



## Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>  
Eprints ID : 14605

**To cite this version** : Thooris, Charles *Mise en œuvre d'une méthode de Data Mining pour appréhender le comportement d'un sujet en état de tunnélisation attentionnelle.* (2011) [Mémoire]

Any correspondance concerning this service should be sent to the repository administrator: [staff-oatao@listes-diff.inp-toulouse.fr](mailto:staff-oatao@listes-diff.inp-toulouse.fr)

Rapport intermédiaire  
Projet de Fin d'Etude

Mise en œuvre d'une méthode de Data  
Mining pour appréhender le comportement  
d'un sujet en état de tunnélisation  
attentionnelle



CAS - Centre Aéronautique et Spatial

ISAE - Institut Supérieur de l'Aéronautique et de  
l'Espace

Dirigé par Frédéric DEHAIS

Charles THOORIS

## Table des matières

Introduction.....	3
I. Présentation du laboratoire au sein de l'Institut .....	4
I.1. L'institut de l'Aéronautique et de l'Espace .....	4
I.1.1. Composition .....	4
I.1.2. Les missions.....	4
I.1.3. Partenaires .....	5
I.2. Le Centre Aéronautique et Spatial (CAS).....	5
I.2.1. Missions du CAS.....	5
I.2.2. L'équipe .....	6
I.3. Objectif du stage .....	7
II. Tunnélisation attentionnelle .....	8
III. Expérience robotique .....	10
III.1. Description de l'expérience .....	10
III.3. Résultat de l'expérience .....	12
IV. Etat de l'art des méthodes de Data Mining.....	17
IV.1. Le besoin d'une méthode de Data Mining (DM) .....	17
IV.2. Nos exigences sur les méthodes d'apprentissages .....	17
IV.3. Trois méthodes de Data Mining mises en évidence.....	18
IV.3.1. Le réseau de neurones .....	18
IV.3.2. Support Vector Machines.....	20
IV.3.3. Adaptive Network-based Fuzzy Inference systems (ANFIS).....	21
IV.3.4. Choix de la méthode d'apprentissage.....	24
V. Mise en œuvre avec ANFIS sur Matlab .....	25
V.1. Description de l'interface ANFIS sous Matlab .....	25
V.2. Modélisation de la tunnélisation attentionnelle avec ANFIS .....	27
V.3. Résultats .....	28
V.3.1. Réglage de l'outil pour un bon apprentissage.....	28
V.3.2. Benchmark de deux méthodes : le Neuro-Fuzzy avec ANFIS sur Matlab et le SVM de libSVM sur Matlab .....	32
V.3.3. Résultats obtenus grâce à la méthode neuro-floue.....	34
Conclusion et travaux futurs .....	38
Bibliographie.....	39

## Introduction

Dans l'aéronautique, on considère que 80% des accidents sont dus à une erreur humaine dans l'aviation civil et militaire (O'Hare, Wiggins, Batt, & Morrison, 1994) (Wiegmann & Shappell, 2003). Ces statistiques ont donc amené nombre de scientifiques à s'intéresser au sujet des facteurs humains. L'idée est d'améliorer la sécurité aérienne en comprenant mieux le comportement humain.

On s'aperçoit en effet que certains accidents rejoués en simulateur par d'autres pilotes expérimentés conduisent parfois au même crash (Wanner & Wanner, 1999). C'est parfois l'environnement qui conduit à l'erreur humaine. Ainsi il est intéressant de rechercher des moyens d'aider l'opérateur dans sa tâche. Ce n'est pas chose si aisée. Van Eslande et al (Van Eslande, Erreur de conduite et besoin d'aide : une approche accidentologique, 2001) (Van Eslande, Alberton, Nachtergaële, & Blancher, 1997) postulent que le comportement des automobilistes est essentiellement conditionné par les infrastructures routières.

Il a été remarqué que les conflits étaient un précurseur remarquable d'erreurs humaines conduisant à l'accident. Des conflits entre l'humain et la machine, ou entre l'opérateur et la tour de contrôle, ou encore entre le pilote et le co-pilote. L'étude des conflits s'avère alors un thème pertinent pour les facteurs humains. C'est dans ce domaine que nous travaillons au CAS au sein de l'ISAE.

Le rapport sera constitué de trois grandes sections. Dans un premier temps nous présenterons l'environnement de travail à l'ISAE. Puis dans les parties suivantes nous définirons de manière plus précise ce qu'est la « tunnélisation attentionnelle » et décrirons plus en profondeur l'expérience du robot qui est notre base de travail. Enfin nous présenterons les résultats en termes de diagnostic de l'état d'un opérateur.

## I. Présentation du laboratoire au sein de l'Institut

### I.1. L'institut de l'Aéronautique et de l'Espace

#### I.1.1. Composition

L'institut Supérieur de l'Aéronautique et de l'Espace est issu du rapprochement de SUPAERO, créée en 1909 et de l'ENSICA créée en 1945. SUPAERO est tout d'abord implantée à Paris puis déménage à Toulouse, à Rangueil. L'ENSICA par ailleurs, est implantée à Jolimont. Suite au rapprochement, les deux formations seront amenées à cohabiter sur le campus SUPAERO, mais resteront néanmoins deux formations séparées.

#### I.1.2. Les missions

L'ISAE assure plusieurs missions, que l'on peut séparer selon les secteurs suivants :

##### Formation

La formation est très complète mais se cantonne à l'enseignement supérieur :

- 2 formations ingénieurs (SUPAERO et ENSICA)
- 2 diplômes d'études supérieures : DESIA et DESII
- 2 Diplômes Nationaux Mastères
- 19 Mastères Spécialisés
- 5 Mastères Recherche
- 6 écoles doctorales

##### Recherche

La recherche est un levier important de la formation à l'ISAE, qui s'organise autour de 6 départements :

- Aérodynamique, énergétique et propulsion (DAEP)
- Mécanique des structures et matériaux (DMSM)
- Electronique, optronique et signal (DEOS)
- Mathématiques, informatique, automatique (DMIA)
- Langue, arts, cultures et sociétés (LACS)
- Centre Aéronautique et Spatial (CAS)

##### Formation Continue

Trois structures au sein de l'ISAE assurent la formation continue pour les personnes déjà en activité dans le monde du travail qui complètent leur formation :

- EUROSAT

- ECATA : European Consortium for Advanced Training in Aerospace
- ITSA : International Training center for Space Applications

### I.1.3. Partenaires

#### Industriels

L'ISAE entretient des relations très étroites avec les entreprises, au niveau national, européen et mondial. Le partenariat se concrétise par l'intervention de nombreux professionnels au sein des formations, par de nombreuses offres de stage et autres projets en entreprise (brillamment relayés par le service stage et carrière que l'on ne peut pas assez remercier), par des études et recherche hébergées à l'ISAE au service des entreprises (Thèses CIFRE, SUPAERO Junior Conseil,...), par la formation continue qui regroupent nouveaux ingénieurs et cadres expérimentés dans les mêmes salles de classes, et par de nombreuses rencontres entre étudiant et entreprises lors de conférences et autres déjeuners débats toujours orchestrés par le service stage et carrière. Par ailleurs, la présence de ces industriels au conseil d'administration et au conseil de la formation et au conseil de la recherche garantissent une bonne adéquation entre la formation et les besoins des entreprises.

#### Académiques

L'ISAE est membre de 3 réseaux d'universités, pour le Cours Supaero :

- Le GEA : Groupe des Grandes Ecoles Aéronautiques et Spatiales de France
- PEGASUS : Partnership of a European Group of Aeronautics and Space Universities
- TIME : Top Industrial Managers for Europe
- ECATA : European Consortium for Advanced Training in Aerospace
- Amicale ISAE-SUPAERO-ENSICA

## I.2. Le Centre Aéronautique et Spatial (CAS)

### I.2.1. Missions du CAS

L'ISAE développe depuis 2004 une activité dans le domaine des interactions hommes-systèmes. Cette activité de recherche, menée par l'équipe Facteurs Humains du CAS est transdisciplinaire : neurosciences, neuropsychologie, ergonomie et intelligence artificielle. Elle vise à élaborer des modèles formels des mécanismes cognitifs pour concevoir de nouveaux systèmes interactifs adaptés à l'homme. Une caractéristique importante de cette activité est de s'appuyer sur une approche expérimentale forte : de nombreux moyens d'expérimentation sont conçus et mis en œuvre (simulateur de vol, avions réels, plateforme robotique..). Des techniques de mesures récentes sont utilisées (ex : imagerie médicale) ou mises au point (ex : capteurs physiologiques et comportementaux) grâce à des collaborations de recherche. Cette activité trouve de multiples champs applicatifs à travers l'assistance au pilotage (aéronef, automobile), aux interactions hommes robots ou encore à l'aide aux patients.

Des expérimentations récentes menées au CAS ont montré que l'apparition d'un fort stress pouvait générer des comportements de « tunnélisation attentionnelle » où les opérateurs se focalisent excessivement sur la réalisation d'une tâche au détriment des autres, et ne sont plus sensibles aux alarmes visuelles et auditives. Dès lors, il devient important de définir des moyens pour détecter en temps réel ce type de comportement, à l'origine de nombreux accidents en aéronautique. De premiers travaux ont déjà été réalisés en collaboration avec Sergio Pizziol, doctorant à l'ONERA, sous la direction de Catherine TESSIER. Il a notamment défini un premier modèle de la tunnélisation attentionnelle par des règles floues. Nicolas Régis recherche de nouvelles métriques lors de son PFE pour obtenir un diagnostic plus robuste. Une deuxième partie est de créer un système combinant les différentes métriques. Cette dernière partie représente le travail de mon stage.

Ce travail sur l'état d'un opérateur s'inscrit dans un contexte plus large : le projet GHOST. Ce projet est mené en lien avec Airbus.

## I.2.2. L'équipe

Le CAS est composé d'une dizaine de personnes. La figure ci-dessous présente un organigramme expliquant le lien entre les personnes. Frédéric Dehais nous encadre : Nicolas Régis, Eric Menant et moi-même. Nous étions tous les trois dans l'équipe des facteurs humains au sein du Centre Aéronautique et Spatial. Nous avons chacun la responsabilité d'une tâche spécifique mais comme l'équipe est petite et que nous sommes issus de la même formation nous nous aidions mutuellement dans nos tâches respectives. Frédéric Dehais et Mickael Causse nous apportent une expertise en ce qui concerne la neuroscience, la psychosociologie de la prise de décision, la gestion des conflits et la distribution de l'attention en fonction de la tâche et de marqueurs physiologiques associés. Dans cette configuration Eric Menant s'occupait plus particulièrement de la tenue de l'expérience que nous réalisions sur le simulateur. Nicolas Régis cherchait des marqueurs physiologiques et comportementaux de la « tunnélisation attentionnelle ». Enfin mon stage portait sur la dernière partie du travail : créer à partir des données de Nicolas Régis un système capable de diagnostiquer l'état de « tunnélisation attentionnelle » d'une personne.

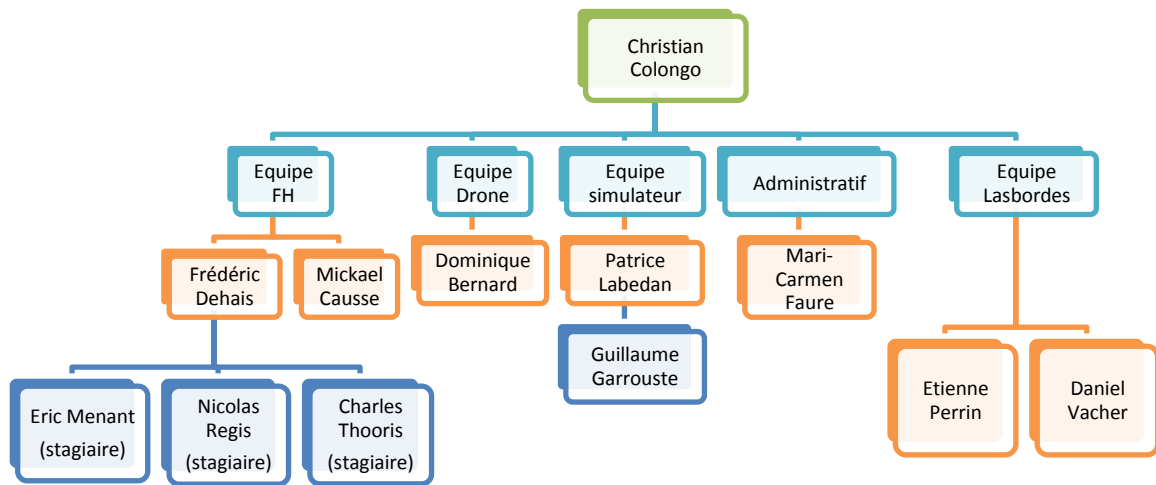


Figure 1 : Organigramme de l'équipe du CAS

Au sein du laboratoire nous utilisons le simulateur de vol dans le besoin d'une expérience visant à déclencher la « tunnélisation attentionnelle » sur certains participants. Guillaume Garrouste et Patrice Labedan s'occupe en partie d'assurer le développement et la mise à jour du simulateur. Daniel Vacher et Etienne Perrin sont notamment en relation avec Lasbordes pour tous ce qui est essais en vol. Marie-Carmen Faure s'occupait de la partie administrative du département. Enfin, Christian Colongo dirige le département.

### I.3. Objectif du stage

L'objectif de mon stage dans ce contexte est de trouver une méthode formelle pour diagnostiquer la « tunnélisation attentionnelle » à partir de données physiologiques et oculométriques. Ainsi L'objectif de mon travail consiste à étudier différentes techniques d'apprentissage et de choisir celles qui conviennent le mieux à la problématique de recherche. Ensuite, l'objectif est d'appliquer certaines de ces méthodes sur des données enregistrées lors d'une expérimentation sur la plateforme robotique avec une vingtaine de sujets.



## II. Tunnélisation attentionnelle

La tunnélisation attentionnelle est définie comme « *l'allocation de l'attention sur une certaine information, sur une hypothèse particulière dans un diagnostic, sur une tâche dans un certain but, ceci pendant une durée qui dépasse l'optimal, ce qui entraîne la négligence des autres informations, hypothèses où l'échec d'autres tâches à effectuer.* » (Wickens C. , 2005). Des expérimentations menées au Centre Aéronautique et Spatial à l'ISAE (Dehais, Tessier, & Chaudron, Ghost : experimenting countermeasures for conflicts in the pilot's activity, 2003) (Dehais, Method and Device for Detecting absences of manual and automatic piloting of an aircraft, 2009) (Dehais, Causse, Vachon, & Tremblay, 2011) montrent que la tunnélisation attentionnelle peut conduire des opérateurs humains à ne plus être sensibles aux alarmes. Ce type de comportement pourrait être à l'origine de nombreuses catastrophes aériennes où les pilotes n'auraient pas réagi à des alarmes.

