

# MODELES NEURONAUX POUR L'ESTIMATION, LA COMMANDE ET LA SIMULATION

## Vers de nouvelles stratégies non-linéaires pour la synthèse et la validation des lois de commande de vol

Fabien LAVERGNE<sup>1</sup> (AIRBUS France / LAAS-CNRS)

Directeur(s) de thèse: Fabrice VILLAUME (AIRBUS-France) et Félix MORA-CAMINO (ENAC)

Laboratoire d'accueil:

LAAS-CNRS (groupe DISCO)  
Toulouse (France)

Etablissement d'inscription:

Université Paul Sabatier  
Toulouse (France)

---

### Résumé

*Historiquement, les lois de commandes de vol s'appuient sur les techniques de commande linéaires. Aujourd'hui, l'émergence de lois de commande non-linéaires, basées notamment sur l'inversion de la dynamique, crée un besoin nouveau en termes de modèles non-linéaires multivariables et embarquables.*

*Depuis plus de vingt ans les outils d'identification non-linéaires tels que les réseaux de neurones ont acquis une certaine maturité. Cet article présente un outil d'identification automatique par réseaux de neurones qui pourrait permettre d'optimiser le processus de synthèse de lois de commande de vol et de leur validation dans le cadre non-linéaire.*

### Mots clés

*Commande non-linéaire, inversion de la dynamique, réseaux de neurone, identification, commandes de vol, estimation, simulation.*

---

## 1 INTRODUCTION

Avec l'automatisation de plus en plus poussée des commandes de vol dans l'aéronautique commerciale, un besoin croissant de modèles numériques se fait sentir. Ces modèles peuvent être utilisés à différentes fins :

- l'estimation,
- la commande,
- la simulation.

S'agissant de modèles potentiellement embarquables, ils doivent donc impérativement posséder les caractéristiques suivantes :

- compacité (taille mémoire limitée),
- simplicité (pour l'exécution temps réel),
- précision,

---

<sup>1</sup> Fabien.lavergne@airbus.com

- et surtout la fiabilité.

Les systèmes de commande de vol, dont la fiabilité se mesure en nombre de défaillances par heure de vol, sont des applications critiques. Il est donc nécessaire de connaître la probabilité d'occurrence d'erreurs de modélisation supérieures à un seuil critique (nommés ci-après « points aberrants »).

Dans la première partie de cet article, deux des outils d'identification non-linéaires les plus utilisés sont présentés et comparés: les polynômes interpolateurs et les réseaux de neurones.

Dans une deuxième partie, on décrit un outil générique d'identification non-linéaire répondant au cahier des charges présenté précédemment.

## 2 TECHNIQUES D'IDENTIFICATION NON-LINEAIRE

Deux solutions s'affrontent ici : l'identification par polynômes interpolateurs ou celle par réseaux de neurones. Des critères précis (présentés en introduction) sont susceptibles de les départager.

### 2.1. IDENTIFICATION PAR POLYNOMES INTERPOLATEURS

#### 2.1.1. Principe

Il s'agit d'approximer une courbe connue en quelques points expérimentaux par un polynôme d'ordre quelconque (à déterminer par l'utilisateur). C'est la méthode d'identification la plus simple et la plus courante. Voici le type de travail qu'il faut réaliser :

- 1°/ Collecte des données en nombre « suffisant »,
- 2°/ Mise en forme

3°/ Interpolation en augmentant successivement l'ordre du polynôme jusqu'à ce que l'approximation apparaisse « suffisante ».

Tout le problème est de déterminer le sens de « suffisant ». L'expérience de l'opérateur joue ici un rôle prépondérant. Dans l'exemple qui suit, on s'occupe de l'effet de sol affectant la contribution du braquage des spoilers au coefficient de tangage d'un avion de transport de type Airbus (CmLD).

