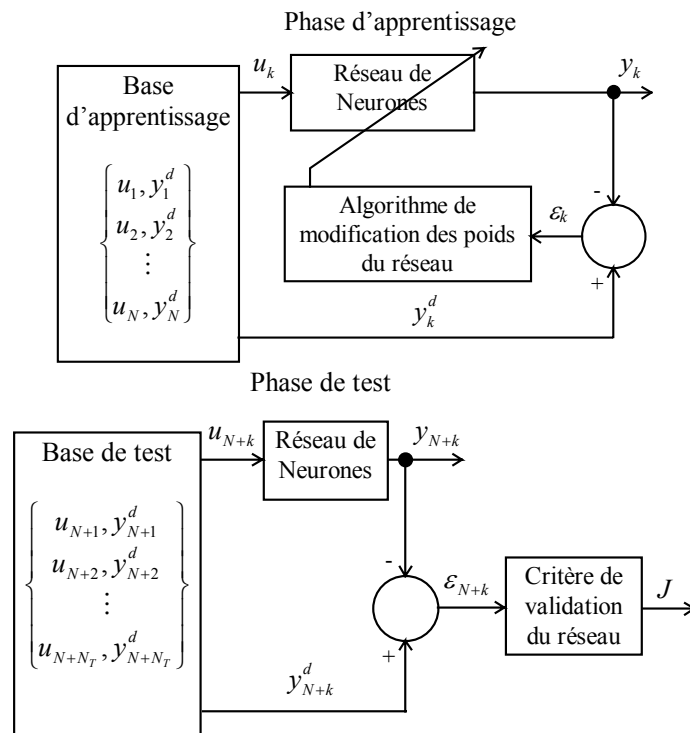


Figure 5. Phases d'apprentissage et de test.

L'apprentissage d'un réseau de neurones pour l'approximation d'une fonction non linéaire inconnue peut être décomposé en trois phases distinctes : la création d'une base (ou fichier) d'apprentissage, l'apprentissage proprement dit du réseau et enfin une phase de test permettant la validation du réseau. Le fichier d'apprentissage est créé à partir de l'ensemble des données disponibles, issu du phénomène à modéliser. La phase d'apprentissage (cf. figure 5) est le procédé de calcul des poids des connexions entre chaque neurone pour qu'à chaque entrée donnée on obtienne en sortie la réponse désirée. Le réseau est alors fixé, on ne le modifie plus.

La phase de test (cf. figure 5) consiste à appliquer au réseau des exemples n'ayant pas été utilisés lors de la phase d'apprentissage du réseau. Cet ensemble est appelé le fichier test, il permet d'évaluer les capacités de généralisation du réseau. Pour une étude plus détaillée concernant l'apprentissage des réseaux de neurones, on pourra consulter [4], [7].

4. ADEQUATION DE NOS MODELES DANS LE CADRE DE LA FIABILITE DES MACRO-COMPOSANTS

Pour cette étude, deux méthodes vont être utilisées successivement pour des soupapes de sécurité dont les taux de défaillance sont connues et rassemblés dans une base de données. Ces données permettront de modéliser le taux de défaillances, puis de vérifier l'adéquation du modèle, pour les deux méthodes.

4.1 Base de donnée NSWC

La base de données NSWC, est issue de la Marine américaine. C'est un modèle de prédiction de taux de défaillance pour des soupapes de sécurité. Il peut être utilisé pour le développement d'équipements mécaniques et fournir une estimation de la fiabilité pour une conception novatrice ou une évolution de produit. Le modèle est prévu pour utiliser les caractéristiques mécaniques des soupapes mais aussi leur environnement. Le taux de défaillance d'une soupape de sécurité est donc déterminé en fonction de ses paramètres intrinsèques et extrinsèques, le résultat est le produit d'un taux de défaillance de base et de plusieurs facteurs modélisant l'effet d'un paramètre appartenant à la soupape de sécurité. Pour une soupape tout ou rien, le taux de défaillance est déterminé comme suit :

$$\lambda_{PO} = \lambda_{PO,B} \times C_p \times C_Q \times C_F \times C_V \times C_N \times C_S \times C_{DT} \times C_{SW} \times C_W$$

