



# LYCÉE LA MARTINIÈRE MONPLAISIR LYON

SCIENCES INDUSTRIELLES POUR L'INGÉNIEUR

CLASSE PRÉPARATOIRE M.P.S.I.

ANNÉE 2017 - 2018

---

C6 : ANALYSE FRÉQUENTIELLE DES SYSTÈMES ASSERVIS

## TP 4 - Modélisation et simulation des systèmes causaux avec Matlab Simulink(C6)

---

### Compétences

- **Analyser :**
    - Apprécier la pertinence et la validité des résultats : grandeurs utilisées;
    - Appréhender les analyses fonctionnelles et structurelles : Architectures fonctionnelle et structurelle : chaîne directe, système asservi, commande
    - Caractériser les écarts
  - **Modéliser :**
    - Identifier et caractériser les grandeurs physiques : caractéristiques fréquentielles
    - Systèmes linéaires continus et invariants
    - Signaux canoniques d'entrée : signaux sinusoïdaux
    - Schémas blocs, fonctions de transferts
  - **Résoudre :** Procéder à la mise en oeuvre d'une démarche de résolution numérique
    - Réponse fréquentielle : systèmes du 1<sup>er</sup> et 2<sup>ème</sup> ordre, intégrateur.
    - Paramètre de résolution numérique
    - Grandeurs simulées
  - **Communiquer :** Rechercher et traiter des informations techniques
  - **Expérimenter :**
    - S'approprier le fonctionnement d'un système pluritechnologique : paramètres influents
    - Proposer et justifier un protocole expérimental : modèles de comportements d'un systèmes
    - Mettre en oeuvre un protocole expérimental : Identification temporelle et fréquentielle d'un modèle de comportement.
- paramètres influents

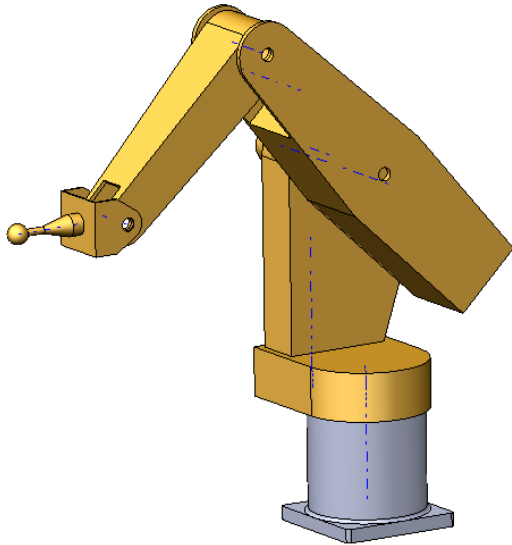
### 1 Présentation du système réel

#### a) Le robot Erric3

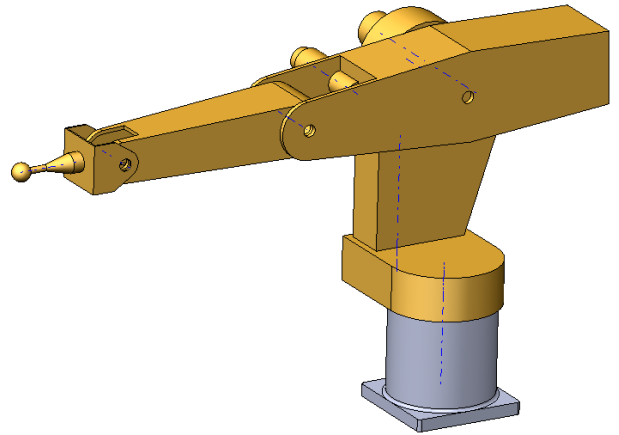
Le Robot Erric3 est un robot qui présente un caractère anthropomorphique. Il est constitué de 5 axes asservis en position.

On s'intéresse ici uniquement à l'asservissement autour de l'axe de lacet.