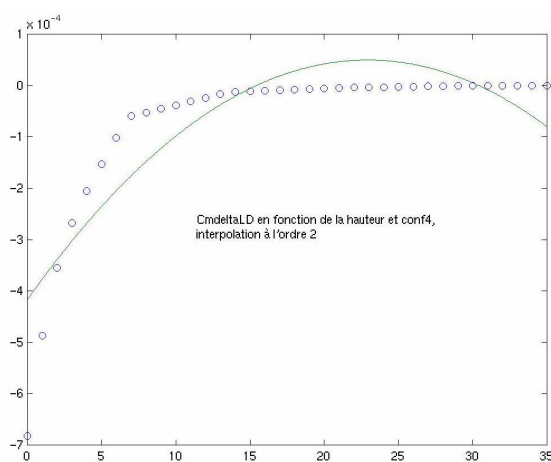


Fig.1.  $Cm_{LD}$  en fonction de la hauteur en conf. 4  
- Interpolation à l'ordre 2

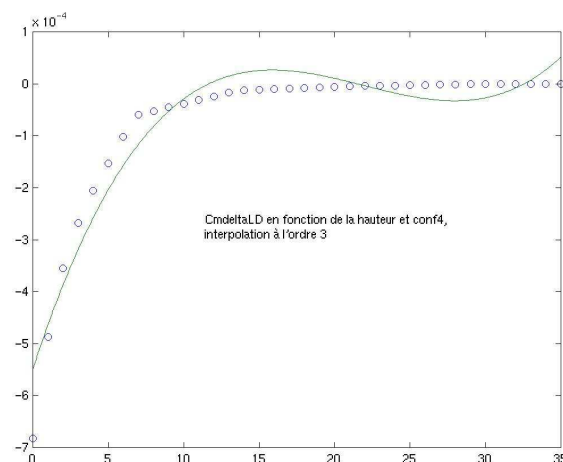


Fig.2.  $Cm_{LD}$  en fonction de la hauteur en conf. 4  
- Interpolation à l'ordre 3

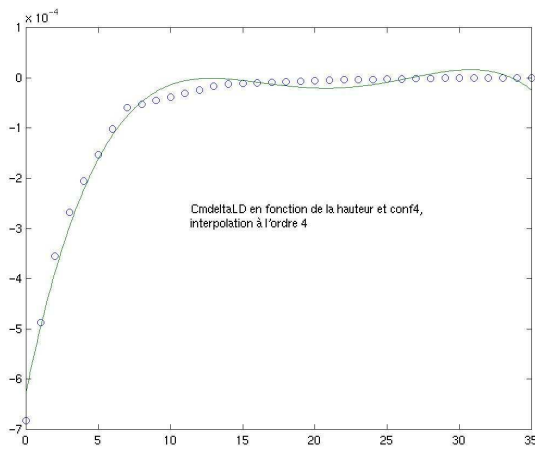


Fig.3.  $C_{mLD}$  en fonction de la hauteur en conf. 4  
- Interpolation à l'ordre 3

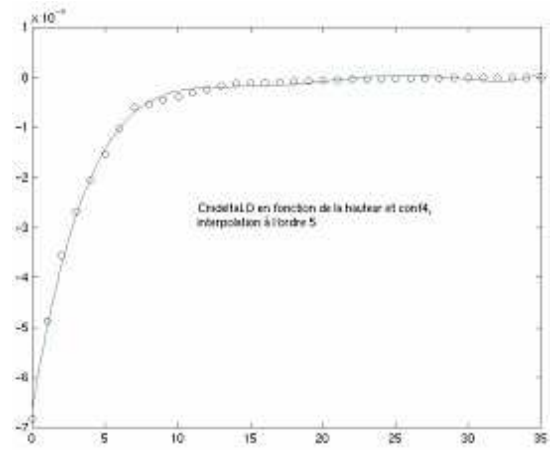


Fig.4.  $C_{mLD}$  en fonction de la hauteur en conf. 4  
- Interpolation à l'ordre 5

### 2.1.2. Application

Partant de ce principe, une identification numérique complète des coefficients aérodynamiques d'un avion a pu être réalisée. A noter qu'un coefficient aérodynamique est une fonction non-linéaire multivariable donnant les caractéristiques aérodynamiques de l'objet auquel il se rapporte. Ceci a permis de souligner les nombreuses faiblesses de la technique d'interpolation polynomiale :

- Elle fait appel à une modélisation formelle simplifiée afin de pouvoir tourner en temps réel. La rapidité est donc mieux assurée que la précision.
- L'identification multivariable est délicate car on peut difficilement présumer de la forme mathématique du couplage entre plusieurs variables.
- Le processus ne peut pas être automatisé. Ainsi, tout changement de caractéristiques aérodynamiques de l'avion (jusqu'à plusieurs fois par mois en phase de développement d'un programme) rend le travail déjà effectué obsolète.

