Modélisation système avec Simulink®

# Introduction

## Présentation de Matlab® Simulink®

Matlab® est un langage de programmation dédié aux ingénieurs et aux scientifiques. Il dispose de nombreuses similarités avec Python : Il ne se compile pas et le typage est faible : par exemple il n’est pas nécessaire de déclarer le nom ou le type (booléen, entier, flottant…) des variables avant de les utiliser. L’interpréteur se charge de « deviner » vos intentions en fonction du type de calcul. Matlab est un logiciel propriétaire. Il dispose d’un environnement de programmation intégré : éditeur de texte, interpréteur bibliothèques et éléments de visualisation sont dans un même package.

Matlab intègre aussi de nombreux outils dédiés aux problèmes scientifiques et d’ingénierie que l’on ne retrouve pas dans Python. Par exemple Simulink est un outil permettant de mettre en place des modèles par schéma blocs avec une interface graphique et un solveur intégré. Ainsi l’utilisateur dispose les blocs dans une fenêtre, les relies, définit les équations de leurs fonctions de transfert, les entrées et les sorties souhaitées. Ensuite il peut résoudre et observer les résultats de sa modélisation. Ainsi Simulink est très largement utilisé dans l’industrie (aérospatiale, environnement, défense etc…).

## Objectifs du TP

Dans ce TP nous allons apprendre à se servir de ce logiciel de simulation autour d’un exemple concret, la modélisation d’un axe de positionnement robotique.

## Objet du TP

Nous étudierons l’axe numérique Didalab. Ce système permet de positionner ou de réguler la vitesse d’un chariot motorisé par une machine à courant continu. Il pourrait par exemple servir à positionner la tête d’impression d’une imprimante grande dimensions, la broche d’une machine-outil (avec un moteur plus puissant) ou encore la tête d’une imprimante 3D. Cet axe permet de faire des expériences dans un environnement de prototypage contrôlé et de valider des choix techniques avant de réaliser une machine plus complète. Idéalement cet axe doit être rapide et précis, des critères quantifiés étant disponible dans le cahier des charges de la machine que l’on souhaite réaliser.

## Déroulement du TP :

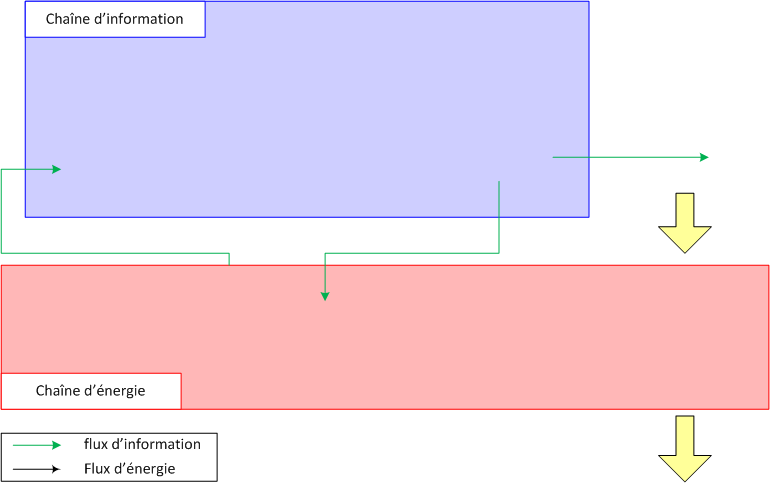
1. Préparation
   1. Analyse du système, de ses constituants, du schéma bloc et des équations.
   2. Prise en main de Simulink
2. Modélisation de l’axe numérique en **boucle ouverte** et interface de **courant**.
3. Modélisation de l’axe numérique en **boucle fermée de vitesse** et interface de **courant**.
4. Modélisation de l’axe numérique en **boucle fermée de position** et interface de **courant**.
5. Modélisation de l’axe numérique en **boucle fermée de position avec retour tachymétrique** et interface de **courant**.
6. Modélisation de l’axe numérique en **boucle ouverte** et interface de **tension**.