On considérera deux configurations

**Configuration 1 : bras replié**

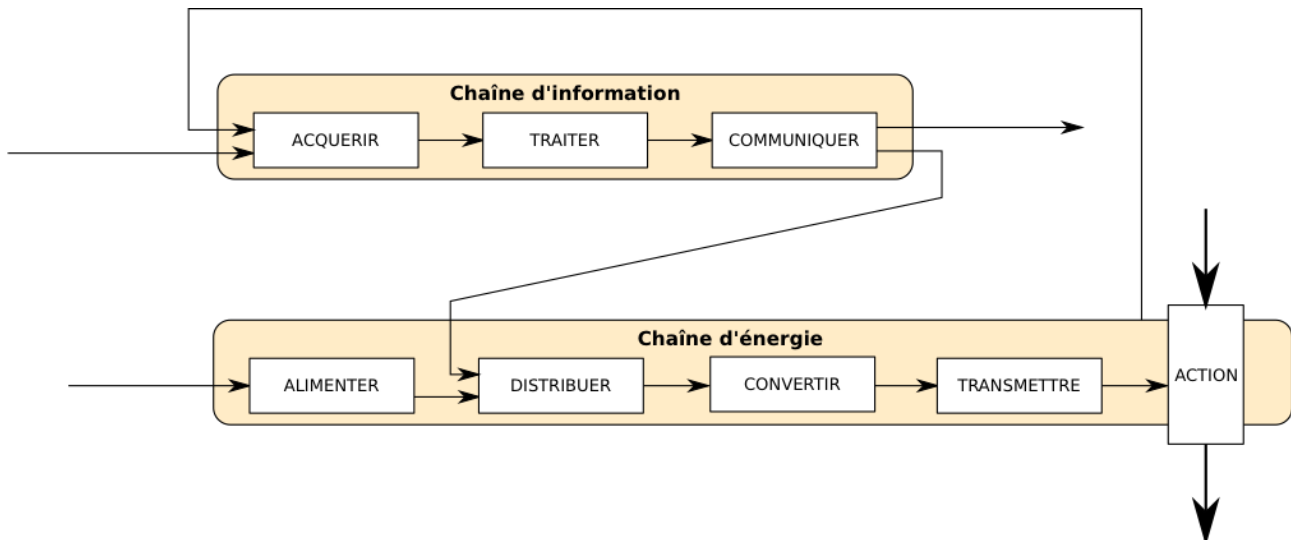
$Epaule = -39^\circ$ ;  $Lacet = 0^\circ$ ;  $Coude = 00^\circ$ ,  
 $Poignet = -51^\circ$

**Configuration 2 : bras déplié**

$Epaule = 0^\circ$ ;  $Lacet = 0^\circ$ ;  $Coude = -90^\circ$ ,  $Poignet = 0^\circ$

