



CYCLE 5

## RESOLUTION DE PROBLEMES PAR UTILISATION DE L'INGENIERIE NUMERIQUE OU L'APPRENTISSAGE AUTOMATISE

TP2 - PSI

### TP 2.1

### CONTROL X – RESEAUX DE NEURONES

*L'objectif de ce TP est de piloter le Control'X par un correcteur de type réseau de neurones.*

## 1 MISE EN SITUATION

Le système industriel duquel est extrait Control'X est un robot portique 3 axes Lexium Max R du constructeur Schneider Electric. Ce robot portique est constitué :

- d'un axe portique double Lexium MAX S assurant un déplacement selon la direction X
- d'un axe portique double Lexium MAX H assurant un déplacement selon la direction Y.
- d'un axe Cantilever Lexium CAS 4 ou Lexium CAS 3 assurant un déplacement selon la direction Z.

L'application choisie pour contextualiser Control'X et celle du « pick and place » dans le domaine du placement de composants électroniques. Il s'agit d'un processus de précision consistant à positionner des composants électroniques sur des circuits imprimés.

Dans ce contexte d'utilisation, la particularité mécanique tient au fait que les efforts résistants extérieurs exercés sur l'axe sont nuls : le moteur sert uniquement à vaincre les efforts inertiels ainsi que les résistances passives. Le moteur est souvent en prise directe avec la poulie motrice ou, s'il y a un réducteur, le rapport de réduction est généralement faible.

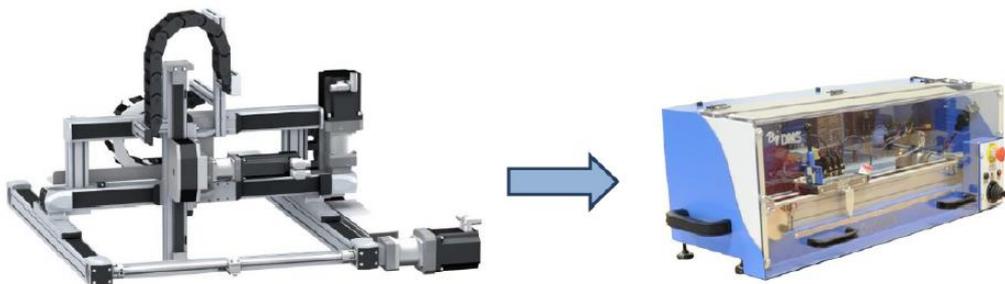


FIGURE 1 – Du Max R au Control'X

Le cahier des charges fixe les paramètres suivants pour une entrée échelon :

- pas de dépassement ;
- écart statique nul ;
- temps de réponse à 5 % de 0,2 s.

## 2 IDENTIFICATION DU MODELE DE LA BO PAR UN RESEAU DE NEURONES

### Activité 1

Prendre connaissance du schéma-blocs proposé dans le fichier `Activite_01_CX_Lin.slx`. Identifier ce que modélise le système d'ordre 1. Préciser ce qui ne semble pas avoir été modélisé.

Le système d'ordre 1 modélise le variateur, le moteur, le réducteur et le système poulie-courroie. Les frottements secs et visqueux ainsi que les saturations ne sont pas modélisés.

### Activité 2

Sachant que  $T_{eq} = 0,022$  s et que la commande est saturée à  $\pm 10$  V, proposer des entrées permettant d'entraîner un réseau de neurones permettant de modéliser la boucle ouverte.

### Activité 3

On se propose d'entraîner le modèle avec une succession aléatoire d'échelons. Proposer une durée minimale et maximale des échelons ainsi qu'une amplitude minimale et maximale. Pour générer une séquence d'échelons aléatoires, on utilisera le fichier `Activite_03_GenerationEchelons.m`.

Exemple d'utilisation :

```
[U_discret] = Activite_03_GenerationEchelons(0.01, 10, 1, 10, 5);
plot(U_discret)
```

- Sur le Control'X la tension de seuil permettant de modéliser les frottements secs est de l'ordre de 1,5 V. Expliquer pourquoi un entraînement entre  $-2$  V et  $2$  V est peu judicieux.
- Expliquer pourquoi il est judicieux que, pour l'entraînement, la position estimée en utilisant le modèle de la boucle ouverte du Control'X soit comprise entre  $-250$  mm et  $250$  mm.

