



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.

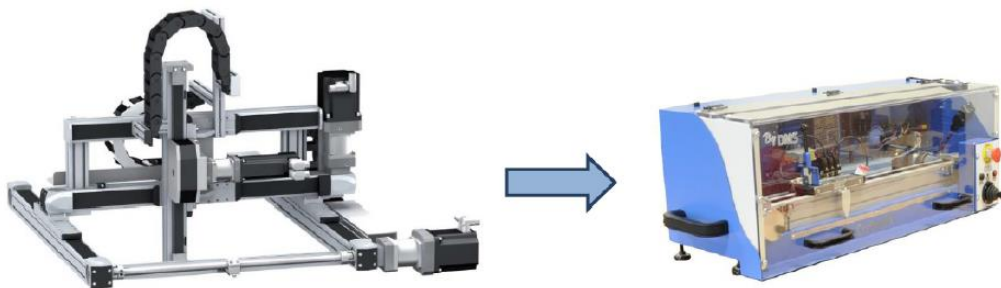


FIGURE 1 – Du Max R au Control'X

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.

Activité 2

Sachant que $T_{eq} = 0,022$ s et que la commande est saturée à ± 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.

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 *Model Reference Adaptive 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.

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.

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 le 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.

3.3 Implémentation sur le système

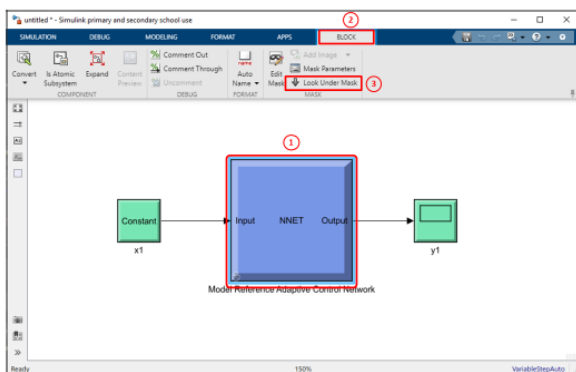
Activité 9

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

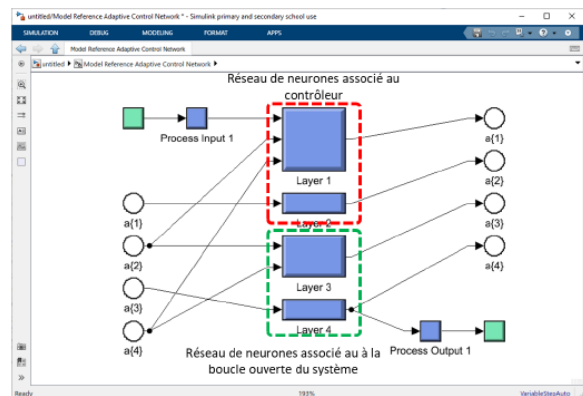
Pour implémenter le réseau de neurone associé au contrôleur, nous allons copier les blocs Simulink associés au réseau de neurone du contrôleur et les coller sur une feuille permettant de piloter le système. Commençons par générer le bloc Simulink associé au MRAC en utilisant la commande `gensim(mrac_net)` 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.

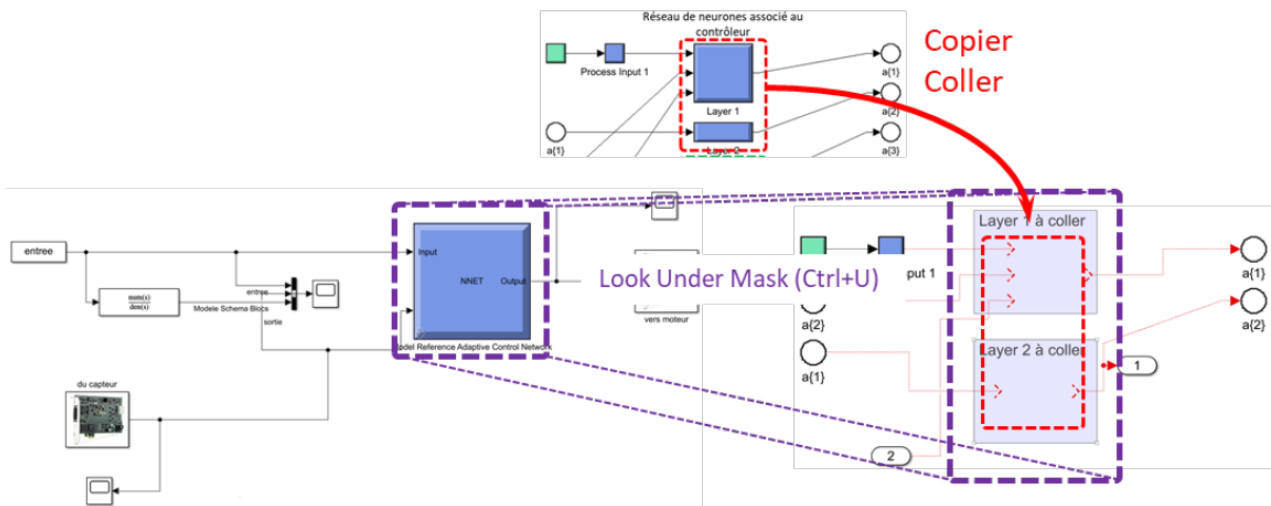


FIGURE 5 – Copie du contrôleur

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).