# Préparation

## Analyse du système

Lancer le logiciel Axnum et choisir le mode déconnecté. Ensuite aller dans « choisir/Unité/Sortie régulateur » et choisir unités IP. Cela permet de s’affranchir de la conversion en unité interne au microprocesseur et simplifie le modèle, et l’adaptation des signaux.  Ouvrir la documentation technique PDF qui contient les informations sur le système, pages 24 à 35.

### QA1 Compléter la chaine fonctionnelle :



### QA2 Schéma bloc du système Boucle ouvert et Interface de courant.

Dans le logiciel Axnum sélectionner « Choisir/Mode de commande/Boucle ouverte et Choisir/Interface de puissance/Commande Courant. » A l’aide de la chaine, de cette vue et de la documentation identifier les entrées sorties et compléter le schéma bloc pour ce mode de fonctionnement. Il devra faire apparaitre les variables suivantes : Couple réducteur **Cr**, Force motrice **F**, Courant commande **Im**, Couple moteur **Cm**, Accélération **Mx’’**, et Vitesse **Mx’.**

C:\Users\canar\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Axnum (1).png

Les gains ou fonctions des blocs sont à spécifier

### QA3 Schéma bloc du système Boucle fermée vitesse et Interface de courant.

Dans le logiciel Axnum sélectionner « Choisir/Mode de commande/Boucle fermée vitesse et Choisir/Interface de puissance/Commande Courant. ». On n’utilisera que la correction proportionnelle soit Sri et Srd=0 ce qui simplifie le modèle.

Il faudra ajouter le correcteur, le transducteur (gain K2), le capteur, les variable écart , correction proportionnelle **Srp** et le gain de conversion de la commande courant tel que 2 incréments = 1mA.

### QB1 Expérimentation avec Simulink

Créer un répertoire SimulationAxnum dans un répertoire perso sauvegardé. Lancer MATLAB. Placer l’adresse du répertoire dans la barre d’adresse de Matlab. Tester le mode calculatrice en tapant directement un calcul dans la barre d’exécution en bas de l’écran. Faite click droit dans la fenètre de gauche « New Script » pour créer un fichier de commandes Matlab dans le repertoire de travail. Ce fichier permettra de faire des calculs, des programmes et de définir des variable ce qui sera utilse pour Simulink. Par exemple taper « A=3+pi » dans le script et l’exécuter (onglet lecture). La variable A apparait dans le « Workspace ». Désormais presser le bouton Simulink puis choisir un « Blank model » pour créer votre premier schéma bloc à simuler. En pressant le bouton de couleur on peut choisir ses blocs. Dans les rubriques « commonly » (blocs courants) « source » (entrées) « sinks » (sorties) « continuous » (intégrale, dérivé, fonction de transfert) et « Math » (polynomes, trigonométrie, seuils) vous trouverez l’essentiel des blocs de base dont vous aurez besoin.

Placer un signal échelon « source » et paramétrez le à la valeur 3, un bloc gain dont vous donnez la valeur 7 et un bloc scope « sink ». Reliez les et exécutez (bouton lecture). Commenter le résultat.

Remplacer la valeur 3 par la variable A et observer le nouveau résultat. En modifiant la source faite démarrer le « Step » dés le début (t=0).

Taper B=2 dans le script, le ré-exécuter, remplacer A par B, et ajouter un bloc intégrateur « Continuous/Integrator » dans la boucle. Commenter.

### Mettez en œuvre vos compétences nouvellement acquise pour faire ces simulations :

Modélisation de l’axe numérique en **boucle ouverte** et interface de **courant**.

Modélisation de l’axe numérique en **boucle fermée de vitesse** et interface de **courant**.

Modélisation de l’axe numérique en **boucle fermée de position** et interface de **courant**.

Modélisation de l’axe numérique en **boucle fermée de position avec retour tachymétrique** et interface de **courant**.

Modélisation de l’axe numérique en **boucle ouverte** et interface de **tension**.

Par « copier-coller » on pourra largement réutiliser le travail déjà effectué.