### b) Analyse structurelle du robot

**Q 1 : Compléter la chaîne fonctionnelle décrivant la chaîne cinématique "axe de lacet".**

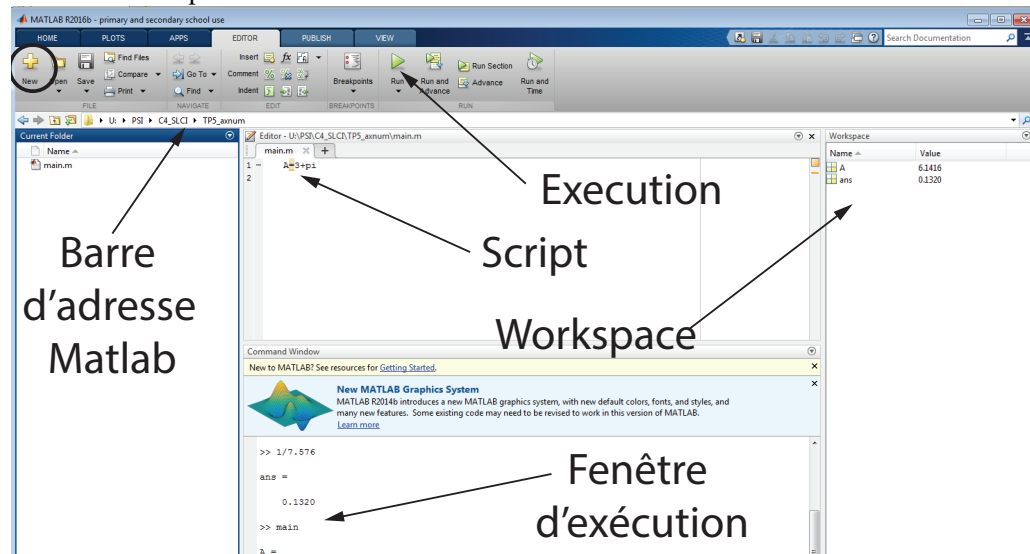


## 2 Mise en oeuvre de la simulation

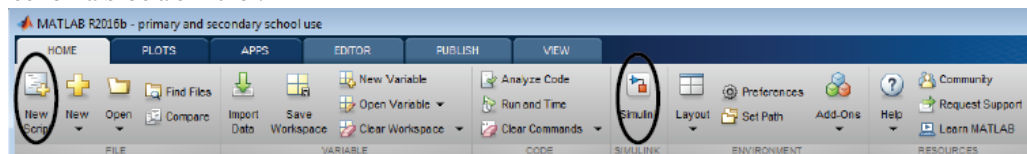
### a) Prise en main de Matlab Simulink

- Créer un répertoire **TP4** à la racine de *U* : \Perso. Lancer MATLAB (dernière version). Placer l'adresse du répertoire créé dans la barre d'adresse de Matlab.
- Copier le contenu du dossier **TP4** situé dans le dossier **transfert>MPSI2** dans le dossier que vous venez de créer.
- Tester le mode calculatrice en tapant un calcul dans la fenêtre d'exécution en bas de l'écran.
- Faites click gauche dans la fenêtre de gauche « New Script » pour créer un fichier de commandes Matlab dans le répertoire de travail. Ce fichier permettra de faire des calculs, des programmes et de définir des variables ce qui sera utile pour Simulink. Par exemple taper «  $A=3+\pi$  » dans le script et l'exécuter (onglet lecture). La variable A

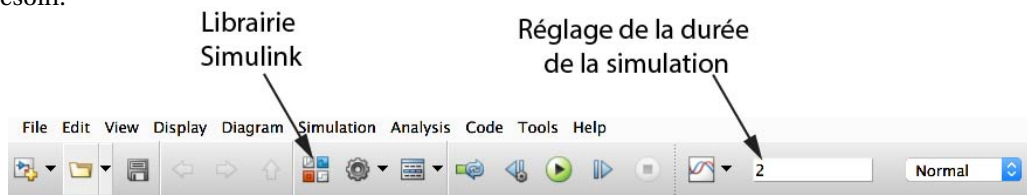
apparaît dans le « Workspace ».



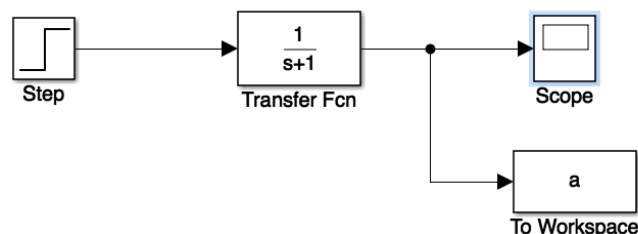
- Désormais presser le bouton Simulink dans l'onglet "Home" puis choisir un « Blank model » pour créer votre premier schéma bloc à simuler.



- En pressant le bouton de couleur on peut choisir ses blocs (bibliothèque Simulink). Dans les rubriques « commonly used block » (blocs courants) « source » (entrées) « sinks » (sorties) « continuous » (intégrale, dérivé, fonction de transfert) et « Math » (polynômes, trigonométrie, seuils) vous trouverez l'essentiel des blocs de base dont vous aurez besoin.



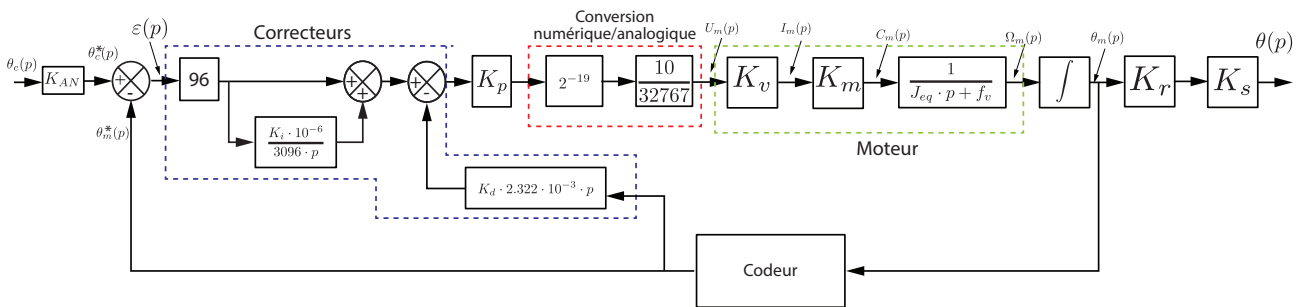
- Glisser un signal échelon « source » dans le nouveau modèle et paramétrez-le à la valeur finale 3, ajoutez un bloc **"transfert function"** de manière à modéliser un premier ordre avec une constante de temps égale à 1. Ajouter un bloc **Scope** et un bloc **to workspace** (sink). Reliez-les et exécutez (bouton lecture). Ouvrir le scope et commenter le résultat. Observer dans le workspace les variables créées.



- Taper *a*, puis *a · Time* et *a.Data* dans la fenêtre d'exécution.