Un entraînement sur des tensions trop souvent inférieures à la tension de seuil (en valeur absolue) ne sera pas pertinent car on sera dans une plage de travail où le modèle est peu fidèle à la réalité.

De même, si les échelons générés provoquent des déplacements en dehors de la plage de déplacement du Control'X, on s'entraînera sur une zone qui n'est pas la zone d'utilisation.

Enfin, il faudra veiller à ce que les déplacements couvrent toute la zone de travail afin que le réseau de neurones puisse s'entraîner sur l'ensemble de la plage accessible.

### Activité 4

En utilisant le fichier `Activite_04_NN_CX.mlx` configurer le réseau de neurones en vue de son entraînement (choix des données d'entrée, choix des paramètres du réseau).

## 3 CORRECTION PAR RESEAU DE NEURONES

Il s'agit en préambule d'avoir lu le cours sur les réseaux de type NARX et la section concernant le contrôle des systèmes dynamiques.

### Activité 5

Faire une synthèse des Modèles Référence Adaptative Controller (MRAC).

### 3.1 Pré-entraînement

#### Activité 6

Proposer un modèle idéal à identifier afin de satisfaire les exigences du cahier des charges ? On pourra se demander quel est le comportement souhaité, en régime permanent et en régime transitoire, quel ordre de fonction permet de vérifier ceci, etc.

La réponse temporelle d'un système d'ordre 1 ou d'ordre 2 avec  $z > 1$  permet de valider l'exigence de non dépassement. L'exigence de précision peut être atteinte quel que soit l'ordre avec une amplification statique de 1 ; enfin la rapidité n'influence pas le choix. Les auteurs proposent de choisir plutôt l'ordre 2 par la non-discontinuité de sa dérivée en  $0^+$  afin de rendre plus souple le comportement du correcteur par réseau de neurones. Ainsi la fonction de transfert objectif à suivre est :

$$H_{obj}(p) = \frac{1}{1 + 2 \times 1 \frac{p}{30} + \frac{p^2}{30^2}} \quad (1)$$

Une clé de la bonne modélisation par réseau de neurones réside dans les données.

#### Activité 7

Proposer et générer des données permettant un bon entraînement du réseau de neurones correcteur. Préciser les amplitudes de variation des caractéristiques de ce signal : amplitude, durée, nombres et échantillonnage.

Le système sera sollicité essentiellement en réponse à des échelons de positions. On se fixe une période d'échantillonnage de  $T_e = 0,01$  s. La durée de ces échelons doit être comprise entre 0,1 s (choix arbitraire) et 1,5 s (choix permettant de mettre en évidence le régime permanent). Les amplitudes de sortie correspondent aux possibilités du système. Ici, nous souhaitons entraîner le réseau de neurones sur une amplitude  $[-100; 100]$  mm. Le nombre de 20 000 points permet d'obtenir globalement une petite centaine d'échelons, ce qui semble être un nombre correct vis-à-vis de entraînement qui suit.

L'identification du système par un réseau de neurone a été effectué lors de l'activité 4. Par conséquent, on peut passer à l'étape suivante : l'entraînement du contrôleur.

### 3.2 Entraînement

#### Activité 8

À l'aide du fichier `Activite_08_Train_Controller.mlx`, créer le réseau de neurone du contrôleur. Pour cela :

- renseigner les caractéristiques des données d'entrée ;
- configurer la configuration du système attendu en boucle fermée ;
- configurer les caractéristiques des échelons d'entrée ;
- configurer les caractéristiques du réseau de neurone associé au contrôleur.

Lancer alors l'entraînement du réseau avec le contrôleur.

La structure proposée pour le réseau de neurone donnée figure 2.