### 2.1.3. Performances

|              |    | Polynômes interpolateurs        |                       |                   |                                    |
|--------------|----|---------------------------------|-----------------------|-------------------|------------------------------------|
|              |    | Pourcentage de points aberrants | Erreur moyenne (en %) | Ecart-type (en %) | Encombrement mémoire (en octets)   |
| Conf. 4      | Cm | 0                               | -0.5259               | 2.8251            | 2748                               |
|              | Cx | 9.9                             | -0.4538               | 12.3227           | 915                                |
|              | Cz | 0.4                             | -0.0191               | 4.2118            | 2379                               |
| Conf. 5      | Cm | 0.1                             | -0.2459               | 3.3961            | 2864                               |
|              | Cx | 0.6                             | 0.8630                | 7.2505            | 1021                               |
|              | Cz | 0.1                             | -0.2195               | 3.0956            | 2433                               |
| <b>Total</b> |    |                                 |                       |                   | 6042 en conf. 4<br>6318 en conf. 5 |

Tableau. 1. Performance de l'identification par polynômes interpolateurs pour différentes configurations aérodynamiques (braquage des becs et volets hypersustentateurs) de l'A321

## 2.2. IDENTIFICATION PAR RESEAUX DE NEURONES

### 2.2.1. Définition d'un neurone formel

Un réseau de neurones est une structure formée de plusieurs éléments de calcul interconnectés, appelés neurones formels ou mathématiques. Un neurone mathématique est une modélisation simplifiée d'un neurone humain. Ces deux types de neurones réagissent à des signaux d'entrée.

Plus précisément, un neurone mathématique produit une réponse non-nulle si la somme pondérée des signaux d'entrée est supérieure ou égale à son seuil d'activation. L'amplitude de cette réponse est modulée par une fonction d'activation. Un neurone mathématique est donc caractérisé par le nombre de ses entrées et leurs pondérations (poids synaptiques), son seuil, sa fonction d'activation et le nombre de ses sorties. On peut le décrire par la fonction suivante :

$$\forall j, 1 \leq j \leq n_s, y^j = \Phi \left( \sum_{i=1}^{n_e} w_i \cdot x^i - w_0 \right)$$

avec  $n_e$  nombre d'entrées du neurone,  $x^i$  la  $i^{\text{ième}}$  entrée avec  $1 \leq i \leq n_e$ ,  $n_s$  nombre de sorties,  $y^j$  la  $j^{\text{ième}}$  sortie avec  $1 \leq j \leq n_s$ ,  $w_i$  le poids synaptique associée à l'entrée  $i$ ,  $w_0$  le seuil d'activation,  $\Phi$  la fonction d'activation. Le seuil  $w_0$  peut être considéré comme une entrée fictive  $e_0$  de poids  $-1$ . Soit la formulation suivante :

$$\forall j, 1 \leq j \leq n_s, y^j = \Phi \left( \sum_{i=0}^{n_e} u_i \cdot x^i \right)$$

### 2.2.2. Neurones en réseau

Un réseau de neurones est défini comme une structure paramétrée complexe composée d'un ensemble de neurones connectés les uns aux autres. L'idée est de retrouver, par ce biais, des propriétés d'apprentissage et de généralisation par analogie avec le cerveau humain. Un réseau est caractérisé par son architecture et par les valeurs de ses paramètres (poids, seuil, etc.). Ces dernières valeurs ou paramètres sont estimés lors de l'apprentissage qui consiste à optimiser ces valeurs en fonction de l'utilisation que l'on souhaite faire du réseau de neurones.

### 2.2.3. Apprentissage

L'apprentissage consiste en l'estimation des paramètres d'un réseau de neurones dont l'architecture est fixée. En fonction de l'utilisation du réseau, c'est-à-dire du phénomène qu'il doit apprendre, des programmes d'optimisation permettent d'estimer les valeurs des poids synaptiques et des seuils. Le choix de l'algorithme dépend de l'architecture du réseau et des propriétés recherchées.

### 2.2.4. Application

Les réseaux de neurones permettent de résoudre, avec une grande précision, les problèmes de modélisation ou d'identification linéaires et non-linéaires. On dispose d'un système dont on connaît les entrées et les sorties quantitatives. A partir d'un certain nombre d'exemples fournis aux réseaux, ceux-ci apprennent à se comporter comme le système. Une fois cet apprentissage réalisé, on obtient des modèles ayant un nombre réduit de paramètres. L'apprentissage de ces réseaux peut ensuite être aisément renforcé sur de nouveaux exemples. Cette dernière remarque revêt une importance particulière dans le domaine de l'aéronautique où les modèles sont d'abord mis au point sur de nombreuses simulations puis recalés par ses essais en vol ponctuels mais très coûteux.