Avec : λ_{PO} = Taux de défaillance d'une soupape de sécurité tout ou rien en panne/million d'opération

$\lambda_{PO,B}$ = Taux de défaillance de base d'une soupape de sécurité, 1.40 panne/million d'opération

C_p = Facteur multiplicatif pour modéliser l'effet de la pression du fluide sur le taux de défaillance de base

C_Q = Facteur multiplicatif pour modéliser l'effet des fuites permises sur le taux de défaillance de base

C_F = Facteur multiplicatif pour modéliser l'effet de l'état de surface sur le taux de défaillance de base

C_V = Facteur multiplicatif pour modéliser l'effet de la viscosité du fluide sur le taux de défaillance de base

C_N = Facteur multiplicatif pour modéliser l'effet de la pollution du fluide sur le taux de défaillance de base

C_S = Facteur multiplicatif pour modéliser l'effet de l'effort de fermeture sur le taux de défaillance de base

C_{DT} = Facteur multiplicatif pour modéliser l'effet de la taille du clapet sur le taux de défaillance de base

C_{SW} = Facteur multiplicatif pour modéliser l'effet de la taille du siège sur le taux de défaillance de base

C_W = Facteur multiplicatif pour modéliser l'effet du débit de fluide sur le taux de défaillance de base

La détermination de ces facteurs est décrite dans la méthode NSWC et nécessite des connaissances sur les soupapes et leur environnement.

4.2 Fabrication d'une base de donnée

Un algorithme est réalisé pour pouvoir calculer le taux de défaillance rapidement en entrant les caractéristiques de la soupape. Il permet de représenter plusieurs types de soupape dans plusieurs environnements. Le traitement des données s'effectuera sur une série de 500 soupapes différentes avec chacune leur taux de défaillance en panne/million d'opération et leurs différentes caractéristiques (pression d'entrée et de sortie, état de surface, fuite acceptable, géométrie du clapet, etc.)

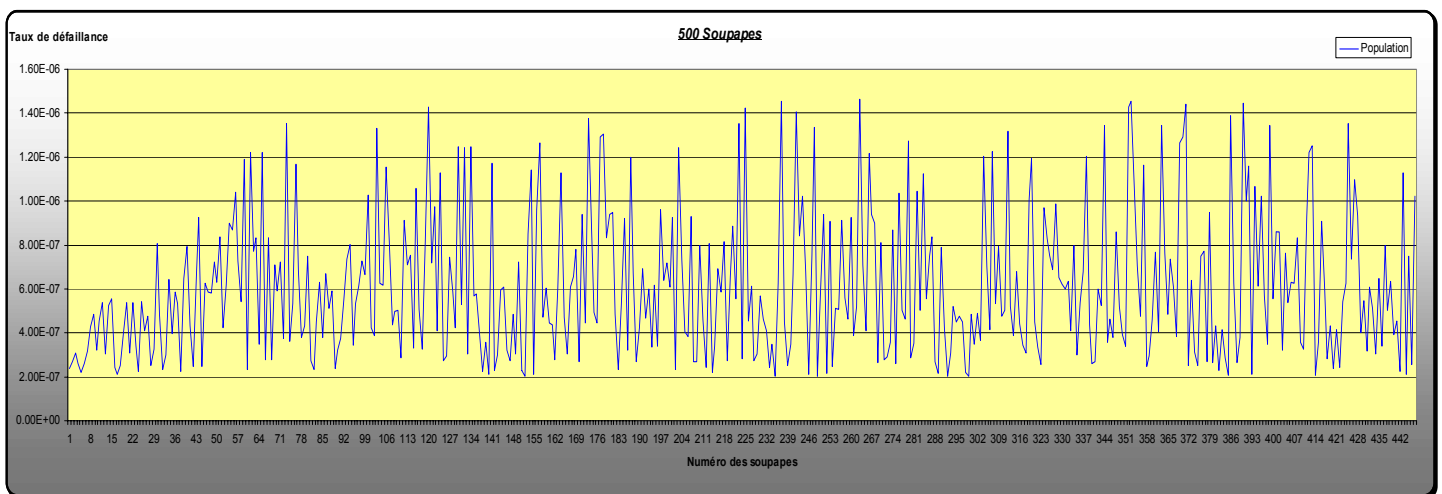


Figure 6. Population de la base de donnée.