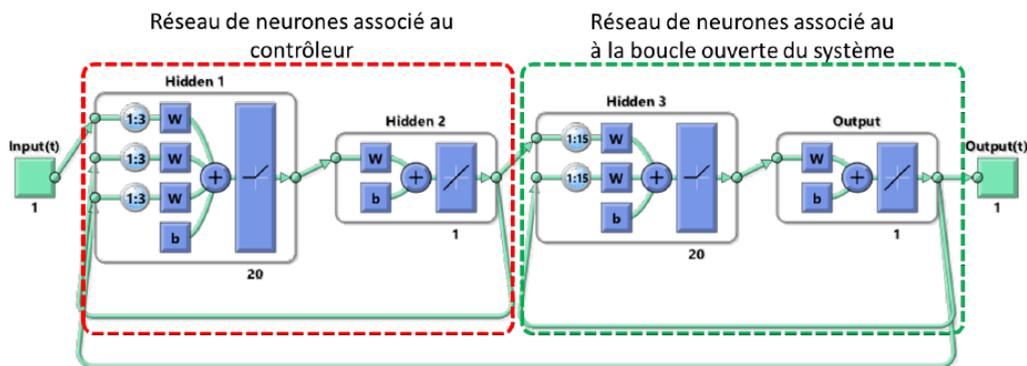
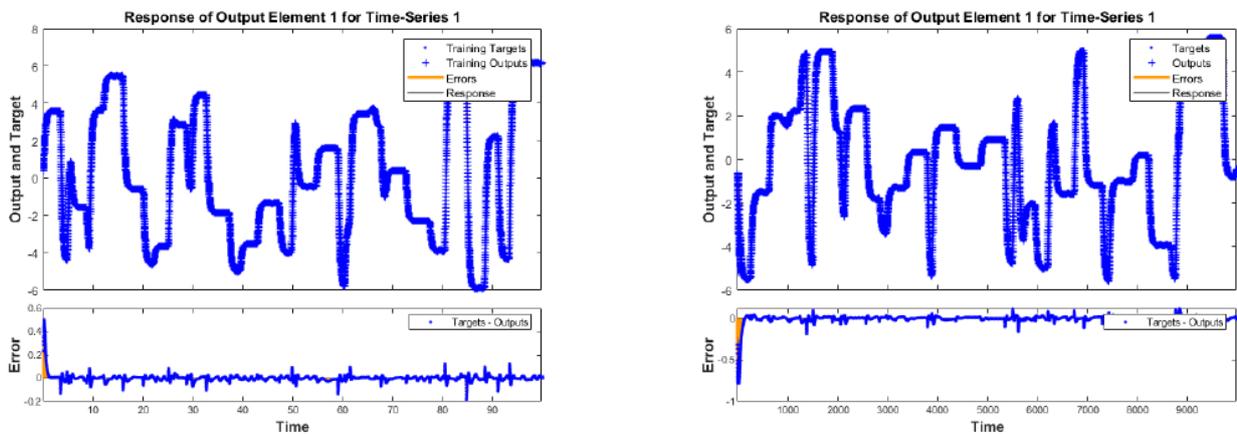


FIGURE 2 – Structure du MRAC



(a) Résultat de l'entraînement du contrôleur

(b) Simulation du réseau avec des données de test

FIGURE 3 – Entraînement et inférence du MRAC

### 3.3 Implémentation sur le système

#### Activité 9

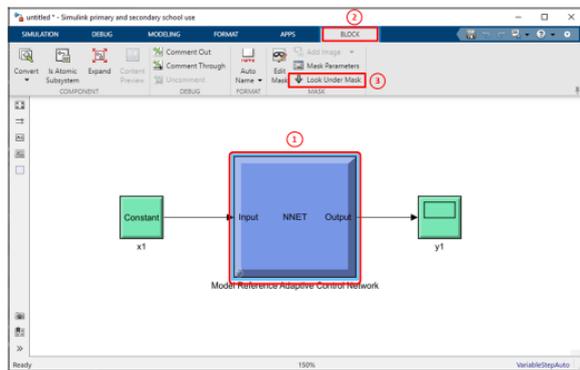
En utilisant les consignes suivantes, générer le fichier Simulink permettant de piloter le système par un réseau de neurones. On utilisera le fichier `Activite_09_PilotageANN_Vide.slx`.

