

---

# Travaux pratiques d'automatique avancée

## Identification neuronale

---

### Partie A

#### Modélisation statique

---

Dans cette partie, on cherche à approximer une fonction réelle bruitée à l'aide d'un réseau de neurones. Les mesures sont sauvegardées dans le fichier *approx.mat* ( $x$  mesures de l'entrée,  $y$  mesures de la sortie,  $y\_val$  second ensemble de mesures de la sortie).

Afin d'aborder rapidement l'utilisation de la toolbox *nnsysid*, un fichier *exemple.m* a été préparé.

*A.1. Ouvrir le fichier d'exemple et étudier son fonctionnement (les différentes fonctions utilisées sont détaillées dans l'annexe).*

*A.2. Étudier les influences du choix de l'algorithme et du nombre d'itérations sur l'apprentissage.*

On dispose de deux jeux de mesures de la sortie  $y$  et  $y\_val$ .

*A.3. Proposer une méthode permettant de déterminer un nombre de neurones cachés suffisant et évitant le sur-apprentissage. Tracer la courbe de validation correspondante.*

### Partie B

#### Modélisation dynamique

---

Dans cette partie, on veut obtenir un modèle paramétrique d'un système dynamique non linéaire : l'oscillateur de Van Der Pol. L'évolution de cet oscillateur est décrit par l'équation différentielle suivante :

$$\frac{d^2x(t)}{dt^2} + (3kx^2 - 2z\omega_0)\frac{dx(t)}{dt} + \omega_0^2x(t) = u(t) \quad (1)$$

Lorsque  $k \neq 0$ , ce système dissipatif possède une dynamique régulière caractérisée par un attracteur en forme de cycle limite. Le fichier simulink *vanderpol.mdl* permet de simuler le fonctionnement de cet oscillateur.

*B.1. Mettre en évidence les non-linéarités de cet oscillateur.*

*B.2. Réaliser un modèle ARX de l'oscillateur pour des commandes de faibles amplitudes. Mettre en évidence les limitations de ce modèle.*

*B.3. Réaliser un modèle neuronal (NARX) de l'oscillateur à l'aide des commandes `nнарx`, `ninvalid`, `nnsimul`.*

*B.4. Ajouter un bruit de boucle et recalculer un modèle NARX. Calculer la variance de l'erreur de prédiction. Conclusion ?*