Dès lors la tunnélisation attentionnelle présente un paradoxe aux concepteurs d'interfaces : comment peut-on espérer sortir les opérateurs de leur cécité attentionnelle si les alarmes / systèmes conçus pour les prévenir sont négligés ? Une solution consiste à adopter une approche "neuroergonomique" (Parasuraman & Rizzo, Neuroergonomics : the brain at work, 2007) pour concevoir des contre-mesures cognitives en proposant des concepts d'interface neuropsychologiquement compatibles (Prévic, 2000). Posner et al (Posner & Dehaene, Attentional networks. Trends in neurosciences, 1994) postulent que les processus attentionnels sélectifs sont réalisés par des réseaux attentionnels dits d'alerte, d'orientation, et de contrôle exécutif. Des recherches menées sur des patients traumatisés crâniens suppose que la déficience du réseau d'orientation peut provoquer des négligences visuelles (Posner, Walker, Friedrich, & Rafal, 1984). De plus, (Pecher, Quaireau, Lemercier, & Cellier, 2010) ont montré que l'émotion affecterait sélectivement la fonction d'orientation chez des sujets sains. En particulier, ce réseau d'orientation repose sur un mécanisme procédant en trois temps et qui consiste à désengager, déplacer, et réengager l'attention (Posner & Dehaene, Attentional networks. Trends in neurosciences, 1994) (un schéma représente le modèle de Posner ci-dessous). L'expérimentation menée dans un simulateur de vol (Dehais, Tessier, & Chaudron, Ghost : experimenting countermeasures for conflicts in the pilot's activity, 2003) a suggéré que l'absence de réaction à des alarmes sonores ou visuelles peut être expliquée par une incapacité à dégager l'attention, une hypothèse cohérente avec les travaux de certains auteurs (Philipot & Brutoux, 2008) (Donaldson, Lam, & Mathews, 2007) (Hills, 1980). Or, les alarmes sonores et visuelles reposent sur l'ajout d'information pour attirer l'attention, mais ne sont pas suffisamment efficaces pour la désengager. Au contraire, le principe des contre-mesures cognitives repose sur une suppression temporaire des informations sur lesquelles se concentrent l'opérateur humain et leur substitution par un stimulus visuel explicite pour changer le focus attentionnel. Dans cette optique, l'interface utilisateur viendrait agir comme une prothèse cognitive qui effectue le désengagement attentionnel.

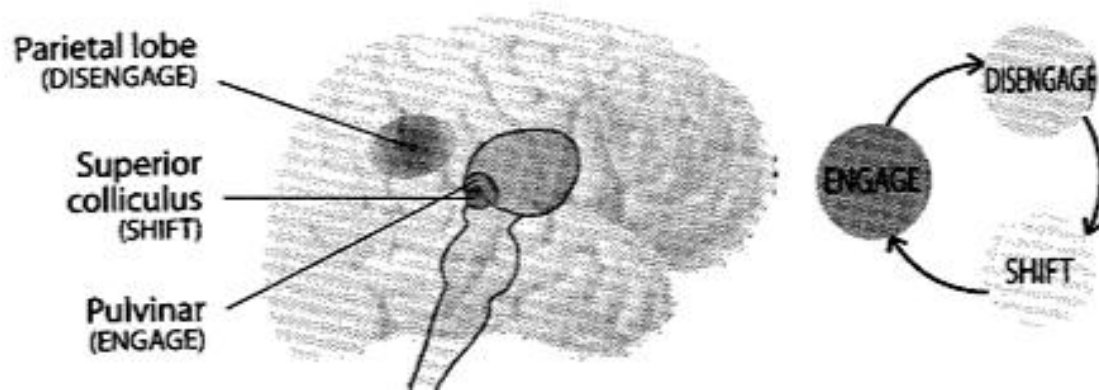


Figure 2 : Modèle attentionnel de Posner, 1984

Avec ces informations, une expérience a été mise en place à l'ISAE qui met en évidence la tunnélisation attentionnelle. Elle est décrite dans la partie suivante.

### III. Expérience robotique

Afin d'évaluer précisément l'impact du conflit sur le comportement cognitif d'un opérateur humain, l'équipe des facteurs humains a développé un environnement robotique constitué d'un robot terrestre autonome, d'une station sol de pilotage et de supervision ainsi que d'une interface de magicien d'Oz pour déclencher des aléas (voir figure ci-dessous).



Figure 3 : L'image de gauche montre le robot développé par l'ISAE. L'image de droite représente la station sol de supervision et de pilotage du robot. Les parties critiques de l'interface sont numérotées : (1) écran de la vidéo panoramique, (2) synoptique, (3) carte Google tactique, (4) panneau de dialogue, (5) états du GPS, des ultrasons et de la batterie, (6) mode de pilotage du robot.)

#### III.1. Description de l'expérience

Une première expérimentation a été réalisée dans cet environnement où un scénario opérationnel consistait en une mission d'identification d'une cible (Dehais, Causse, Vachon, & Tremblay, 2011). Dans cette mission (figure ci-dessous), le robot rejoint en mode supervisé une zone de recherche, dans laquelle il effectue un balayage. Lorsque le robot a détecté la présence de la cible, il rend la main à l'opérateur humain pour que celui-ci l'identifie. Afin de provoquer un conflit d'autorité, une chute de tension de la batterie du robot se produit (aléa déclenché par le magicien d'Oz) au moment où l'opérateur prend en main le robot pour identifier la cible. Cet aléa conduit le robot à une prise d'autorité qui interrompt la mission pour retourner à la base en mode supervisé. Cet événement est perceptible pour l'opérateur sur son interface à l'aide de plusieurs alertes visuelles mais comme cet aléa se produit à un moment crucial de la mission où l'opérateur est particulièrement focalisé sur sa tâche d'identification en mode pilotage manuel, il est attendu qu'il ne perçoive pas ces changements d'état et que chaque agent – opérateur et robot - s'entête à poursuivre son but (retrouver la cible vs. retour base). Ainsi, la mission se déroule en 4 segments : S1 « aller sur zone » en mode autonome, S2 « rechercher cible » en mode autonome, S3 « identifier cible » en mode manuel et S4 « conflit ».

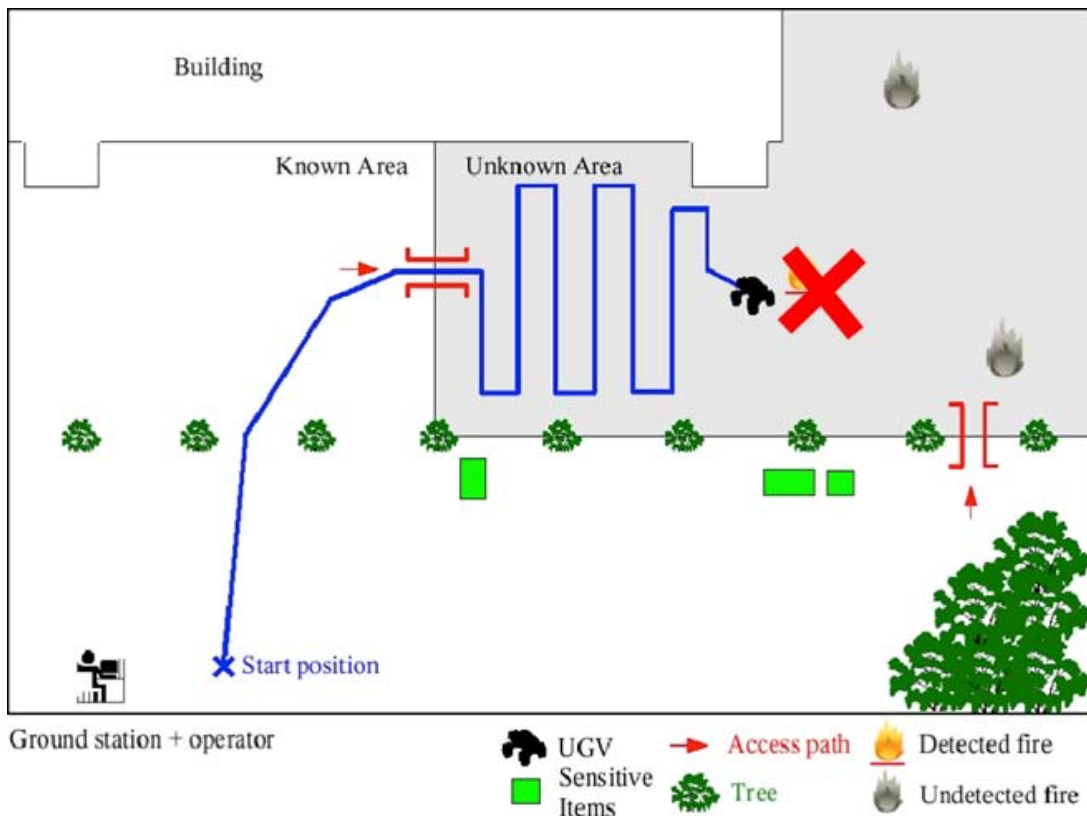


Figure 4 : La mission opérationnelle : les opérateurs sont placés dans une salle fermée sans contact visuel avec le robot qui évolue sur le terrain. Lorsque le robot détecte une cible, l'opérateur prend le contrôle pour l'identifier mais à ce moment-là le magicien d'Oz déclenche une panne de batterie. Un conflit se produit alors entre l'opérateur (rester "sur zone" pour continuer l'identification) et le robot (rentrer le plus rapidement à la base)

### III.2. Participants

La population des participants en deux groupes. 11 volontaires sur lesquels nous avons envoyé une contre-mesure lors de l'apparition de la panne et 12 volontaires sur lesquels nous n'avons pas donné de contre-mesure particulière. La panne se caractérisait alors par les indicateurs habituels (voir figure 5). La contre-mesure se caractérisait par le retrait de l'information que le candidat regarde. Ici ce sera l'écran vidéo (voir figure 6 pour avoir la contre mesure que nous avons proposé aux candidats). Ce retrait va permettre au participant de récupérer cette faculté de désengagement (dont nous avons parlé dans la partie précédente). Nous étudions cette persévérance à travers des capteurs que nous avons mis sur le candidat. Sur cette expérience il est équipé d'un eye tracker qui nous permet de suivre son regard. Il est aussi équipé d'électrodes nous donnant son électrocardiogramme.

9 participants sur 12 n'ont pas détecté la panne. Ils ont persévéré et essayé à tout prix de finir leur mission : lire le message sur la cible. Sur les 11 participants pour qui la contre-mesure a été délivrée, 10 volontaires ont suivi la procédure et ont donc laissé le robot rentrer seul à la base en mode supervisé.

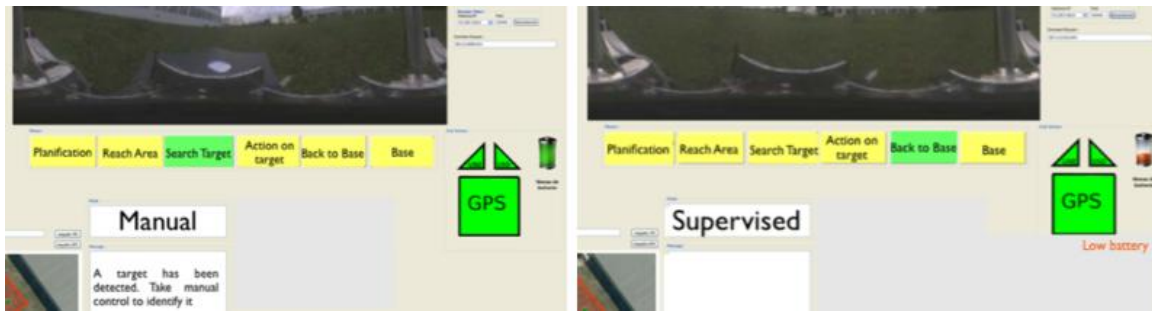


Figure 5 : l'écran de droite montre l'interface avant la panne alors que l'écran de gauche montre l'interface avec la panne de batterie



Figure 6 : Ces quatre images représentent respectivement les quatre étapes chronologiques de la contre-mesure cognitive. Image supérieure gauche - étape 1 : la vidéo panoramique est supprimée pendant 1 seconde. Image supérieure droite - étape 2 : les informations pertinentes pour comprendre le comportement du robot sont placées dans le champ visuel pendant 3 secondes. Image inférieure gauche - étape 3 : la vidéo panoramique réapparaît avec les informations pertinentes en surimpression pendant 3 secondes de plus. Image inférieure droite - étape 4 : fin de la contre-mesure cognitive.

### III.3. Résultat de l'expérience

Sur cette dernière phase nous avons pu constater des différences au niveau physiologique et oculométrique entre les personnes ayant tunnelisé et persévéré et les personnes ayant laissé le robot rentrer seul à la base. C'est-à-dire entre les personnes ayant reçu une contre mesure et les autres. Les trois première phases ne sont pas discernables ce qui nous montre que la différence se fait uniquement grâce à la contre-mesure.