Pour implémenter le réseau de neurone associé au contrôleur, nous allons copier les blocs Simulink associés au réseau de neurone du contrôleur et les coller sur une feuille permettant de piloter le système. Commençons par

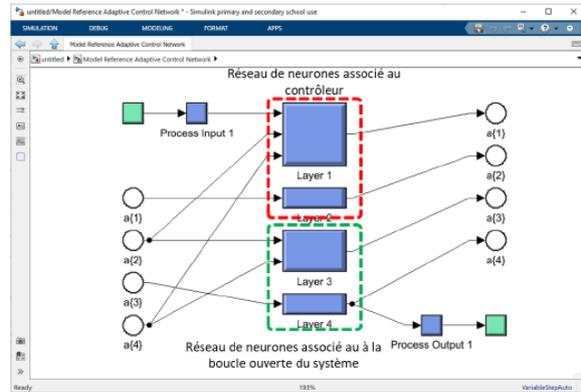
générer le bloc Simulink associé au MRAC en utilisant la commande `gensim(mrac_net)` et affichons ensuite la structure du réseau (Figure 4). Pour cela :

1. cliquer sur bloc bleu (MRACN) ;
2. cliquer sur l'onglet « Block » ;
3. cliquer sur « Look Under Mask » (raccourci clavier : Ctrl+u).

Copier alors les couches 1 et 2 dans le MRAC associé au pilotage du Control'X.



(a) Bloc Simulink du MRAC



(b) Structure Simulink du réseau de neurones

FIGURE 4 – Récupération du réseau de neurones du contrôleur

La figure 5 (il faut avouer qu'elle est super belle) illustre la zone où copier le réseau de neurones dans le fichier `Activite_09_PilotageANN_Vide.slx`.

**Activité 10**

Implémenter sur le système réel en utilisant le bouton « Run in Real Time » et évaluer les écarts.

Sur le premier échelon :

- écart statique souhaité : 0 mm. Écart statique mesuré : 8 mm.
- temps de réponse souhaité 0,2s. Temps de réponse mesuré : < 0,2s.

On constate qu'au début du régime transitoire les allures de la courbe cible et de la mesure évoluent de façon similaire. La commande du contrôleur ANN n'est ensuite plus satisfaisante car la tension délivrée est inférieure à la tension de seuil du système. En effet, ce phénomène n'a pas été correctement modélisé. Il est donc normal que la commande soit moins satisfaisante.