Au-delà de la modélisation et de l'identification, les réseaux de neurones ont un grand nombre d'applications potentielles [Calise et al., Kaneshige et al., Raimbault, Shen et al., Swern et al., Villaumé].

Les réseaux qui ont été utilisés ici sont de type mono-couche (une seule couche interne), à fonction d'activation sigmoïdale. Le choix de l'architecture découle du théorème d'approximation universelle qui indique que toute fonction continue sur un compact peut être approchée avec un réseau à une seule couche cachée contenant un nombre fini de neurones [Haykin]. L'utilisation de fonctions d'activation sigmoïdales a elle été dictée par l'empirisme : ce sont les fonctions les plus usitées aujourd'hui pour les applications de type identification. L'apprentissage réalisé est de type supervisé (cf. plus haut) sur un ensemble de l'ordre du millier de points.

### 2.2.5. Résultats

Le tableau ci-après contient des valeurs obtenues par double optimisation :

- optimisation de l'apprentissage retenu,
- optimisation de l'architecture

|                    |    | Réseau de neurones              |                       |                   |
|--------------------|----|---------------------------------|-----------------------|-------------------|
|                    |    | Pourcentage de points aberrants | Erreur moyenne (en %) | Ecart-type (en %) |
| A321<br>Conf 5     | Cm | 0                               | -0.0661               | 1.9775            |
|                    | Cx | 0                               | -0.0493               | 1.8551            |
|                    | Cz | 0                               | -0.0774               | 1.0159            |
| A340-300<br>Conf 5 | Cm | 0                               | -0.1194               | 2.1244            |
|                    | Cx | 0                               | 0.0074                | 0.9505            |
|                    | Cz | 0                               | -0.0429               | 1.4312            |

Tableau. 2. Performance de l'identification par réseaux de neurones des coefficients aérodynamiques de l'A340-300 et l'A321 en configuration 5 (full)

Par rapport aux polynômes, la précision obtenue est ici nettement meilleure. De plus, l'encombrement mémoire des réseaux est de 1517 octets pour le Cm et le Cz des deux avions et 933 octets pour le Cx, soit 4900 octets pour le total. Par rapport aux polynômes on gagne ainsi 20% d'encombrement mémoire.

### 2.3. COMPARATIF ENTRE LES POLYNOMES INTERPOLATEURS ET LES RESEAUX DE NEURONES

On peut dresser un bilan comparé des deux solutions étudiées précédemment :

|                          | Critère d'évaluation |                      |           |                        |            |
|--------------------------|----------------------|----------------------|-----------|------------------------|------------|
|                          | Encombrement mémoire | Rapidité d'exécution | Précision | Durée de mise au point | Généricité |
| Polynômes interpolateurs | Moyen                | Bon                  | Moyen     | Mauvais                | Mauvais    |
| Réseaux de neurones      | Bon                  | Bon                  | Bon       | Bon                    | Bon        |

Tableau. 3. Performances comparée de l'identification par polynômes interpolateurs et par réseaux de neurones

Les réseaux de neurones apparaissent ici clairement comme la meilleure solution à notre problème d'identification non-linéaire.

### 3 VERS UN LOGICIEL D'IDENTIFICATION AUTOMATIQUE

Ayant constaté la généralité de la démarche d'identification neuronale, l'idée de réaliser un logiciel d'identification automatique peut prendre forme.

#### 3.1. INTRODUCTION

Un outil d'identification non-linéaire automatique basé sur des réseaux de neurones doit être parfaitement autonome une fois lancé. Il faut donc définir a priori ou laisser le logiciel optimiser les paramètres suivants :

- Architecture
- Fonction d'activation
- Critères d'arrêt

Jusqu'à maintenant, l'intervention d'un opérateur humain était toujours requise pour ce type d'actions. Au-delà de l'automatisation, une bonne prédiction des niveaux d'erreur est nécessaire si l'on veut pouvoir développer un tel outil.

La démarche proposée est de réaliser une double optimisation et une évaluation sur trois jeux de données indépendants. Tout d'abord l'apprentissage du réseau de neurones est exécuté à partir des **données d'apprentissage**. L'apprentissage fait appel à un algorithme de rétro-propagation du gradient dont l'initialisation est aléatoire. En réalisant plusieurs apprentissages, on peut ainsi explorer plusieurs vallées d'erreur et conserver le réseau qui donne les meilleures performances pures. Le réseau retenu est ensuite évalué sur les **données de validation**.