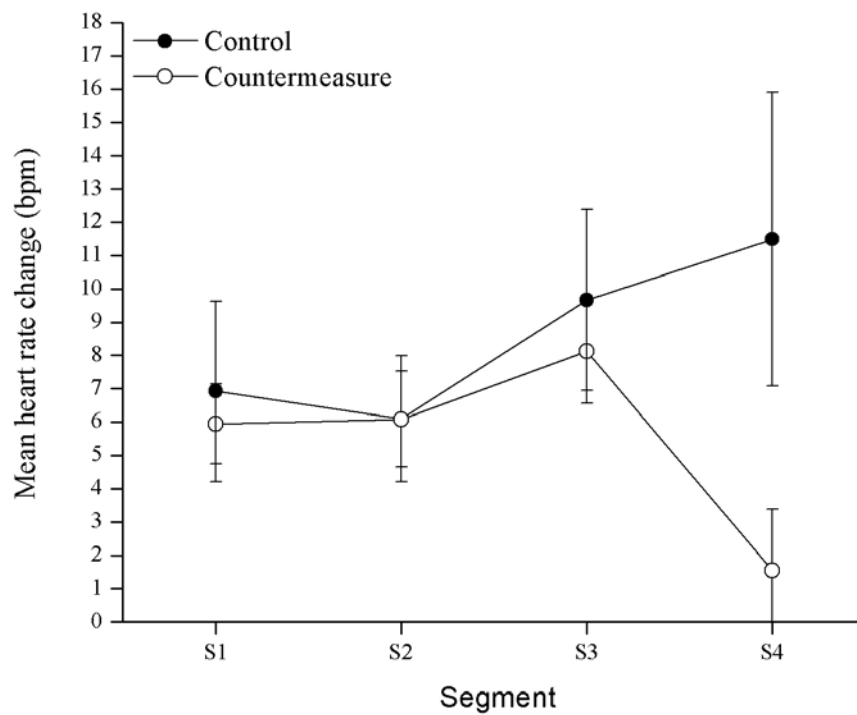


Figure 7 : Les candidats ayant reçu une contre-mesure ont un rythme cardiaque plus faible que les autres

Prenant l'exemple du rythme cardiaque (figure ci-dessus), nous avons constaté qu'il se crée un différence sur la phase 4 entre les personnes ayant reçu une contre mesure et les autres

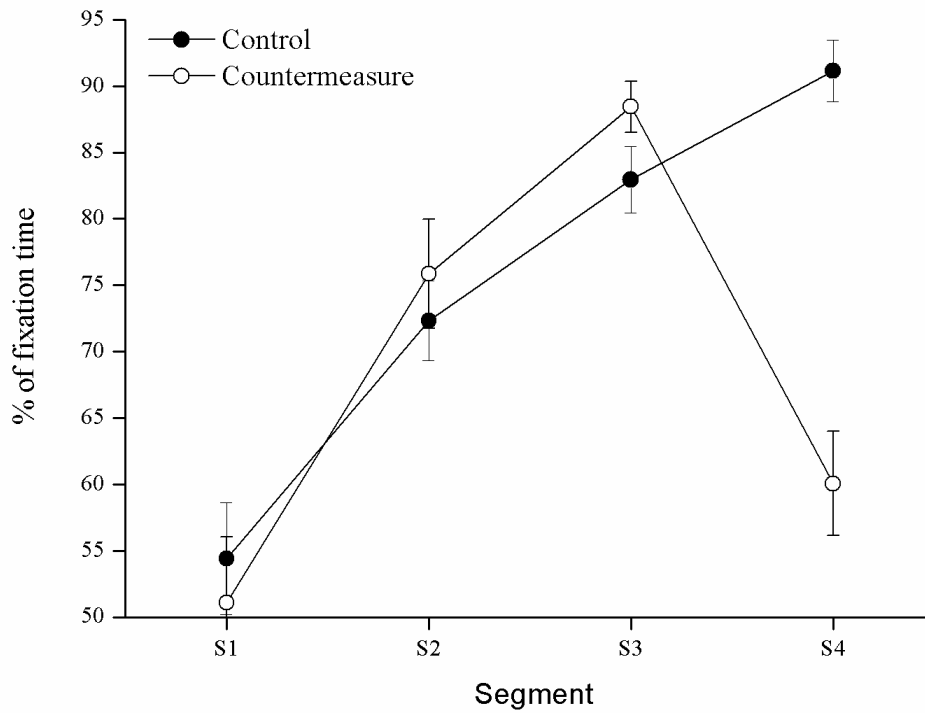


Figure 8 : Les candidats ayant reçu une contre-mesure font moins de fixation que les autres

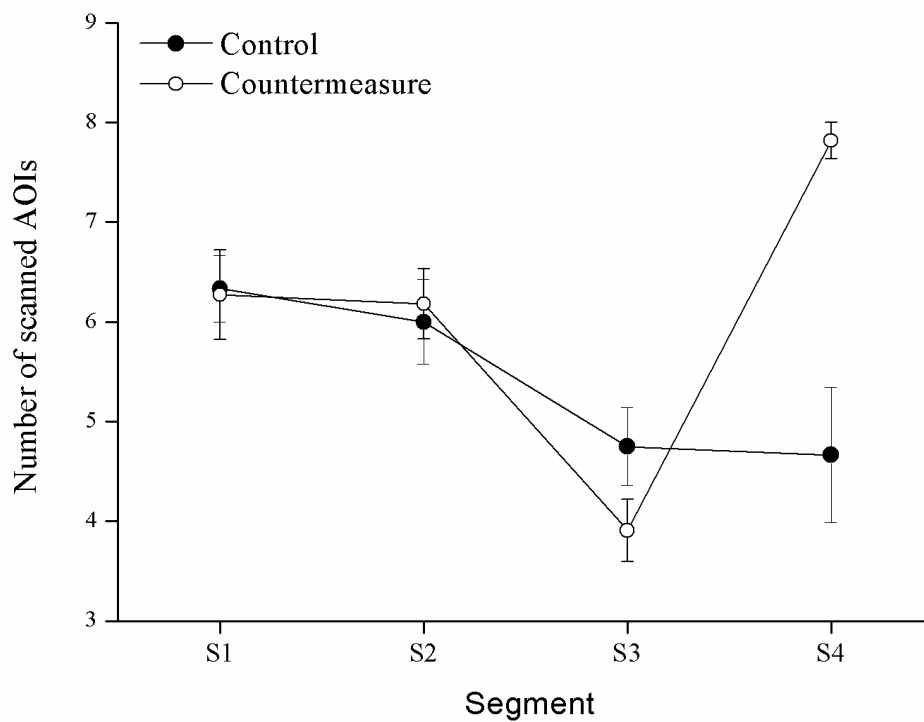


Figure 9 : Les candidats ayant reçu une contre-mesure scannent plus de zones que les autres

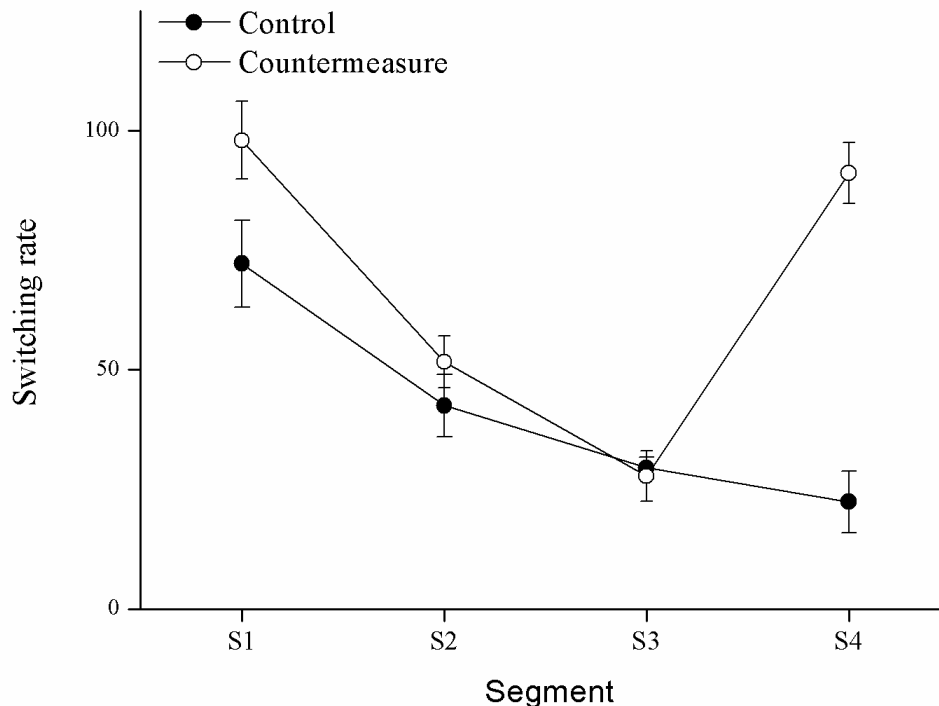


Figure 10 : Les personnes ayant reçu une contre-mesure ont un taux de transition oculaire plus élevé que les autres

### III.4. Perspectives à cette expérimentation : vers un système de diagnostic temps réel

Une perspective de recherche intéressante est de disposer d'un système qui pourrait diagnostiquer l'apparition de la tunnélisation attentionnelle à partir de ces données. En particulier, ce système pourrait envoyer automatiquement en temps réel des contre-mesures. Nous espérons arriver au système présenté sur la figure ci-dessous. La figure présente un système basé sur la logique floue développé par Sergio Pizziol. Dans une démarche formelle proche de celle de Mandryk (Mandryk & Atkins, A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies, 2007), il a proposé l'utilisation de la logique floue pour agréger des mesures comportementales et psycho-physiologiques en vue d'estimer le degré d'engagement attentionnel d'un opérateur humain. L'usage de cette logique se justifie dans la mesure où il n'existe pas de modèle mathématique pour décrire ce phénomène attentionnel, où les variables d'entrée sont en partie continues (ex : rythme cardiaque) et où il est possible d'établir une relation intuitive, en langage naturel, entre les variables d'entrée et de sortie [Cox, 1992]. Une première formalisation de la "tunnélisation attentionnelle" (Pizziol, Dehais, & Tessier, 2011) est réalisée à partir des données oculométriques et cardiaques enregistrées lors des expérimentations présentées en chapitre premier. L'identification de la "tunnélisation attentionnelle" est établie à partir des indicateurs connus dans la littérature. Les métriques utilisées sont résumées plus bas. Ce seront les mêmes métriques que nous utiliserons pour construire notre futur système.



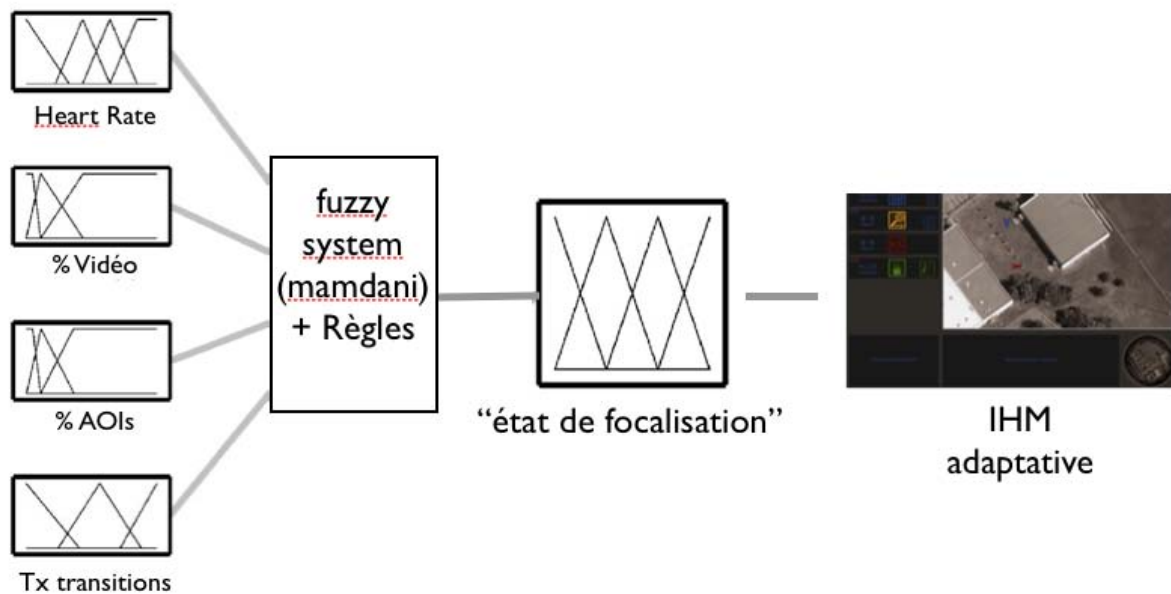


Figure 11 : Projet de système qui évalue en temps réel l'état de focalisation. Un premier système a été réalisé par Sergio Pizzoli en utilisant la logique floue

Il s'agit de créer un système d'inférence pour diagnostiquer la tunnélisation attentionnelle à partir des données de l'expérience robotique. Des travaux ont montré le bien fondé de telles techniques d'inférence (Parasuraman, Mouloua, & Hilburn, Adaptive aiding and adaptive task allocation enhance human-machine interaction, 1999) pour déterminer l'état des opérateurs en intégrant différentes techniques de mesures psychophysiques et comportementales (John, Kobus, Morrison, & Schmorrow, 2004). Ainsi, une étude dans le domaine de l'aéronautique a montré qu'il était possible de déterminer le niveau de charge de travail via des mesures cardiaques et de la résistance électrodermale (Haarmann, Boucsein, & Schaefer, 2009) et des expérimentations conduites sur un véhicule de l'armée américaine ont permis de dériver la charge cognitive en temps réel à travers l'analyse de données encéphalographiques (Dixon, et al., 2009). Par ailleurs (Marshall, 2007) a déterminé un nombre de métriques oculaires (clignement, dilatation pupillaire, vergence...) pour identifier le niveau d'engagement attentionnel dans de nombreuses tâches telles que la conduite automobile. Différents formalismes sont proposés pour diagnostiquer l'état des opérateurs en combinant des systèmes de règles et de classification (Rani, Sarkar, Smith, & Kirby, 2004) (Rani, Sims, Brackin, & Sarkar, 2002) tels que les machines à vecteurs de support (Kapoor & Picard, 2005), la logique floue (Mandryk & Atkins, A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies, 2007), les stateflows (Liu, Conn, Sarkar, & Stone, 2008) ou les approches stochastiques comme les chaînes de Markov cachées (Kulic, Carter, & Brown, 1992). Ainsi l'efficacité de ces systèmes inférentiels dépend tant du choix des capteurs physiologiques pour mesurer des états cognitifs ou émotionnels (Causse, Pavard, Senard, Demonet, J.-F., & Pastor, 2007) (Dehais, Sisbot, Alami, & Causse, 2011) que de l'utilisation d'un formalisme adapté pour rendre compte de leur évolution en temps réel. C'est de ce dernier sujet dont nous allons discuter dans la partie suivante.

## IV. Etat de l'art des méthodes de Data Mining

### IV.1. Le besoin d'une méthode de Data Mining (DM)

Nous avons cherché à produire un système permettant de diagnostiquer la tunnélisation attentionnelle à partir de données physiologiques et oculométriques. Il résidait plusieurs problèmes quant à la réalisation de ce système. Le premier problème était l'important volume de données à traiter. Nous devons traiter près de 90 000 données (23 personnes avec plus de 1000 données par personne, 4 entrées et une sortie). D'autre part nous n'avons, à ce jour, pas de connaissance experte concernant la tunnélisation attentionnelle. De plus nous avons des données hétérogènes qui proviennent de la physiologie comme de l'eye tracker. Les personnes traitant ces informations seront sujettes à leurs aprioris.. Notre approche doit répondre au mieux à ces besoins. Dans ce sens nous proposons d'utiliser des méthodes de fouille de l'information appelées méthodes de Data Mining (DM) ou Machine Learning.