La base de donnée utilisée pour la méthode de Cox et les réseaux de neurones doit comprendre des entrées (les différents paramètres intrinsèques et extrinsèques à la soupape) et une sortie (le taux de défaillance). Leur choix est fonction de la facilité à connaître la valeur de l'entrée. Ici, quatre paramètres sont retenus pour notre étude:

- la pression d'entrée de la soupape
- la pression de sortie de la soupape
- la viscosité du fluide
- le diamètre d'entrée de la soupape

4.2.1 Codage de la base de donnée pour la méthode Cox

Pour la méthode Cox, il est nécessaire de classer les données d'entrée ou les variables indépendantes. Ces cinq paramètres ont été classés par famille pour réaliser une base de donnée codée. Voici le codage réalisé :

- la pression d'entrée : Z_{PE}
 - o Base pression = 1 (<145 psi)
 - o Pression moyenne = 2
 - o Haute pression = 3 (>725 psi)
- la pression de sortie : Z_{PS}
 - o Atmosphère = 1 (14 psi)
 - o Autre = 2

- la viscosité du fluide. Trois types de fluide : Z_V
 - o Ammoniac = 2
 - o Azote = 3
 - o Ethylène = 1
- le diamètre nominal d'entrée : Z_{DNE}
 - o petit = 1 (< 50 mm)
 - o moyen = 2
 - o grand = 3 (> 150 mm)

Si dessous, un extrait de la base de donnée issue de la méthode de prédiction NSWC avec le codage par famille. Chaque ligne représente une soupape de sécurité et les différentes colonnes sont les paramètres codés nécessaires à leur classement. La deuxième partie représente les entrées appliquées au réseau de neurone. Le codage mis en place pour Cox n'a pas été utilisé pour le réseau de neurones car les entrées sont numériques et donc bien adapté à l'algorithme de calcul des poids et biais des neurones.

	Résultats nswc	Codage pour Cox				Pas de codage pour le réseau de neurones			
	calcul du taux de défaillance pour une soupape tout ou rien	PRESSION ENTREE	PRESSION SORTIE	VISCOSITE	DN ENTREE	PRESSION ENTREE (PSI)	PRESSION SORTIE (PSI)	VISCOSITE (lb-min/in ²)	DN ENTREE (inch)
170	2.69083E-07	2	1	2	3	456.49	14.00	1.77E-03	8.70
171	9.37602E-07	3	1	2	3	843.64	14.00	1.72E-03	8.86
172	4.43366E-07	2	1	2	3	584.68	14.00	1.72E-03	10.60
173	1.37585E-06	3	1	2	3	983.34	14.00	1.72E-03	6.73
174	8.71816E-07	3	1	2	2	789.12	14.00	1.77E-03	2.19
175	4.91568E-07	3	1	3	2	784.63	14.00	2.99E-03	3.30
176	4.45985E-07	3	1	3	3	764.46	14.00	2.99E-03	7.66
177	1.29306E-06	3	1	2	2	983.72	14.00	1.77E-03	3.97
178	1.30355E-06	3	1	2	2	949.60	14.00	1.72E-03	4.45
179	8.32752E-07	3	1	2	2	763.77	14.00	1.72E-03	2.38
180	9.3736E-07	3	1	2	3	855.79	14.00	1.77E-03	9.22
181	9.46183E-07	3	1	2	2	813.16	14.00	1.72E-03	4.88
182	4.85304E-07	3	1	3	2	788.48	14.00	2.99E-03	4.56
183	2.34533E-07	2	1	2	3	424.62	14.00	1.77E-03	6.04
184	5.28385E-07	3	1	3	2	810.28	14.00	2.99E-03	4.08
185	9.2108E-07	3	1	2	3	836.79	14.00	1.77E-03	6.64
186	3.22608E-07	2	1	3	2	627.01	14.00	2.99E-03	2.40
187	1.19847E-06	3	1	2	3	936.54	14.00	1.77E-03	8.71
188	6.72578E-07	2	1	2	2	695.25	14.00	1.72E-03	5.38
189	2.70817E-07	2	1	3	3	590.10	14.00	2.99E-03	10.95
190	4.25068E-07	2	1	2	3	575.13	14.00	1.77E-03	9.18
191	6.93841E-07	2	1	2	3	724.09	14.00	1.72E-03	9.92
192	4.68921E-07	3	1	3	3	786.70	14.00	2.99E-03	8.24
193	6.0137E-07	2	1	2	2	679.19	14.00	1.77E-03	5.05
194	3.32983E-07	2	2	2	3	718.41	217.55	1.77E-03	9.60
195	6.1716E-07	2	1	2	3	686.85	14.00	1.72E-03	10.12
196	3.4079E-07	2	1	3	2	643.41	14.00	2.99E-03	5.75