Reste à choisir l'architecture du réseau (nombre de neurones sur la couche cachée). Pour cela, on exécute la boucle d'optimisation précédente en commençant avec un neurone sur la couche interne. On incrémente ensuite le nombre de neurones jusqu'à un critère d'arrêt. A l'issue de cette optimisation, le réseau choisi est évalué sur les **données de test**.

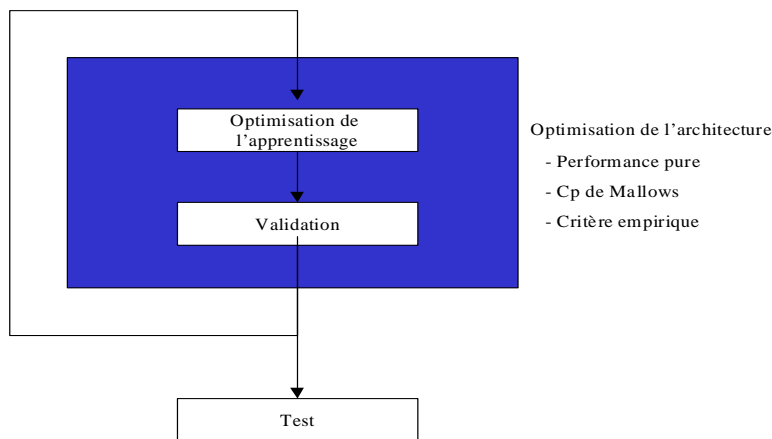


Fig.5. Principe d'un logiciel d'identification automatique basé sur les réseaux de neurones

#### 3.2. CONSTRUCTION DES FICHIERS D'APPRENTISSAGE, DE VALIDATION ET DE TEST

La définition des fichiers d'apprentissage, de validation et de test est la suivante. On récupère tout d'abord un fichier d'entrées-sorties du processus à modéliser. On va faire un tirage

aléatoire sans remise des données de ce fichier qu'on va répartir équitablement dans les fichiers d'apprentissage, de validation et de test. Ceci permet d'obtenir trois fichiers totalement indépendants et d'assurer la validité de la démarche d'identification.

Ces fichiers sont ensuite normalisés de manière à ce que chaque entrée et chaque sortie soient comprises entre -1 et 1. Cela permet d'avoir un poids apparent identique pour chaque entrée et donc un apprentissage plus rapide et précis.

### 3.3. OPTIMISATION DE L'APPRENTISSAGE

Les réseaux de neurones choisis ont une seule couche cachée. Les fonctions d'activation retenues pour cette couche cachée sont de types sigmoïdales. L'apprentissage supervisé retenu est de type Levenberg-Marquardt [Dreyfus et al.]. On réalise deux apprentissages (un pour chaque type de fonction d'activation). Les critères d'arrêt sont les suivants :

- Soit le gradient d'erreur moyenne quadratique devient inférieur à une certaine valeur.
- Soit le nombre d'itérations de la rétro-propagation devient supérieur au nombre d'arborescences existant dans le réseau de neurones à un facteur près.

Enfin, on conserve le réseau présentant la plus faible erreur quadratique moyenne sur l'échantillon de validation.

### 3.4. OPTIMISATION DE L'ARCHITECTURE

L'optimisation de l'architecture des réseaux est nécessaire pour que celle-ci soit représentative de la complexité du modèle à identifier. Pour cela une démarche originale est retenue. Il s'agit d'un algorithme de descente du gradient modifié.

Ce nouvel algorithme part du principe du développement en série de Taylor. Soient  $x$  et  $x_0$  deux réels. Toute fonction  $C^n$  de  $x$  peut s'écrire de la manière suivante au voisinage de  $x_0$  :

$$f(x) = f(x_0) + \sum_{i=1}^n (x - x_0)^i \cdot \left. \frac{d^i f}{dx^i} \right|_{x=x_0}$$

Si  $f$  est le nombre de neurones sur la couche interne correspondant au réseau d'erreur maximale  $x$ ,  $x_0$  étant l'erreur du réseau courant ( $E_c$ ) et  $f(x_0)$  l'architecture du réseau courant ( $A_c$ ), on peut ainsi utiliser cette relation pour trouver l'architecture désirée ( $A_d$ ) en fonction de l'erreur maximale désirée ( $E_d$ ) :

$$A_d = \text{round} \left( A_c + \sum_{i=1}^n (E_d - E_c)^i \cdot \left. \frac{d^i f}{dx^i} \right|_{x=x_0} \right)$$