Dans le développement de l'intelligence artificielle, les méthodes de DM sont des techniques d'apprentissages autonomes qui permettent à partir d'expériences multiples d'en déduire des règles générales. Elles se sont développées à partir de 1989 aux conférences de l'IJCAI (Nikhil & Lakhmi, 2005). Ces méthodes sont particulièrement utiles lorsque le nombre d'informations est trop élevé pour être traité par une personne. Elles sont basées sur la statistique et l'informatique pour pouvoir faire émerger des tendances à partir de masses de données. Ces méthodes ont déjà été éprouvées notamment avec des entrées physiologiques dans la détection du stress (Barreto, Zhai, & Adjouadi, 2007) où de la fatigue (Rigas, Goletsis, Bougia, & Fotiadis, 2011), elles donnent des taux de réponses très corrects avec des erreurs quadratiques moyennes descendant en dessous de 20%. C'est pour cette raison qu'il nous a semblait juste de l'utiliser dans notre cas.

Outre le volume de données, la recherche par méthode de DM nous permet de rester ouverts sur toutes les possibilités de combinaisons sans avoir d'hypothèses préalables. En effet nous n'avons pas de connaissances experte sur le sujet, cette neutralité est donc importante. Le système va nous la garantir en fonctionnant sans a priori et en se basant sur la statistique pour fonctionner.

Enfin les méthodes de DM devraient fournir un poids optimal pour chaque entrée de façon automatique. Ce sera donc un moyen rapide et plus sûr qu'un opérateur humain de vérifier qu'une entrée nous permet d'affiner notre diagnostic.

### IV.2. Nos exigences sur les méthodes d'apprentissages

Les méthodes de DM sont très nombreuses. Nous ne pouvons pas les utiliser toutes pour les comparer. Le meilleur moyen est de définir nos exigences concernant ces méthodes et de faire un état de l'art pour savoir laquelle nous conviendra le mieux. En fonction des critères que nous avons sélectionnés nous espérons qu'une de ces méthodes va ressortir. Nous nous basons sur 3 critères principaux.

Le premier des critères est le taux d'erreur de notre système. Nous voulons en premier lieu que notre modèle réponde le mieux possible au problème que nous lui posons. Dans notre cas nous voulons que le diagnostic soit le plus proche de la réalité en fonction des données que nous lui donnons. Si nous diagnostiquons des personnes comme tunnélisées alors qu'elles ne le sont pas nous risquons d'envoyer des contre-mesures qui n'auraient pas lieu d'être et qui risque simplement de perturber le pilote. Ce premier critère est fondamental. A lui seul il pourra exclure une méthode par rapport à une autre.

Ensuite nous cherchions un système répondant en temps réel car notre objectif à plus long terme est d'avoir une interface adaptative fonction de l'état de l'opérateur. Un temps d'entraînement long (qui sert à régler notre système) ne nous dérange pas particulièrement. Par contre nous souhaitons qu'il y ait un temps le plus court possible entre le moment où le pilote est tunnélisé et celui à partir duquel nous pouvons diagnostiquer la tunnélisation attentionnelle.

Enfin nous désirons un système capable de nous trouver différents niveaux de tunnélisation attentionnelle. Un système développant une sortie continue serait l'idéal. Nous imaginons qu'il existe plusieurs niveaux de tunnélisation attentionnelle, des montées spécifiques de la tunnélisation attentionnelle, et nous voulons pouvoir les diagnostiquer.

Comme nous le disions précédemment il existe beaucoup de méthodes capables de fouiller des données pour en faire ressortir des tendances. Nous nous sommes intéressés à trois méthodes en particulier : la méthode des Support Vector Machines (SVM), les réseaux de neurones et la méthode neuro-floue. Nous allons en faire un état de l'art dans la partie suivante pour savoir d'après la littérature celle qui répondrait le mieux à nos exigences.

### **IV.3. Trois méthodes de Data Mining mises en évidence**

Cette liste n'est pas exhaustive mais rassemble des méthodes qui fonctionnent bien dans le cas d'un système utilisant des données physiologiques et eye tracker en entrée. Nous n'avons pas encore tout étudié mais nous avons choisi des méthodes que nous retrouvons souvent dans la littérature (SVM et réseau de neurones) et une dernière méthode plus récente rassemblant les réseaux de neurones à la logique floue. Cette dernière nous a paru intéressante quant à notre sujet même si les références manquent encore à son sujet. Le but est bien de trouver une méthode qui marche plutôt que de recenser toutes les méthodes existantes.

#### **IV.3.1. Le réseau de neurones**

Cette méthode met en place un système de connections neurones/synapses qui permet un apprentissage par la modulation du poids des différentes connections. Il s'inspire des connections que nous retrouvons dans le cerveau humain. Nous avons un schéma d'un neurone artificiel ci-dessous. Il prend un certain nombre et leur applique à chacun un certain poids pour ensuite donner une sortie. Dans un réseau les poids des différents neurones vont aussi être modulés pour donner une grande structure. Cette dernière va combiner les neurones selon certaines formes.

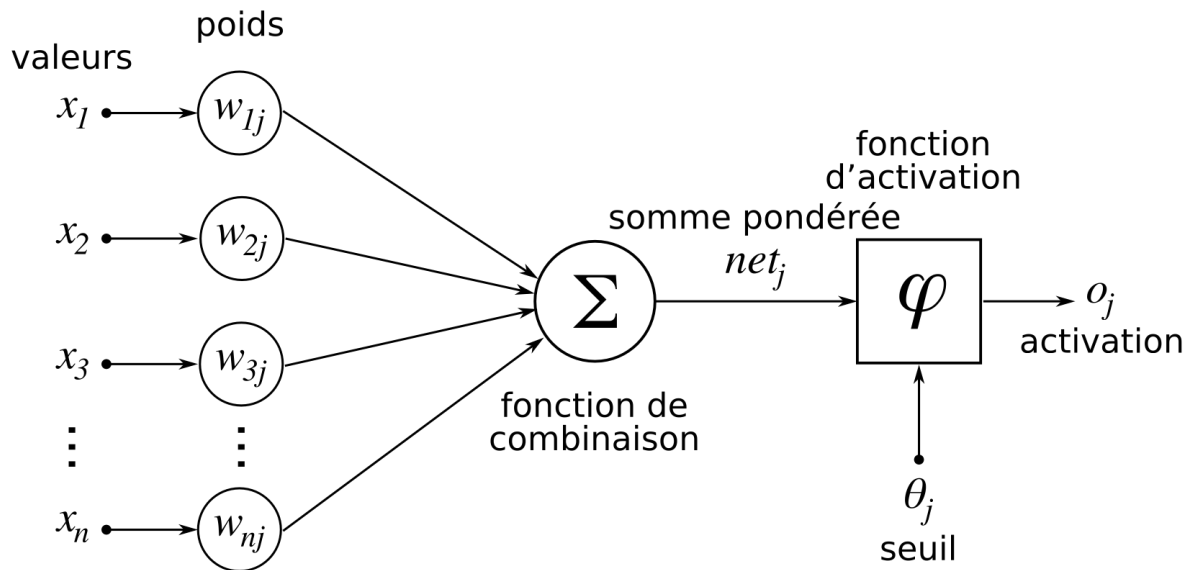


Figure 12 : structure d'un neurone artificiel

La figure ci-dessous donne un exemple de forme de réseau de neurones. Celle-ci combine 4 couches de neurones. Dans ce cas précis une couche prend ses entrées de la couche précédente et donne ses sorties à la couche suivante.

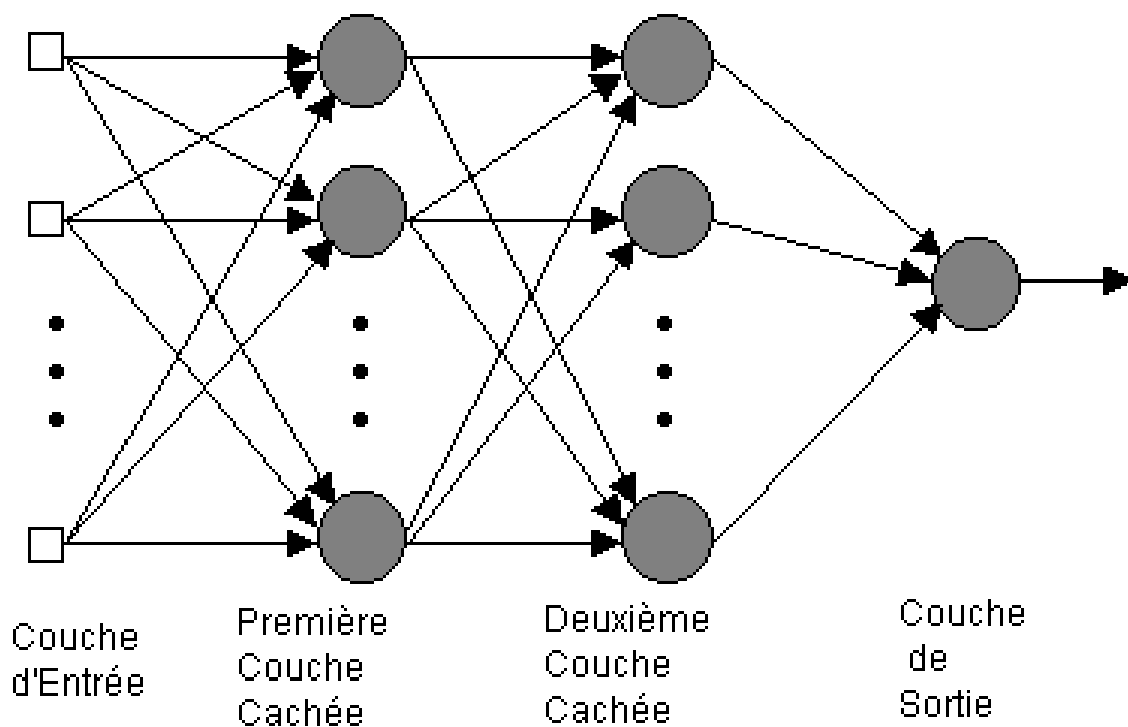


Figure 13 : réseau de neurones multicouche (Zenou)

Le réseau va ensuite apprendre à partir d'exemples que nous allons lui donner. Il va fonctionner selon un algorithme à partir duquel il va minimiser l'erreur qu'il fait par rapport à la

sortie que nous lui imposons. Un des algorithmes les plus connus est celui de la rétro-propagation du gradient. Elle consiste à regarder la différence entre la valeur cible attendue par le réseau et la valeur donnée par le réseau après l'apprentissage. Ensuite le réseau va modifier le poids des neurones en fonction du gradient de cette différence de façon à toujours diminuer l'erreur. Les neurones participant le plus à l'erreur vont recevoir un poids plus faible et les autres un poids plus fort.

### Avantages et inconvénients de la méthode :

- L'apprentissage par réseau de neurone est lent pour l'entraînement mais donne un taux de réponse en temps réel très rapide.
- Le taux d'erreur est généralement plus fort que la méthode des machines à vecteurs de support (SVM). Malgré tous des comparaisons ont été faites et ont déjà prouvé que parfois la méthode des réseaux de neurones avait un taux d'erreur inférieur au SVM. L'exemple dans le papier cité ici compare les deux méthodes dans le cadre de la distraction au volant (Tango F. , Botta, Minin, & Montanari, 2010).
- Le lien entre les entrées et les sorties est une boîte noire dans laquelle il n'est pas possible d'avoir une connaissance sur notre système. C'est le plus gros inconvénient du réseau de neurones.

## IV.3.2. Support Vector Machines

La méthode Support Vector Machines (SVM) est une des méthodes les plus connues et les plus utilisées dans les méthodes de DM car elle permet d'obtenir un des meilleurs taux d'erreur par rapport aux autres méthodes. Ainsi ce sera une référence car elle nous donnera un élément de comparaison dans notre futur choix. Cette méthode est basée sur la séparation des différents états de sortie grâce à des hyperplans. Elle va donc rechercher un hyperplan optimal pour la séparation de nos données. Pour des cas non séparables linéairement on peut appliquer une fonction de Kernel. Cela revient à faire une projection dans une autre dimension permettant enfin de séparer nos données de façon linéaire. Le choix de cette fonction va donc se révéler important quand à la réponse de notre système. On peut faire plusieurs projections voir même une infinité de projection ,en théorie n'importe quel ensemble est séparable selon cette méthode.

### Avantages et inconvénients de la méthode :

- La méthode passe par une fonction de Kernel lorsque les données ne sont pas séparables de façon linéaire (ce sera sans doute notre cas). Bien qu'elle soit plus rapide à entraîner, elle est

aussi plus longue en temps réel car elle doit calculer cette fonction à chaque moment contrairement au cas de la solution neuronal.

- Le taux de réponse correct est généralement bon en comparaison des autres méthodes (Tango F. , Botta, Minin, & Montanari, 2010).
- Il semble plus compliqué d'obtenir une sortie continue que dans le cas du réseau de neurones.

### IV.3.3. Adaptive Network-based Fuzzy Inference systems (ANFIS)

Cette dernière méthode couple l'apprentissage du réseau de neurone à la logique floue. Le principe de l'apprentissage est le même que celui des réseaux de neurones classique. A partir des données que nous lui injectons, la méthode va nous donner des règles basées sur la logique floue.

Nous allons ici détailler ce qu'est l'architecture du neuro-flou. Ce dernier va permettre de construire un système flou grâce aux réseaux de neurones. Le fonctionnement d'un réseau neuro-flou particulier Adaptative-Neuro-based Fuzzy Inference Système (ANFIS) est plus détaillé dans la littérature (Jang & Sun, 1995). Nous allons entraîner un réseau qui nous permettra de tuner nos règles pour chaque entrée et donner des poids de sortie correspondant à un optimal. L'architecture générale de ce type de réseau est donnée par la figure ci-dessous.