Figure 7. Extrait de la base de donnée.

4.2 Traitement des données et comparaisons

Pour vérifier l'adéquation des modèles, trois modèles de fiabilité sont utilisés. Un modèle grossier de Weibull qui représente la loi de fiabilité de l'échantillon connu, puis un modèle de Cox qui prend en compte l'environnement des soupapes de sécurités.

4.2.1 Notations

Dans la suite nous utiliserons les notations suivantes :

T_i^{NSWC} = L'inverse du taux de défaillance calculé par la NSWC, temps de bon fonctionnement des soupapes de sécurités.

T_i = Temps calculé par la loi de Weibull.

$T_{(i,0)}$ = Temps de base sans la partie environnementale calculés pour le modèle de Cox.

λ_i^{NSWC} = Taux de défaillance calculé par la méthode NSWC.

λ_{MOY} = Taux de défaillance du modèle de Weibull.

$\lambda_{(i,Z)}$ = Taux de défaillance du modèle de Cox.

β^{NSWC} et η^{NSWC} = paramètre de forme et d'échelle de la loi de Weibull associés aux taux de défaillance de l'échantillons des soupapes NSWC.

β^{Cox} et η^{Cox} = paramètre de forme et d'échelle de la loi de Weibull associés aux taux de défaillance de l'échantillons des soupapes NSWC multiplié par la fonction environnementale de Cox.

4.2.2 Modèle Weibull

$$\lambda_{(t)} = \frac{\beta}{\eta} \times \left(\frac{T_i}{\eta} \right)^{\beta-1}$$

Pour ce modèle, les paramètres de Weibull, β et η sont déterminés à partir des taux de défaillance de la base de donnée calculés grâce à la méthode NSWC. Pour les calculs, le logiciel Statistica V7 donne les résultats suivants :

$$\beta^{NSWC} = 2,018 \text{ et } \eta^{NSWC} = 2\,390\,000$$

Grâce à au paramètre β^{NSWC} , il est possible de calculer l'espérance mathématique de durée de fonctionnement $E(t)$ de la base de donnée NSWC.

$$E_{(t)} = \gamma + \eta \Gamma \left(1 + \frac{1}{\beta} \right)$$

Dans notre étude, $\gamma = 0$ donc nous pouvons calculer :

$$E_{(t)} = \eta^{NSWC} \times \Gamma \left(1 + \frac{1}{\beta^{NSWC}} \right) = 2390000 \times \Gamma \left(1 + \frac{1}{2,018} \right) = 2117898.5$$

Il est donc possible, maintenant, de connaître un taux de défaillance constant pour la base de donnée NSWC. Ce taux, λ_{MOY} permet de vérifier l'adéquation de ce premier modèle avec la réalité représenté par notre population NSWC.

$$\lambda_{MOY} = \frac{1}{E(t)} = 4.72 \cdot 10^{-07}$$

L'adéquation du modèle est validée par la moyenne sur l'échantillon d'un critère des moindres carrés :

$$\varepsilon = \frac{1}{n} \left(\sum_{i=1}^n \left[\left(\lambda_{(i)}^{NSWC} - \lambda_{MOY} \right)^2 \right] \right)$$

La comparaison des trois méthodes se fera grâce à ce critère. Pour le modèle de Weibull, le critère des moindres carrés à pour valeur $1.43 \cdot 10^{-13}$.