Le calcul de la dérivée  $i$ ème de  $f$  par rapport à  $x$  nécessite d'avoir au moins  $i+1$  échantillon de  $f$ . Les méthodes actuelles sont les suivantes :

- Pour la descente du gradient on n'utilise que la dérivée première aux deux derniers points de l'optimisation.
- Pour l'algorithme de Newton, on utilise la dérivée première et la dérivée seconde (aux trois derniers points d'optimisation).

Ces principes de calcul sont relativement sensibles à la présence de minimas locaux. C'est pourquoi nous proposons un algorithme adaptatif en fonction du nombre de points d'optimisation. En effet, au fur et à mesure que l'algorithme avance, l'échantillon de points s'enrichit. A l'étape  $n+1$ , on dispose de  $n+1$  points et donc on peut utiliser la forme de Taylor à l'ordre  $n$ . Ceci permet d'affiner l'optimisation au cours du temps. A noter qu'on interdit le test multiple d'une architecture donnée, afin de gagner du temps de calcul.

Lorsque le processus mène à un réseau d'erreur inférieure à celle spécifiée, on l'arrête et on conserve ce réseau final. Il arrive toutefois que les réseaux ne puissent pas converger vers la solution, par manque de points d'apprentissage. Dans ce cas, on stoppe l'optimisation d'architecture du réseau lorsque les trois dernières architectures testées apportent une dégradation de performance. L'opérateur doit donc enrichir la base de données avant de relancer le processus s'il souhaite conserver ses exigences en terme d'erreur maximale du modèle.

### 3.5. QUALIFICATION DES MODELES

L'erreur maximale observée ne saurait être une garantie de l'erreur maximale réelle du modèle en conditions opérationnelles. C'est pourquoi on s'attache à la prédire. Cette prédiction est opérée sur la base de la loi statistique paramétrique pour laquelle le test d'adéquation de Kolmogorov-Smirnov donne la meilleure valeur. On teste ainsi les principales lois statistiques paramétriques existantes (normale, gamma, Student, lognormale, Fisher, Weibul).

La loi statistique possédant la meilleure adéquation avec l'échantillon d'erreur est retenue. Elle sert ensuite de base à la définition d'intervalles de confiance à  $10^{-3}$ ,  $10^{-5}$ ,  $10^{-7}$ , et  $10^{-9}$ , conformément aux objectifs imposés par la criticité de l'application considérée.

Il faut noter que ce module de prévision d'erreur est générique et peut lui aussi être appliqué à tout autre type de problème (dispersion des points d'impact à l'atterrissage, comportement s'un système soumis à des incertitudes, tolérances sur la taille de pièces mécaniques, etc.).

## 4 CONCLUSION

L'outil présenté dans cet article répond à de forts besoins industriels. La production automatique de modèles identifiés non-linéaires compacts, précis et fiables est un problème difficile à résoudre et l'approche proposée ici permet d'y apporter des solutions.

Ces solutions sont d'ores et déjà éprouvées puisqu'elles ont servi de base à la production de modèles pour l'estimation, la simulation et les commandes de vol. Aujourd'hui, le problème de l'embarquabilité de modèles non-linéaires semble donc résolu.

### Références

- [1] DREYFUS G., MARTINEZ J.-M., SAMUELIDES M., GORDON M. B., BADRAN F., THIRIA S., HERAULT L., Réseaux de neurones - Méthodologie et applications, *Ed. Eyrolles*, Août 2002.
- [2] HAYKIN Simon, Neural networks - A comprehensive foundation, *Prentice Hall*, 1994.
- [3] LINSE Dennis J., STENGEL Robert F., Identification of aerodynamic coefficients using computational neural networks, *Journal of guidance, control and dynamics*, Vol. 16, No. 6, November-december 1993.
- [4] MOODY John E., The effective number of parameters: an analysis of generalization and regularization in nonlinear learning systems, *Advances in neural information processing systems*, No. 4, 1992.
- [5] RAIMBAULT Nathalie, Statistiques et réseaux de neurones: applications à la modélisation et à l'optimisation du pilotage d'un avion civil, *Rapport de thèse*, 13 décembre 2001.