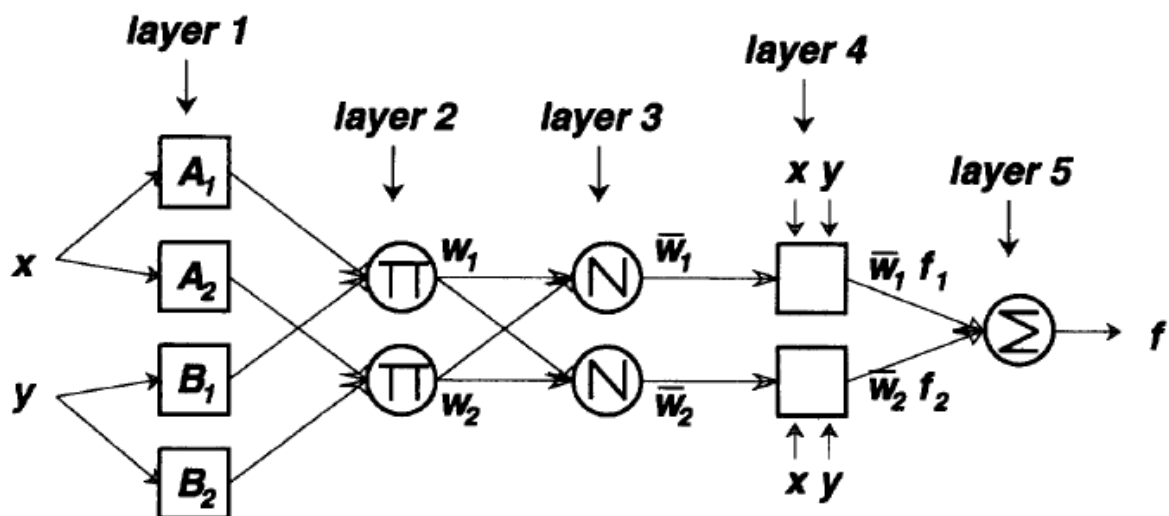


Figure 14 architecture d'un système neuro-flou ANFIS(extrait de Jang & Sun,1995)

#### Architecture neuro-flou

##### Couche 1 : La fuzzification

Selon la forme des séparations floues choisie il y aura des paramètres à ajuster pour chaque fonction. Si nous prenons par exemple des fonctions séparatrices du type trapézoïdales il y aura deux

paramètres à choisir pour l'addition de chaque fonction d'appartenance. Ces paramètres vont être modifiés par l'apprentissage du réseau de neurones et les données d'apprentissage.

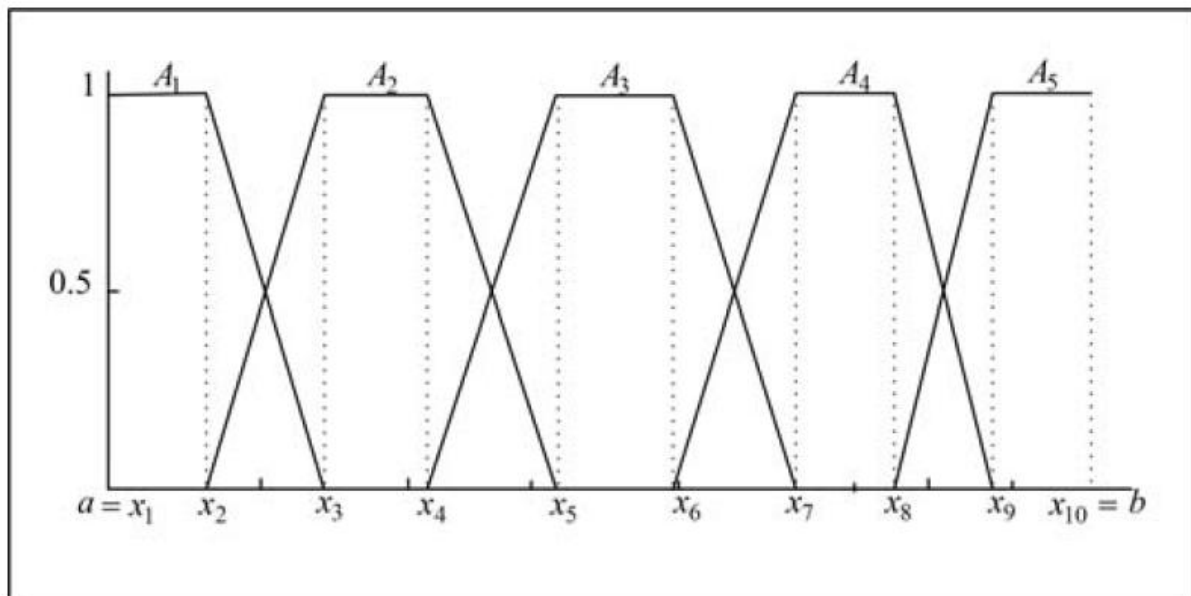
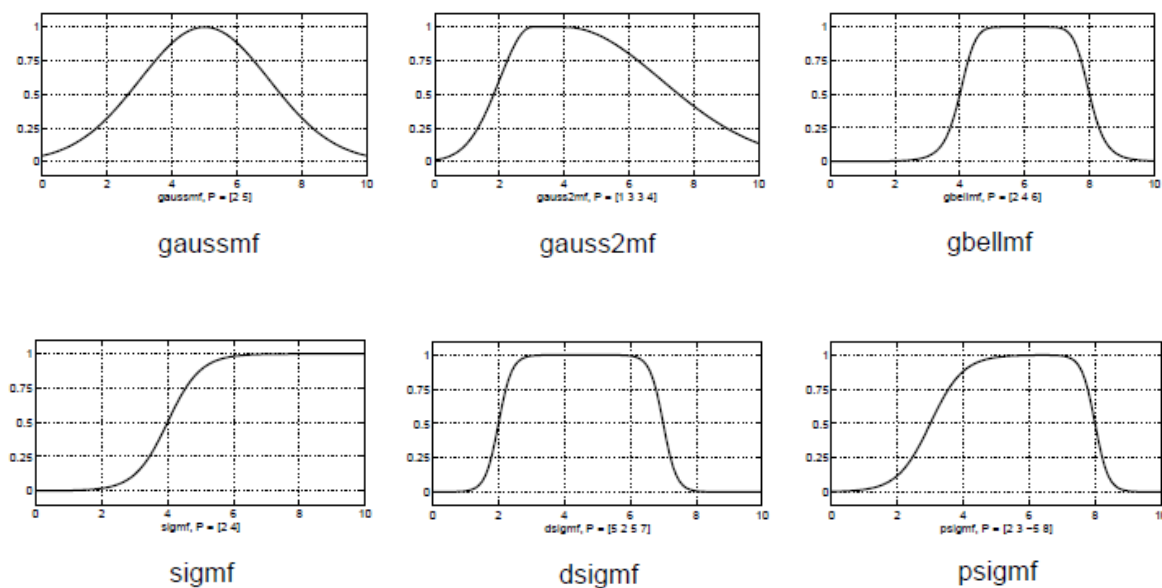


Figure 15 : exemple de partition avec 5 fonctions d'appartenance

Les fonctions d'appartenances sont nombreuses. Nous en avons un résumé dans la figure qui suit.



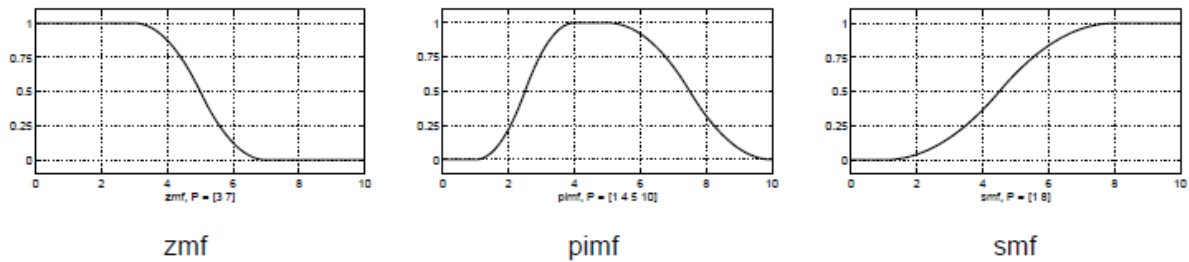


Figure 16 : différentes fonctions séparatrice avec leur sigles

## Couche2 : Multiplication des entrées entre elles

Selon la forme du réseau choisie, le système va combiner les différentes sorties de la couche précédente (la fuzzification) en les multipliant entre eux. Pour cette raison nous n'avons pas formule générique pour cette couche.

## Couche3 : Normalisation des données d'entrées

$$W_j = \frac{W_j}{\sum_i W_i}$$

## Couche 4 : Couche adaptative avec modulation des poids

Dans cette couche la méthode va adapter les coefficients  $p_i$ ,  $q_i$  et  $r_i$  pour répondre au mieux aux sorties demandées. C'est coefficients font partie du réseau. C'est en partie grâce à eux que le système va s'optimiser.

$$w_i f_i = w_i * (p_i * x + q_i * y + r_i)$$

## Couche 5 : recombinaison des sorties

Le système recombine alors tous les outputs de la couche 4 pour donner l'output désiré.

$$f = \sum_i w_i f_i$$

## Avantages et inconvénients de la méthode :

- Le temps de réponse en temps réel est très court malgré un temps d'entraînement plutôt long comparé aux autres méthodes. En effet la méthode est consommatrice de cas et a besoin de beaucoup de données pour fonctionner.
- La sortie floue autorise différents niveau de réponse. Un point très positif dans notre cas où on cherche à obtenir différents niveaux de tunnélisation attentionnelle en sortie. Pour cela il faut quand même définir ces niveaux pour les données en entrainement. Le système ne les



général pas lui-même. On va avoir une marge grâce à l'erreur quadratique qui pourrait peut-être être associé à un niveau intermédiaire de tunnélisation attentionnelle (voir les perspectives).

- L'approche floue associée au réseau de neurones nous permet de plus avoir une boîte noire entre les inputs et les outputs contrairement aux réseaux de neurones vus plus haut (Catto, et al., 2003). Les règles floues sont explicitement données. Par leurs caractères linguistiques nous avons une compréhension rapide et claire des règles importantes.

#### **IV.3.4. Choix de la méthode d'apprentissage**

En considérant tous les avantages et inconvénients de chaque méthode que nous avons étudiée dans l'état de l'art nous avons choisi de sélectionner l'option neuro-flou pour avoir une boîte permettant d'avoir un diagnostic. D'après la littérature c'est l'option la plus crédible que nous avons actuellement étudiée. Nous allons quand même faire une comparaison avec la méthode SVM pour nous assurer que la réponse que nous obtenons a un taux d'erreur correct.

## V. Mise en œuvre avec ANFIS sur Matlab

Nous avons opté pour l'approche neuro-floue. Nous présentons ici l'environnement Matlab qui a permis de mettre en œuvre cette méthode. Par ailleurs, les résultats que nous avons obtenus seront comparés avec les SVM, une technique de référence en DM.

### V.1. Description de l'interface ANFIS sous Matlab

Les méthodes de DM permettent en général de tirer une/des règles afin de répondre au problème donné, à savoir un certain nombre d'inputs donnant un certain nombre d'outputs. Avec le bruit, les données fausses, il se peut que la règle ne soit pas forcément juste. C'est pour cela qu'il convient de fonctionner par étapes pour ne pas obtenir de règles qui soient juste dans notre cas précis mais non généralisable. Les trois étapes à suivre sont énoncés ci-dessous.

- **Etape d'entraînement** : Lors de cette étape, nous allons fournir à la machine un package de données qui lui permettra d'en sortir une règle générale. Le système boucle sur lui-même jusqu'à obtenir une erreur minimale (que nous décidons au départ).
- **Etape de vérification** : Cette deuxième étape est cruciale. Elle fonctionne avec un package de données indépendante de la première étape. Elle se fait en parallèle et elle décide de l'arrêt de la boucle. A partir du moment où l'erreur sur le package de données augmente de nouveau cela signifie que nous sommes en train de trouver des règles non généralisables. Il faut donc arrêter la boucle au minimum de l'erreur de ces données.
- **Etape de test** : nous prenons encore un set de data extérieur aux autres pour, après tout apprentissage, analyser la réponse du système enfin créé.

La répartition des sujets entre les différentes étapes est importante. Dans des systèmes assez complexes, l'erreur va atteindre un minima pour un certain nombre de cas étudiés à l'entraînement. Dépassant ce nombre de cas on augmente donc l'erreur faite sur le modèle. L'étape de vérification nous permettra de nous en rendre compte au fur et à mesure. Cela va justifier que les règles que nous trouvons sont généralisables car elles peuvent s'appliquer à des personnes dont les mesures n'ont pas été exploitées pour notre système. Des papiers sur les méthodes de DM nous indiquent des répartitions suivantes (Tango & Botta, Evaluation of Distraction in a Driver-Vehicle-Environment Framework: An Application of Different Data-Mining Techniques, 2009) (Mukkamala, Janovski, & Sung, 2002) : 60% de données pour l'entraînement et 40% pour la vérification. C'est ce que nous allons suivre dans la suite de notre apprentissage.

La figure ci-dessous nous donne l'interface que nous avons sous Matlab pour la méthode neuro-fuzzy. L'écran central nous permet d'accéder aux fenêtres tout autour.