4.2.3 Modèle de Cox

$$\lambda_{(t,Z)} = \frac{\beta}{\eta} \times \left(\frac{Ti}{\eta} \right)^{\beta-1} \times e^{BZ}$$

Calcul tout d'abord les $T_{(i,0)}$. Les taux de défaillance de la méthode NSWC peuvent être considérés comme la multiplication d'un taux de base et d'un facteur environnementale. Les taux de défaillance peuvent donc s'écrire comme :

$$T_i^{NSWC} = T_{(i,0)} \times \left(e^{BZ} \right)^{1/\beta}$$

On peut donc tirer :

$$T_{(i,0)} = \frac{T_i^{NSWC}}{\left(e^{bZ} \right)^{1/\beta}} \text{ avec } \beta = \beta^{NSWC} = 2,018.$$

Pour la partie environnementale, les coefficients de Cox (b_i) ont été calculé par Matlab à partir des taux de défaillance de la méthode NSWC et sont :

FACTEUR de COX			
b_{PE}	b_{PS}	b_V	b_{DNE}
0.8617	-0.5826	-0.4354	-0.0637

Figure 8. Facteurs environnementaux de Cox.

De cette nouvelle liste de taux de défaillance $T_{(i,0)}$, les paramètres de Weibull associé, Béta et Etat, sont calculé pour le modèle de Cox :

$$\beta^{cox} = 2,57 \text{ et } \eta^{cox} = 2\ 670\ 694$$

Pour cette nouvelle liste de taux de défaillance, il est possible, comme il a été fait pour la loi de Weibull, de calculer l'espérance mathématique de durée de fonctionnement.

$$E_{(t)} = \eta^{cox} \times \Gamma\left(1 + \frac{1}{\beta^{cox}}\right)$$

$$E_{(t)} = 2670694 \times \Gamma\left(1 + \frac{1}{2,57}\right)$$

$$E_{(t)} = 2371576$$

Donc :

$$\lambda_{(i,0)} = 4.22 \times 10^{-7}$$

et

$$\lambda_{(i,Z)} = \lambda_{(i,0)} \times e^{(bZ)}$$

Le critère des moindres carré entre le taux de défaillance NSWC et ceux du modèle de Cox a comme valeur : 3.12×10^{-14} .

4.2.4 Réseaux de neurones

Pour les réseaux de neurones, une base d'apprentissage tirée de la NSWC a été utilisée. Pour la simulation, la même base de donnée donc les même taux de défaillance que pour Cox ont servit à l'évaluation de la réponse du système. La base de donnée n'a pas été codée, les entrées du réseau de neurones sont donc brutes.

Le réseau est composé de 2 couches, une couche cachée de 20 neurones avec la fonction d'activation « tansig », tangente hyperbolique, et une couche d'un neurone avec la fonction d'activation « pureline », fonction linéaire. Il est composé de 5 entrées correspondantes au classement vu plus haut et la sortie est un taux de défaillance.

Le critère des moindres carré a été calculé comme pour Cox et Weibull. La réponse du réseau a été comparée au taux de défaillance de la NSWC. Ceci donne un critère des moindres carré de : $9,86 \cdot 10^{-16}$.

5. CONCLUSION.

En conclusion nous pouvons analyser et comparer les résultats et en tirer un premier enseignement, c'est ce qui est fait à partir du tableau extrait de notre étude ci-dessous. L'évaluation de la performance des trois méthodes est fait à partir de la moyenne d'un critère des moindres carrés. Nous constatons que le modèle de Cox améliore de façon significative celui de Weibull grâce à la fonction environnementale. Le réseau de neurones possède le meilleur critère de moindres carrés, voir le graphique de comparaisons des réponses (ci-après) pour chaque modèle. Cependant le réseau de neurones ne propose pas

un modèle descriptif interprétable, il se comporte comme une « boîte noire », nous pouvons dire que c'est cela son principal défaut.