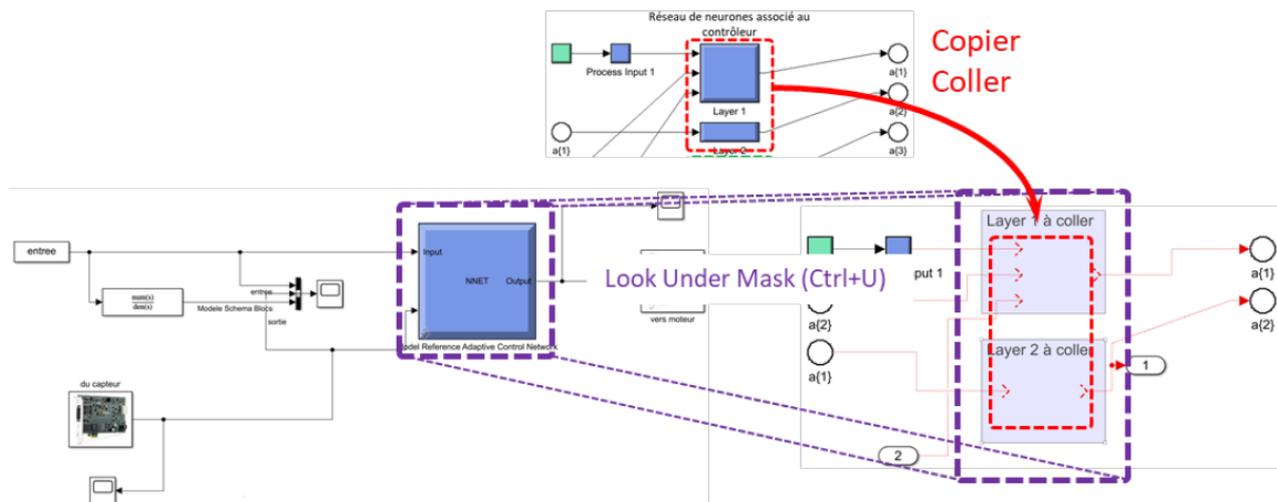


FIGURE 5 – Copie du contrôleur

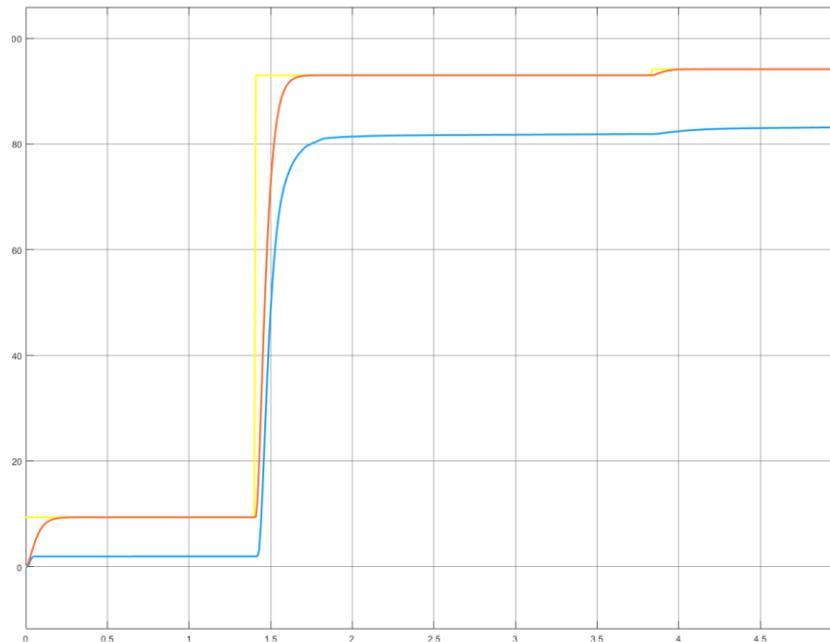


Figure 6 – Résultat donné par le système contrôlé par un réseau de neurones (Jaune : consigne, Rouge : cible, Bleu : mesure sur le Control'X)

Sur le deuxième échelon, la variation de consigne est plus grande, ce qui donne une impression plus satisfaisante. Cependant les mêmes phénomènes sont visibles : régime transitoire satisfaisant et régime permanent qui ne valide pas le cahier de charges.

Évaluation des écarts par rapport au cahier des charges :

- écart statique souhaité : 0 mm. Écart statique mesuré : 8 mm.
- temps de réponse souhaité 0,2s. Temps de réponse mesuré : < 0,2s.

On observe un écart, notamment sur la valeur de l'écart statique. Cela peut s'expliquer par une « mauvaise » identification du comportement du système lors de l'entraînement du réseau de neurones. Il est en effet difficile d'obtenir des résultats satisfaisants avec une identification sur un modèle non linéaire. Une mauvaise modélisation de la tension de seuil peut donc expliquer que le contrôleur ne parvienne pas à supprimer l'écart statique.

Il faudrait donc déterminer des paramètres de chacun des réseaux de neurones permettant d'obtenir de meilleurs résultats.

## 4 SYNTHÈSE

### Activité 11

Proposer une synthèse sur les avantages et les inconvénients sur la correction mise en œuvre dans ce TP par rapport à l'utilisation d'un correcteur PI(D).

Avantages :

- Relative facilité de mise en œuvre du processus de corrections.
- Possibilité de réglage sur PC avant implémentation sur cible.
- Pas/Peu d'essais nécessaires.
- Possibilité d'entraîner le correcteur avec une fonction de transfert directement choisie par le concepteur.

Inconvénients :

- Nécessité de disposer d'un modèle relativement fiable.
- Difficulté de choisir les caractéristiques du réseau de neurones.
- Difficulté de choisir le type d'entrées.
- Processus d'entraînement peu répétable et qui peut échouer.