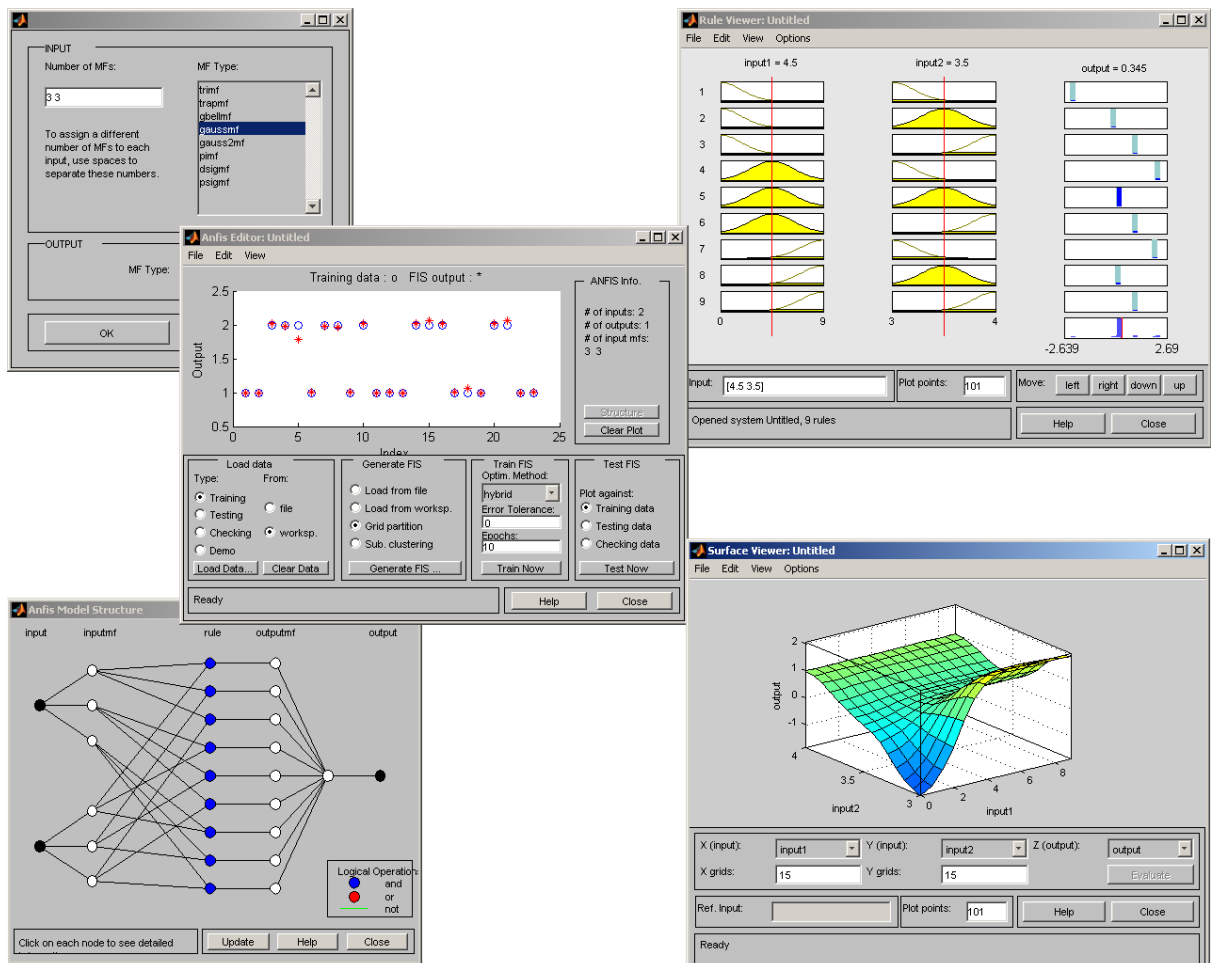


Figure 17 : Interface ANFIS sous Matlab

Nous fonctionnons en interface Matlab et ANFIS fait parti de la Toolbox Fuzzy. Elle est assez simple à utiliser. Dans un premier temps nous devons préparer notre ensemble de données de la façon suivante : ce sera des matrices avec nos entrées sur toutes les premières colonnes et la sortie que nous désirons sur la dernière colonne. Nous pouvons charger la fenêtre centrale grâce à la commande *anfisedit* sur Matlab. A partir de cette fenêtre nous allons charger les données d'entraînement et de checking que nous avons préparées auparavant.

Ensuite il faut choisir un FIS (fenêtre en haut à gauche), c'est le modèle duquel on veut que notre système se rapproche. On peut ainsi choisir le nombre  $m$  de fonctions d'appartenance (Membership Fonctions) que nous souhaitons pour chaque entrée, et le type de fonction que nous voulons. Nous devons donc savoir en combien d'espaces différents nous voulons séparer nos données. Ces choix vont nous donner le nombre de règles  $r$  auxquelles vont répondre le système selon la formule ci-dessous. On considère sur cette dernière qu'il y a au moins deux entrées. Le cas avec une seule entrée est trivial.

$$r = m_1 * \dots * m_i$$

## Avec mi le nombre de séparation de l'entrée i

Nous avons une représentation des liens entre les MFs, les inputs et les outputs (écran en bas à gauche). De même les règles floues obtenues sont visibles dans un autre écran (en haut à droite). A partir de ce dernier nous pouvons choisir de tester des entrées diverses pour étudier le comportement du système entraîné.

### Apprentissage sur ANFIS

L'apprentissage avec ANFIS sous Matlab va permettre d'ajuster tous les paramètres que nous avons présentés précédemment. Des données de références nous permettront de les choisir pour obtenir une sortie qui soit le plus en phase avec celle que nous recherchons. La méthode la plus utilisée s'appelle la rétro-propagation. Nous allons la détailler dans la suite de ce paragraphe.

Le système ANFIS va décrire une boucle. Celle-ci va permettre de déterminer dans un premier temps qu'elles sont les connections responsables de l'erreur et de combien elle participe à cette dernière. En fonction de ces informations on va pouvoir modifier les poids qui ne vont pas.

L'erreur que nous utilisons est du type quadratique (le carré de la différence entre la sortie voulue et de la sortie obtenue). Le principe est modifier le poids des entrées du réseau en fonction de l'erreur que nous obtenons. Ainsi le système baissera le poids des neurones ayant le plus participé à l'erreur et en augmentera le poids des autres neurones.

## V.2. Modélisation de la tunnélisation attentionnelle avec ANFIS

Les données que nous avons mises en entrée sont des données qui ressortaient dans la littérature. Le tableau ci-dessous résume les quatre indicateurs connus de la tunnélisation attentionnelle dans la littérature (première et deuxième colonne). La troisième colonne exprime leur "traduction" sous la forme de métriques utilisables dans le cadre de nos expérimentations robotiques présentées précédemment.

Indicateur de la tunnélisation attentionnelle	Référence	Métrique choisie
Allocation des ressources attentionnelles sur un cadran particulier	(Wickens, 2005)	% de temps passé sur une AOI
Activité saccadique réduite	(Cowen, Ball, & Delin, 2002) (Tsai, Viire, Strychacz, Chase, & Jung, 2007)	Taux de transition oculaire entre AOI

Réduction du nombre de zone d'intérêt (AOI) considérées	(Thomas & Wickens, 2004)	Nombre d'AOI
<b>Stress psychophysiological</b>	(Easterbrook, 1959) (Weltman & Egstrom, 1966) (Bahrick, Fitts, & Rankin, 1952) (Tracy, et al., 2000)	Rythme cardiaque

Nous avons, pour notre cas, choisi 4 métriques dont voici la définition plus explicite :

- La différence entre le rythme cardiaque du candidat pendant la mission et celui au repos moyenné sur 20 secondes
- Le pourcentage de temps passé sur l'écran de vidéo durant les 10 dernières secondes
- Le nombre d'AOI visitées durant les 20 dernières secondes
- Le taux de transition oculaire durant les 10,5 dernières secondes

Ces différentes métriques ont été choisies par Sergio PIZZIOL travaillant en thèse à l'ONERA sur le sujet de la tunnélisation attentionnelle.

Nous avons vu précédemment que nous devons choisir le nombre de fonctions d'appartenance (nombre de séparation) pour chaque entrée et la forme qu'elles vont avoir. Cette opération se fait surtout par expérience. Ainsi nous devons régler notre méthode pour la rendre optimale. Ce sera la première action à faire dans la partie résultat.

Le système ANFIS va donc nous régler un réseau de 24 règles floues. Pour rappel il va modifier les places des fonctions d'appartenances et les poids de sortie pour coller au mieux aux données d'entraînement que nous lui avons soumises.

## V.3. Résultats

### V.3.1. Réglage de l'outil pour un bon apprentissage

- **Quelles données pour notre apprentissage ?**

Dans le cadre de la tunnélisation attentionnelle nous ne disposons d'aucune connaissance experte. Nous avons un problème d'indicateur objectif qui nous dirait quand la personne est tunnélisée. Nous avons choisi de prendre l'alarme de batterie comme indice. Elle sera notre indicateur objectif. Nous pourrions considérer comme tunnélisée une personne qui entre en conflit avec le robot et ne détecte pas la panne à partir de la phase 4 et jusqu'à la fin. Et comme non

tunnélisée une personne qui aura vu la panne grâce à la contre-mesure et du même coup pris conscience du problème. Grâce à la phase 4 de la mission nous avons donc un ensemble de données qui vont nous servir à entraîner le système ANFIS. Nous avons enlevé tous les cas que nous trouvons litigieux (car nous ne pouvons pas les considérer comme complètement tunnélisées où non tunnélisées).

Nous voulons un ensemble de données qui soit le plus propre possible. Pour cela nous avons considéré les personnes ayant reçu une contre-mesure comme non-tunnélisée (sortie égale à 0). Il y a un temps normal de détunnélisation. La contre mesure prend déjà 6 secondes à se mettre en place. De plus des études chez Airbus ont permis de se rendre compte que le temps moyen de réaction d'un pilote face à une alarme « Pull Up » est de 3 à 5 secondes. Nous avons donc choisi de retirer les 10 premières secondes considérant que la personne n'est pas encore complètement détunnélisée. Pour les personnes n'ayant pas reçu de contre-mesure, nous n'avons choisi que les données pendant lesquelles le volontaire entrait en conflit avec le robot (sortie égale à 1). Ces choix sont résumés dans la figure ci-dessous.

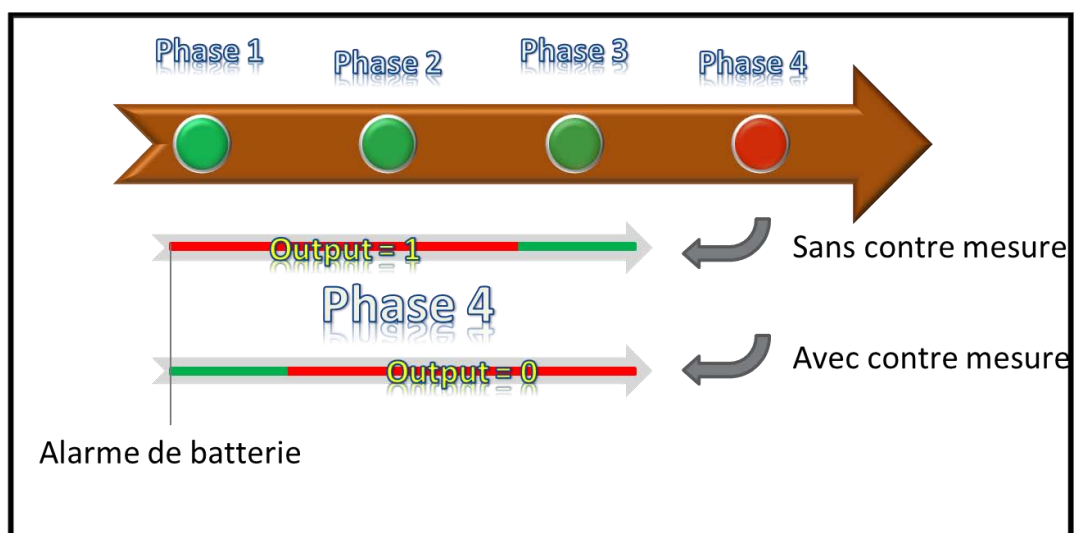


Figure 18 : Résumé du choix des données d'entraînement

L'entraînement peut maintenant commencer à partir de ces données. Il y a un certain nombre de paramètres à choisir dans ANFIS avant de commencer l'apprentissage. Nous allons les voir plus en détail dans le paragraphe suivant lorsque nous réglerons la méthode.

A partir de ce package de données nous pouvons les diviser en deux parties. Une première partie servira à l'entraînement et la deuxième servira au checking comme nous l'avons vu plus haut. Nous allons mettre 60% des données dans le premier set et les 40% restant dans le deuxième (Mukkamala, Janovski, & Sung, 2002). L'entraînement s'est fait sur un certain nombre de boucles pendant lesquelles la méthode va minimiser le taux d'erreur faite sur la partie entraînement. Pour obtenir un optimal il faut attendre que l'erreur sur le set de checking remonte (voir graphe ci-dessous). On

constate sur cette dernière que le système est optimum pour un nombre d'itérations (epochs) de 110.

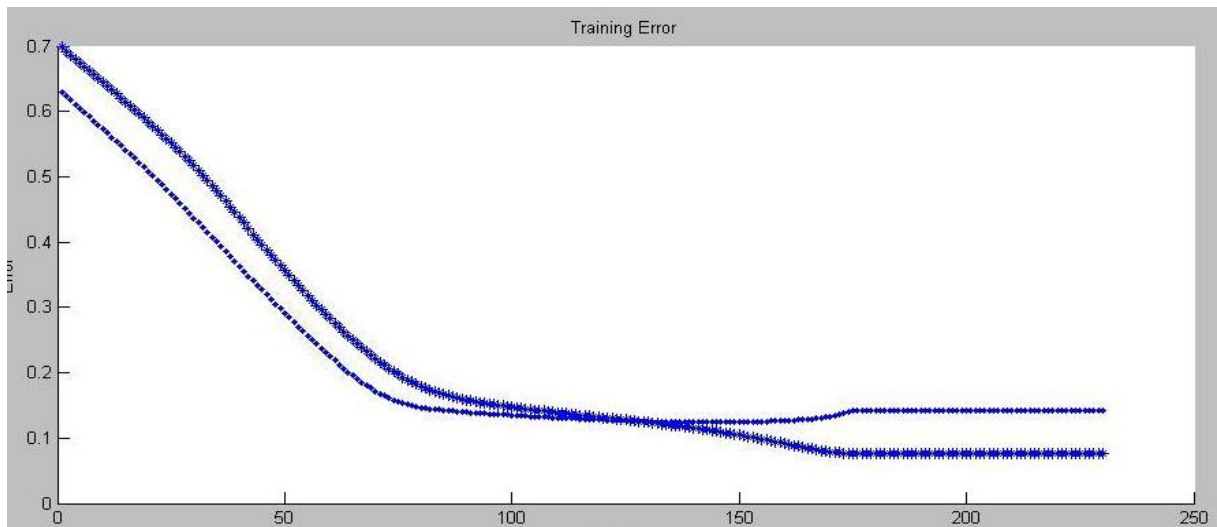


Figure 19 : Exemple d'entraînement d'un système ANFIS

- **Réglage optimale de l'outil ANFIS**

Dans un premier temps nous devons savoir quelle forme de fonction d'appartenance nous allons choisir pour faire notre apprentissage. Nous avons commencé par faire une première série de test sur le set des fonctions d'appartenance que nous propose ANFIS. Nous avons gardé un nombre de séparation de 2 pour chaque entrée. Les résultats sont résumés dans le tableau ci-dessous. Nous nous basons sur l'erreur maximale entre l'erreur sur le package d'entraînement et celui de test. A ce titre nous voyons que l'erreur est moindre avec des fonctions d'appartenances trapézoïdales.