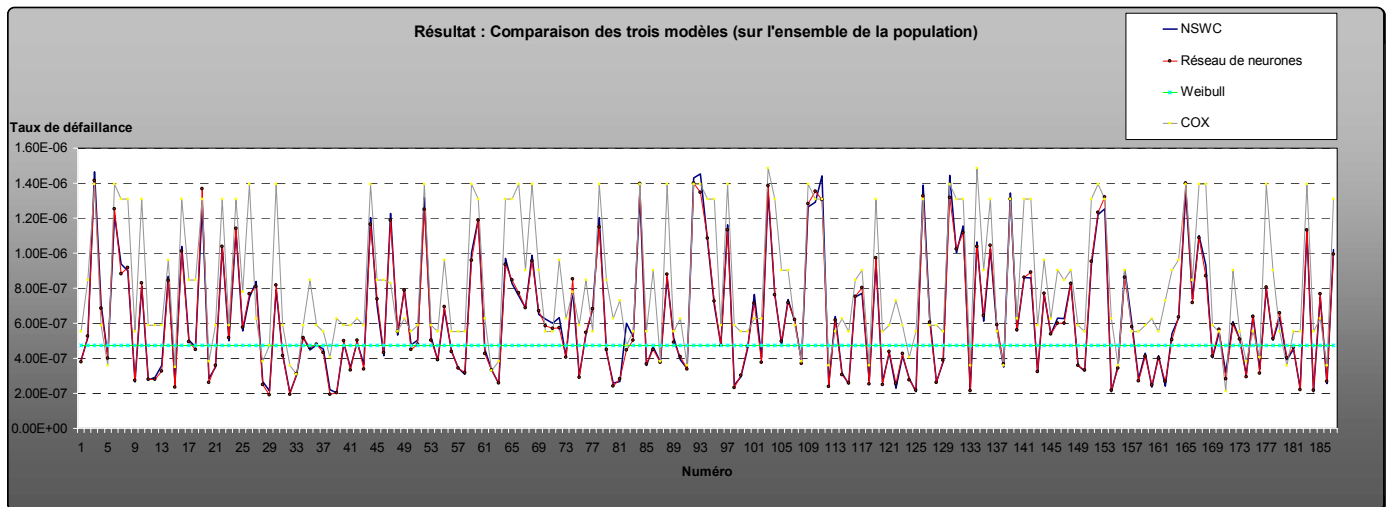


Figure 9. Courbe réponse des différents modèles.

BIBLIOGRAPHIE

- [1] **Augé J.C., Lallement J., Lyonnet P.**, 1997. Reliability of mechanical components accelerated testing and advanced statistical methods. *International conference on safety & reliability, ESREL '97*. Lisbonne, pp 1859-1867.
- [2] **Bagdonavicius, V., Nikulin, M.**, 1995. On accelerated testing of systems, *Européen Journal on Diagnostic and Safety in Automation*, vol. 5 n°3 : pp 307-316.
- [3] **A.R. Barron**. Universal Approximation Bounds for Superpositions of a Sigmoidal Function. *IEEE Trans. Information Theory*, vol. 39, N°3, PP. 930-945, 1993.
- [4] **C. Bishop**. *Neural networks for pattern recognition*. Oxford University Press, 1995.
- [5] **Birolini A.**, 1997. *Quality and reliability of technical systems*. Edition Springer.
- [6] **Desroches A.** 1995. *Concepts et méthodes probabilistes de base de la sécurité*. Edition Tec et Doc Lavoisier, Paris.
- [7] **J. Hertz, A. Krogh, R.G. Palmer**. *Introduction to the theory of neural computation*. Addison-Wesley Publishing Company, 1990.
- [8] **K. HORNIC, M. STINCHCOMBE, H. WHITE, P. AUER**. Degree of approximation results for feedforward networks approximating unknown mappings and their derivatives. *Neural Computation*, Vol 6, PP. 1262-1275, 1994.
- [9] **Lyonnet P.** Ingénierie de la fiabilité, édition Tec et Doc Lavoisier 2006 Paris
- [10] **Ringler J.**, 1996 Précis de probabilités et de statistiques à l'usage de la fiabilité. Collection ISDF édition Octares.