Fonction d'appartenance	Erreur quadratique entraînement	Erreur quadratique test	Nombre d'itérations
Triangulaire	15.9%	17.4%	82
Trapézoïdale	11.2%	11.4%	63
Gbell	11.4%	12.2%	82
Gaussienne	12.1%	13.2%	88
Gaussienne2	11.9%	11.8%	65
Pimf	13.4%	12.5%	62
Dsigmf	13%	12.7%	63

<b>Psigmf</b>	13.4%	12.7%	62
---------------	-------	-------	----

Nous essayons par la suite de savoir combien de fonctions d'appartenance sont utiles pour chaque entrée. Par défaut, toutes les entrées ont 2 séparations trapézoïdales. Nous avons essayé d'en mettre 3 pour certaines. Nous voulons ainsi savoir si cela améliore notre modèle. Les résultats sont résumés dans le tableau ci-dessous. Ainsi seul le rythme cardiaque nécessite plus de fonctions d'appartenance. Dans la suite elle sera séparée en 3 parties. Nous avons essayé de lui en mettre 4 et nous constatons que l'erreur devient plus grande.

<b>Entrée dont la fonction d'appartenance est de 3</b>	<b>Erreur quadratique entraînement</b>	<b>Erreur quadratique test</b>	<b>Nombre d'itérations</b>
<b>Aucune</b>	11.2%	11.4%	63
<b>Rythme cardiaque</b>	5.4%	4%	720
<b>Pourcentage sur la vidéo</b>	12.6%	10.2%	62
<b>Nombre d'AOI visitées</b>	17.7%	14.7%	79
<b>Taux de transition oculaire</b>	5.9%	15.9%	infini
<b>4 fonctions d'appartenance pour le rythme cardiaque</b>	6.9%	32.6%	150

Nous avons par la suite essayé de savoir quelles étaient les métriques importantes pour le diagnostic de la tunnélisation attentionnelle et si chaque métrique participait au diagnostic final. Nous cherchons à minimiser le taux d'erreur. Nous gardons le réglage que nous avons trouvé précédemment. Toutes les entrées auront 2 fonctions d'appartenance trapézoïdale sauf le rythme cardiaque qui en aura 3. Nous considérons pour cette phase que nous ne nous sommes pas trompés dans le diagnostic de la dernière phase. Comme nous le disions, les séquences participant à l'entraînement étaient minutieusement sélectionnées et dès que nous avons un doute nous ne les prenons pas. Nous avons obtenues les résultats qui suivent dans le tableau ci-dessous.

<b>Type</b>	<b>Erreur quadratique entraînement</b>	<b>Erreur quadratique test</b>	<b>Nombre d'itération</b>
-------------	--	--------------------------------	---------------------------



<b>Toutes les métriques</b>	5.4%	4%	720
<b>Tout sauf le rythme cardiaque</b>	8.5%	11.4%	52
<b>Tout sauf pourcentage sur la vidéo</b>	5.5%	4.1%	750
<b>Tout sauf le taux de transition oculaire</b>	5.7%	2.6%	750
<b>Tout sauf le nombre d'AOI visitée</b>	20.38%	44.8%	80
<b>Tout sauf le pourcentage sur la vidéo et le taux de transition oculaire</b>	9.3%	14.4%	130

Ce tableau justifie à lui seul que nous utilisons toutes les métriques. Si nous en retirons seulement une seule l'erreur quadratique que nous faisons augmente. Chaque métrique a son importance même si il ressort clairement que le nombre d'AOI visitées est plus important que les autres métriques.

### V.3.2. Benchmark de deux méthodes : le Neuro-Fuzzy avec ANFIS sur Matlab et le SVM de libSVM sur Matlab

Une campagne de test nous a permis de comparer le neuro-flou avec ANFIS et la méthode SVM avec libSVM que nous avons testé sur Matlab. Le but de cette campagne était de savoir si le SVM, méthode la plus utilisée à ce jour pour son efficacité dans la réponse, répondait mieux que le neuro-flou, méthode encore très peu utilisée dans la littérature aujourd'hui. Pour pouvoir comparer ces deux méthodes il faut d'abord avoir un indicateur nous permettant de faire la comparaison. Sur ANFIS nous fonctionnons sur l'erreur quadratique moyenne alors que sur SVM la réponse est binaire. Ce n'est donc pas le même pourcentage d'erreur. Pour pouvoir comparer les deux nous avons préféré rapporter la réponse de l'ANFIS à une réponse binaire. Ainsi nous obtenons deux valeurs d'erreur qui nous permette de voir si le modèle ANFIS représente une amélioration comparé au modèle SVM.

La réponse des systèmes va dépendre des choix de paramètres que nous allons faire au début. Cette campagne de test va se baser sur l'expérience. Nous avons testé un grand nombre de possibilités pour voir quand le système répond le mieux et ainsi pouvoir comparer une des méthodes les plus utilisées dans la littérature par rapport à la méthode que nous souhaitons utiliser pour son aspect continu.

Pour le SVM la figure résume les résultats que nous avons obtenus sur cette campagne. Nous avons testé les différentes fonctions de Kernel possible en laissant les autres paramètres par défauts.

Type de fonction Kernel utilisée	Erreur sur l'entraînement	Erreur sur le test
Linéaire	99.48% (396/398)	96.60 (256/265)
Polynomiale	100% (398/398)	97.74% (259/265)
Fonction à base radiale	100% (398/398)	89.05% (236/265)
Sigmoïde	51.26% (204/398)	60.38% (160/265)

Pour le neuro-flou sur ANFIS, nous avons testé les différentes fonctions d'appartenances possibles avec un nombre de séparation de 2 à chaque fois. C'est le nombre minimal de séparations que nous pouvons donner à chaque entrée. Nous obtenons les résultats résumés dans le tableau ci-dessous.

Type de fonction d'appartenance	Erreur binaire sur l'entraînement	Erreur binaire sur le test
Triangulaire	99.5%	100%
Trapézoïdale	98.7%	99.7%
Fonction « Gbell »	98.7%	97.7%
Gaussienne	99.5%	98.1%
Gaussienne2	98.5%	97.7%
« Pi shape membership function »	98.5%	97.7%
Différence de deux sigmoïdes (Dsigmf)	98.7%	97.7%
Produit de deux sigmoïde (Psimf)	98.5%	97.7%

Au vue du tableau ci-dessus, nous pouvons affirmer que le taux d'erreur est très faible avec ANFIS. Même avec un nombre de séparation très faible nous arrivons à obtenir de meilleurs résultats qu'avec le SVM et tous les paramètres par défaut. Cela justifie notre choix quant à la méthode de Data Mining.

### V.3.3. Résultats obtenus grâce à la méthode neuro-floue

Grâce à la méthode neuro-floue, nous avons pu construire un système de règles floues qui prennent les entrées dont nous avons discutées précédemment. Elle nous donnera en sortie un indice entre 0 et 1 qui nous dira si, selon le système, la personne est tunnelisée ou non.

Nous en avons mis une représentation ci-dessous. C'est le réseau optimal auquel nous sommes parvenus. Nous avons séparé chaque entrée en deux fonctions d'appartenance sauf celle du rythme cardiaque qui avait besoin d'une séparation en trois domaines. Cela nous donne un total de 24 règles que le système a optimisé. L'erreur quadratique moyenne que nous avons trouvée sur ce système est de 5,4%.

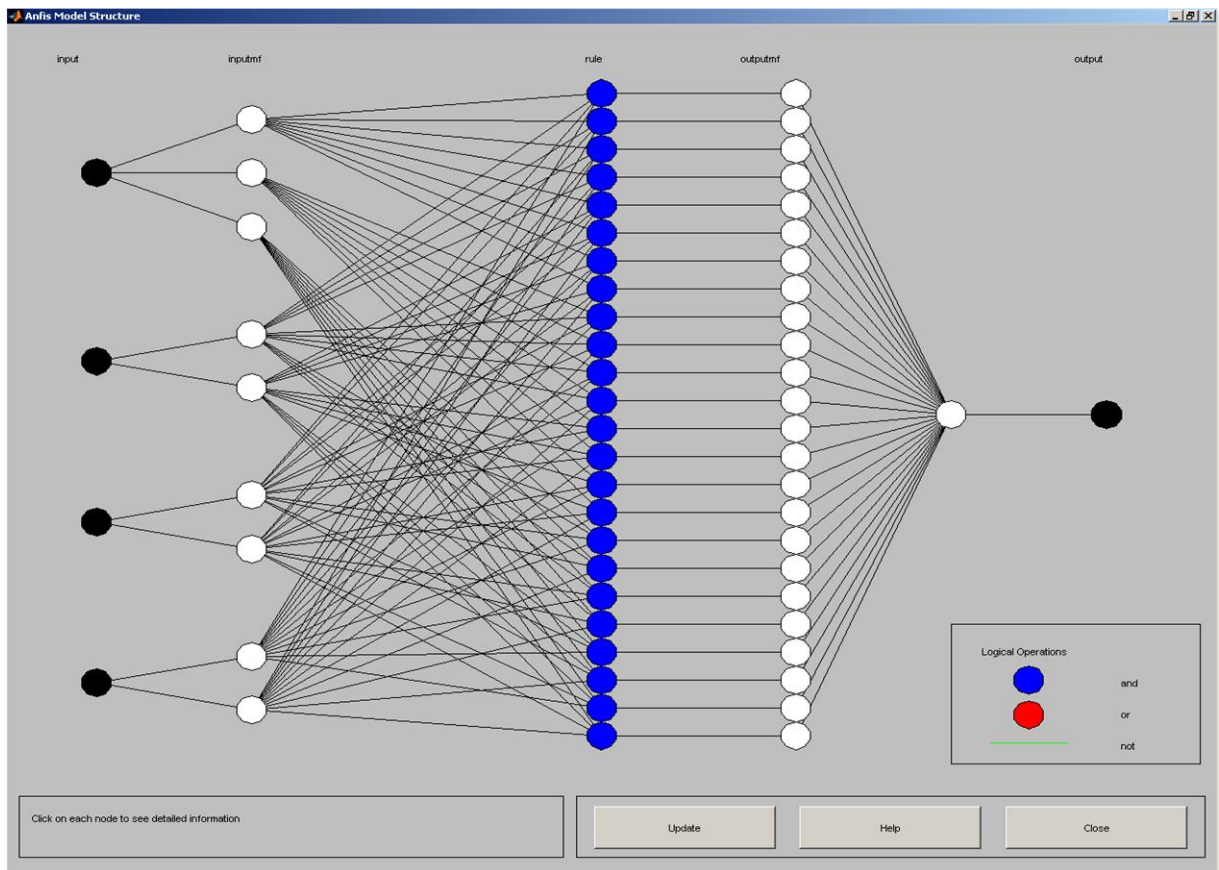


Figure 20 : réseau de neurones obtenu après réglage

Nous avons pu étudier toutes la mission grâce au système ainsi obtenu et analyser le début de la mission. Ce sera la dernière partie. Faire cela revient à voir si notre règle est généralisable à des phases qui ne sont pas les mêmes. Cela permettra dans un premier temps de savoir si les résultats que nous trouvons sont aberrants. Nous avons résumés les choix finaux dans le tableau ci-dessous.

Ces choix sont issus de la partie précédente dans laquelle nous avons cherché le système répondant le mieux.

Métrique	Rythme cardiaque	Pourcentage de temps passé sur la vidéo	Nombre d'AOI visitées	Saccade
Nombre de fonction d'appartenance	3	2	2	2
Type de fonction d'appartenance	Trapézoïdale	Trapézoïdale	Trapézoïdale	Trapézoïdale

Nous allons étudier quelques cas dans cette partie pour analyser la réponse de notre système. Trois cas seront présentés. Le premier, sujet Gabje, a reçu une contre mesure au début de la 4<sup>ème</sup> phase. Le deuxième, Rojan, n'a pas reçu cette contre mesure et est entré en conflit avec la machine jusqu'à la fin de l'expérience. Enfin le troisième, Pense, a réussi à détunnéliser tout seul. Ce dernier cas n'a pas servi à l'apprentissage.

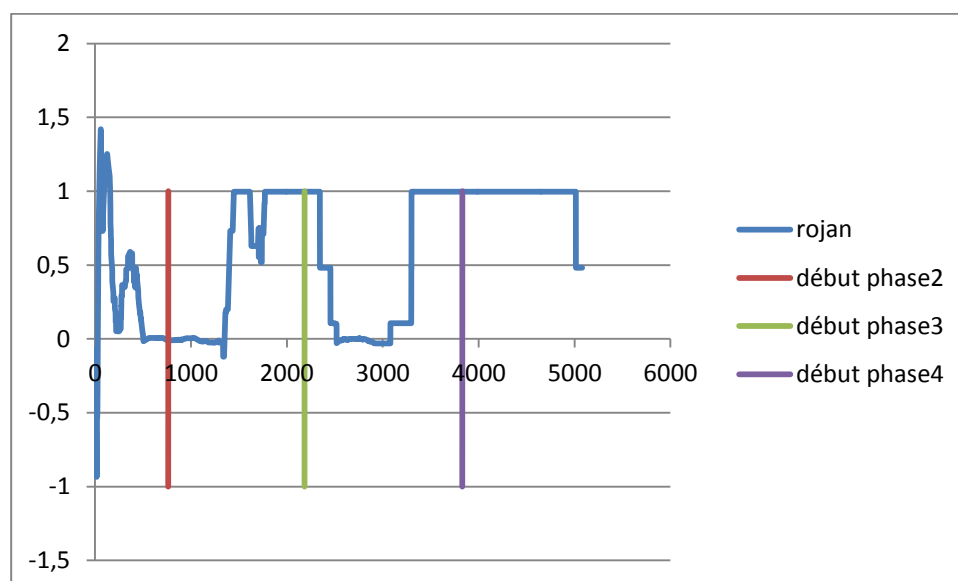


Figure 21 : Un cas sans contre-mesure Rojan. Il monte en pression à la fin de la phase 3. Sans contre-mesure, il reste tunnélisé jusqu'à la fin de la phase 4. C'est pour ce genre de personnes que nous voulons détecter l'état de tunnélisation à l'avance

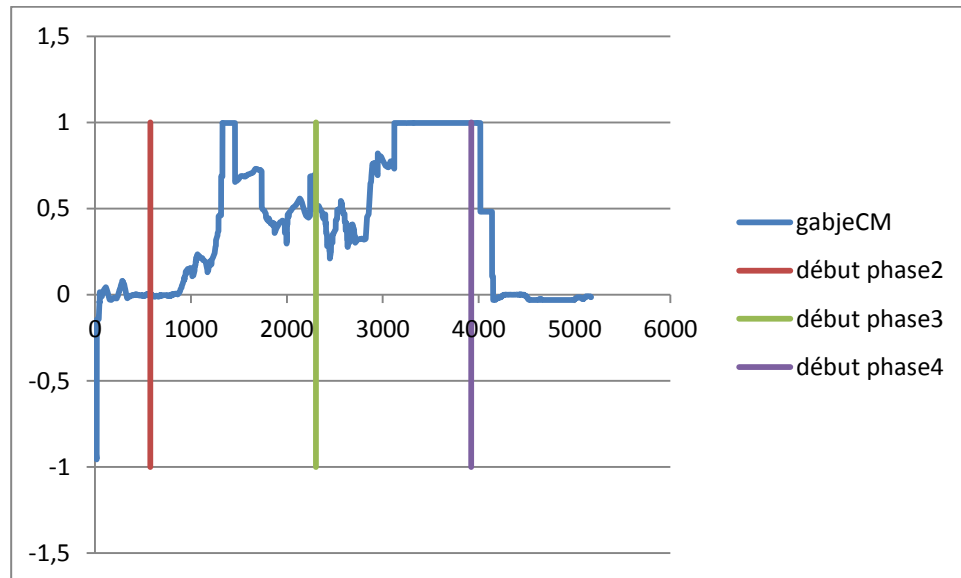


Figure 22 : Un cas de contre-mesure Gabje. Nous constatons une montée en pression pendant la phase 3 (lorsque le sujet pilote le drone manuellement) avant le début de la phase 4. Il détunnélise ensuite grâce à la contre-mesure

Sur ces deux premiers cas nous pouvons à chaque fois constater une montée de l'indice de tunnélisation attentionnelle avant le début de la phase 4. Selon notre système le candidat va se focaliser sur la tâche de conduite de l'appareil jusqu'à oublier qu'il y a des alarmes. Cette dernière apparaît en début de phase 4. Le candidat qui n'a pas de contre mesure va alors s'entêter dans la réussite de la mission et ainsi perdre l'appareil. La sortie tunnélisation attentionnelle est alors égal à 1 jusqu'à la fin de la mission. Quant à la personne ayant reçu une contre-mesure, Elle va laisser le robot rentrer tout seul comme la procédure l'implique en cas d'alarme de batterie. Une fois de plus la sortie tunnélisation attentionnelle nous indique une chute juste après le début de la dernière phase jusqu'à atteindre un nombre proche de zéro indiquant une personne non tunnélisée.

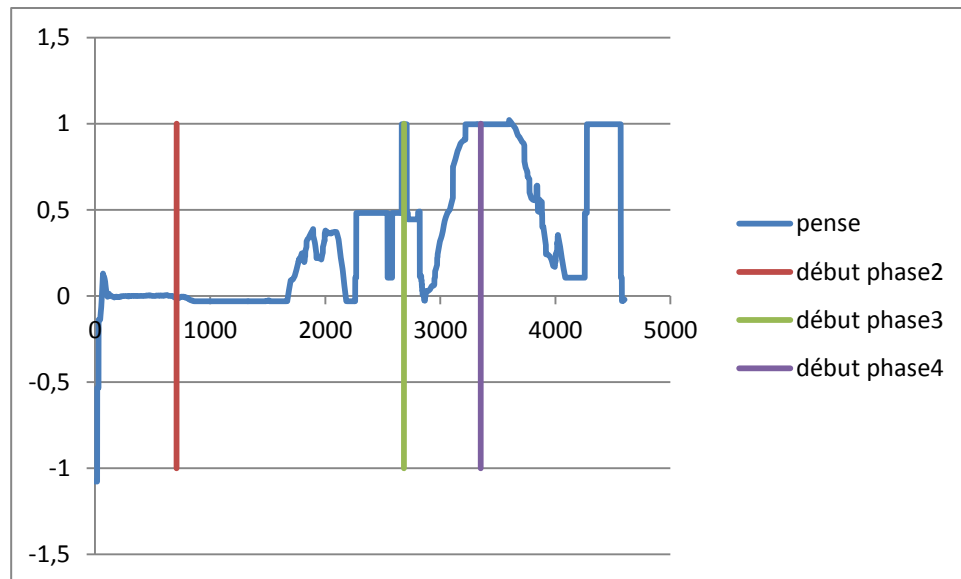


Figure 23 : Un cas de détunnélisation sans contre-mesure Pense

Ce dernier cas est une des personnes qui n'avait pas de contre mesure et qui a détunnélisée toute seule. Nous pouvons constater que la détunnélisation est visible sur notre diagnostic et qu'elle aussi plus lente. Selon notre diagnostic la personne a plus de difficultés à se rendre compte de ce qui ne va pas comparé à la chute de tunnélisation attentionnelle qu'on peut observer sur des cas ayant reçu une contre-mesure.

## Conclusion et travaux futurs

Pour résumer, nous avons réussi à obtenir un système diagnostiquant la tunnélisation attentionnelle sans passer par une construction manuelle. La méthode ANFIS nous a permis de créer un système sans à priori diagnostiquant la tunnélisation attentionnelle avec des résultats encourageant pour la suite. Il reste tout de même des améliorations à apporter pour la suite qui sont détaillé dans les perspectives si jointes.

Nous cherchons dans un premier temps à avoir un diagnostic le plus juste possible. C'est le critère principal. La méthode que nous avons choisie semble y répondre de manière optimale. Pour pouvoir être plus robuste notre système a besoin de plus d'entrée. Ce sera le rôle de Nicolas Régis dans son futur travail de thèse de trouver de nouvelles métriques permettant de détecter la tunnélisation attentionnelle. Une piste intéressante serait de représenter un vecteur d'état de l'œil. Cela nous permettrait de nous passer des zones d'intérêts. Dans le futur nous allons tester les métriques de Nicolas REGIS.

Concernant la suite que je donne à mon travail, nous partons sur une piste d'arbre de diagnostic. Cette technique marche sur le principe que plus il y a d'expert et plus le diagnostic sera robuste. L'étude de cette nouvelle technique nous permettra dans un premier temps de comparer les diagnostics apportés par le neuro-flou et cette nouvelle méthode. Ainsi si par une autre méthode de Data Mining nous avons le même diagnostic alors cela nous confortera dans l'exactitude de notre méthode.

Une future approche pourrait aussi être de se demander ce que veut dire « moyennement tunnélisé ». Lorsque notre système donne un résultat à 0.5, ça ne veut pas dire que l'opérateur est moyennement tunnélisé. Cela signifie que le système n'arrive pas à donner un diagnostic. Il faudrait vérifier par la suite si cette marge se rapproche d'un niveau de tunnélisation attentionnelle.

Le gros problème de la méthode que nous avons utilisée est qu'elle est très consommatrice de cas. Pour pouvoir être sûr de notre diagnostic il faudrait avoir plus de candidat. Une future étude permettra d'étudier de combien de personnes il est nécessaire d'obtenir des informations avant de pouvoir affirmer que notre système est vrai.

Enfin nous avons déjà commencé à mettre en place le diagnostic temps réel grâce à simulink. Le système crée par ANFIS sur Matlab est exportable sur simulink. Il suffira ensuite de connecter les entrée physiologique et eye tracker du volontaire pour avoir un diagnostic en direct et pouvoir agir sur l'IHM et l'adapter en fonction de l'état de la personne. Ainsi il sera possible de vérifier l'efficacité de notre système. Un problème subsiste quand même. Pour l'instant nous ne pouvons pas encore détecter les zones d'intérêts en temps réel. Normalement Pertech devrait améliorer cela dans la prochaine version du logiciel de traitement de l'information eye tracker.

## Bibliographie

- Bahrack, H., Fitts, P., & Rankin, R. (1952). Effect of incentives upon reactions to peripheral stimuli.
- Barreto, A., Zhai, J., & Adjouadi, M. (2007). Non-intrusive Physiological Monitoring for Automated Stress Detection in Human-Computer Interaction.
- Catto, J. W., Linkens, D. A., Abbod, M. F., Chen, M., Burton, J. L., Feeley, K. M., et al. (2003). Artificial Intelligence in Predicting Bladder Cancer Outcome: A comparison of Neuro-Fuzzy Modeling and Artificial Neural Network.
- Causse, M., Pavard, B., Senard, J.-M., Demonet, J.-F., & Pastor, J. (2007). Emotion induction through virtual avatars and its impact on reasoning : Evidence from autonomous nervous system measurements and cognitive assessment.
- Cowen, L., Ball, L. J., & Delin, J. (2002). An eye-movement analysis of web-page usability.
- Dehais, F. (2009). Method and Device for Detecting absences of manual and automatic piloting of an aircraft.
- Dehais, F., Causse, M., Vachon, F., & Tremblay, S. (2011). cognitive conflict in human-automation interactions : A psychophysiological study.
- Dehais, F., Sisbot, A., Alami, R., & Causse, M. (2011). Physiological and subjective evaluation of a human-robot object hand over task.
- Dehais, F., Tessier, C., & Chaudron, L. (2003). Ghost : experimenting countermeasures for conflicts in the pilot's activity.
- Dehais, F., Tessier, C., & Chaudron, L. (2003). Ghost : experimenting countermeasures for conflicts in the pilot's activity.
- Dixon, K., Hagemann, K., Basilico, J., Forsythe, C., Rothe, S., Schrauf, M., et al. (2009). Improved Team Performance Using EEG-and Context-Based Cognitive-State Classifications for a Vehicle Crew.
- Donaldson, C., Lam, D., & Mathews, A. (2007). Rumination and attention in major depression.
- Easterbrook, J. (1959). The effect of emotion on cue utilization and the organization of behavior.
- Haarmann, A., Boucsein, W., & Schaefer, F. (2009). Combining electrodermal responses and cardiovascular measures for probing adaptive automation during simulated flight.
- Hills, B. (1980). Vision, visibility, and perception in driving.
- Jang, J.-S. R., & Sun, C.-T. (1995). Neuro-Fuzzy Modeling and Control. *IEEE*.
- John, M., Kobus, D., Morrison, J., & Schmorow, D. (2004). Overview of the DARPA augmented cognition technical integration experiment.
- Kapoor, A., & Picard, R. (2005). Multimodal affect recognition in learning environments.



- Kulic, D., Carter, M., & Brown, V. (1992). A network simulation of thalamic circuit operations in selective attention.
- Liu, C., Conn, K., Sarkar, N., & Stone, W. (2008). Online affect detection and robot behavior adaptation for intervention of children with autism.
- Mandryk, R., & Atkins, M. (2007). A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies.
- Mandryk, R., & Atkins, M. (2007). A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies.
- Marshall, S. (2007). Identifying cognitive state from eye metrics.
- Mukkamala, S., Janovski, G., & Sung, A. (2002). Intrusion Detection Using Neural Networks and Support Vector Machines.
- Nikhil, P., & Lakhmi, J. (2005). Advanced techniques in knowledge discovery and data mining.
- O'Hare, D., Wiggins, M., Batt, R., & Morrison, D. (1994). Cognitive failure analysis for aircraft accident investigation.
- Parasuraman, R., & Rizzo, M. (2007). Neuroergonomics : the brain at work.
- Parasuraman, R., Mouloua, M., & Hilburn, B. (1999). Adaptive aiding and adaptive task allocation enhance human-machine interaction.
- Pecher, C., Quaireau, C., Lemercier, C., & Cellier, J. (2010). The effects of inattention on selective attention : How sadness and ruminations alter attention functions evaluated with the Attention Network Test.
- Philippot, P., & Brutoux, F. (2008). Induced rumination dampens executive processes in dysphoric young adults.
- Pizziol, S., Dehais, F., & Tessier, C. (2011). Towards human operator "state" assessment.
- Pizziol, S., Dehais, F., & Tessier, C. (n.d.). Towards human operator "state" assessment.
- Posner, M., & Dehaene, S. (1994). Attentional networks. Trends in neurosciences.
- Posner, M., Walker, J., Friedrich, F., & Rafal, R. (1984). Effects of parietal lobe injury on covert orienting of visual attention.
- Prévic, F. (2000). Neuropsychological guidelines for aircraft control stations.
- Rani, P., Sarkar, N., Smith, C., & Kirby, L. (2004). Anxiety detecting robotic system-towards implicit human-robot collaboration.
- Rani, P., Sims, J., Brackin, R., & Sarkar, N. (2002). Online stress detection using psychophysiological signals for implicit human-robot cooperation.

- Rigas, G., Goletsis, Y., Bougia, P., & Fotiadis, D. I. (2011). Towards Driver's State Recognition on Real Driving Conditions.
- Tango, F., & Botta, M. (2009). Evaluation of Distraction in a Driver-Vehicle-Environment Framework: An Application of Different Data-Mining Techniques.
- Tango, F., Botta, B., Minin, L., & Montanari, R. (2010). Non-intrusive Detection of Driver Distraction using Machine Learning Algorithms.
- Thomas, L., & Wickens, C. (2004). Eye-tracking and individual differences in off-normal event detection when flying with a synthetic vision system display.
- Tracy, J., Mohamed, F., Faro, S., Tiver, R., Pinus, A., Bloomer, C., et al. (2000). The effect of automatic arousal on attentional focus.
- Tsai, Y.-F., Viire, E., Strychacz, C., Chase, B., & Jung, T.-P. (2007). Task Performance and Eye Activity : Predicting Behavior Relating to Cognitive Workload.
- Van Eslande, P. (2001). Erreur de conduite et besoin d'aide : une approche accidentologique.
- Van Eslande, P., Alberton, L., Nachtergaële, C., & Blancher, G. (1997). Scénarios types de production de l'"erreur humaine" dans l'accident de la route. Rapport technique 218, Inrets.
- Wanner, J., & Wanner, N. (1999). Opérateur et sécurité. Rapport technique, 5/2.
- Weltman, G., & Egstrom, G. (1966). Perceptual narrowing in novice divers.
- Wickens. (2005). Attentional tunneling and task management.
- Wickens, C. (2005). *Attentionnal Tunneling and Task Management*.
- Wiegmann, D., & Shappell, S. (2003). A human error approach to aviation accident analysis : The human factors analysis and classification system.
- Zenou, E. (n.d.). cours sur les réseaux de neurones.