### b) Construction du modèle du robot Ericc3

On donne le schéma bloc global du système :



- La structure de l'asservissement et numérique. Les grandeurs numériques sont représentées avec un exposant \*.
- L'angle de consigne de lacet se note :  $\theta_c(p)$ .
- L'angle à la sortie de l'arbre moteur se note  $\theta_m(p)$ .
- Le système comporte un correcteur PID (Proportionnel Intégral Dérivé). Ici n'est représenté que le correcteur Proportionnel (de gain  $K_p$ ) et Intégrale (de gain  $K_i$ ). Dans l'étude on n'étudiera que l'influence de  $K_p$ . Ainsi on prendra  $K_i = 0$ .
- Après une conversion numérique analogique, on modélise le moteur avec un **variateur** (de constante  $K_v$ ), une **constante de couple** ( $K_m$ ).
- On note  $U_m(p)$  la tension du moteur,  $I_m(p)$  l'intensité dans le circuit moteur et  $C_m(p)$  le couple délivré par le moteur.
- L'inertie ramené à l'arbre moteur correspondant à la rotation du lacet du robot est donnée par  $J_{eq}$ . Le frottement visqueux est modélisé par le coefficient  $f_v$ .
- On note  $\Omega_m(p)$  la vitesse de rotation (en  $rad/s$ ) de l'arbre moteur.
- Le **système de réduction** de vitesse de fonction de transfert  $K_r$  est composé
  - d'un réducteur poulie-courroie;
  - d'un réducteur Harmonic Drive de rapport de réduction 1/100



## Q 2 : Déterminer le rapport de réduction $K_r$ du système.

- Le **capteur de position** mesure directement l'angle à la sortie du moteur.

La mesure de position s'effectue par un codeur incrémental qui délivre une information du déplacement angulaire du disque sous forme de train d'impulsions. Le nombre d'impulsions décompté à partir d'une origine permet d'avoir accès à la position angulaire, tandis que la fréquence du signal renseigne sur la vitesse du disque. Il est constitué d'une ou plusieurs voies comportant les zones opaques et transparentes régulièrement espacées. Le nombre de zones transparentes définit la résolution du capteur. Le codeur de l'axe de lacet fournit 2000 impulsions par tour.



**Q 3 : Déterminer l'expression du gain du capteur noté  $K_c$ .**

• **Gain d'adaptation**

**Q 4 : Déterminer l'expression du gain d'adaptation  $K_S$  pour garantir :**

$$\varepsilon(p) = 0 \Leftrightarrow \theta_c(p) = \theta(p)$$

**Q 5 : Compléter le schéma bloc "TP\_modele\_matlab\_eleve.slx" pour modéliser le système asservi en boucle fermée.**

- On pourra utiliser le bloc "To Workspace" en nommant la variable de sortie "S1".
- On imposera un échelon de  $30^\circ$ .

### 3 Analyse temporelle des performances du robot

#### a) Comparaison entre les performances réelles et simulées

L'étude portera sur la **configuration 1** (bras en partie replié).

**Q 6 : Importation des données :**

1. Ouvrir le fichier `data_modele_eleve.m` et observer son contenu.
2. Exécuter et observer le message d'erreur.
3. Compléter le programme avec les informations manquantes issues de la partie précédente.

**Q 7 : Exécuter la simulation sur une durée de 2.5s et observer le résultat en double cliquant sur le Scope.**

Pour comparer les données entre le modèle réel et simulé on pourra utiliser et exécuter le fichier `Compare_modele1.m`. Ce fichier permet de traiter les données expérimentales d'un essai réalisé sur le système réel avec  $K_p = 10^6$  (fichier `essai_pos1_kp1000000.csv`). **Q 8 : A l'aide de la courbe qui s'affiche, analyser les écarts.**

**Q 9 : Modifier la consigne pour obtenir un créneau en entrée comme sur l'essai expérimental (Le créneau dure 1,1s).**

#### b) Amélioration du modèle

##### 1. Prise en compte de la saturation

Les données expérimentales de la tension moteur et de l'intensité moteur sont stockées respectivement dans les variables  $Im$  et  $Um$ .

**Q 10 : En vous inspirant des commandes du fichier `compare_modele1.m` tracer ces grandeurs en fonction du temps.**

**Q 11 : Que pouvons nous dire de la saturation? Proposer une modification du modèle tenant compte de ce phénomène.**

##### 2. Prise en compte d'un couple de perturbation

**Q 12 : Proposer une modification du schéma bloc pour tenir compte d'un couple de frottement  $C_r(p)$ . On pourra tester la simulation avec des valeurs comprises entre  $0,01 N \cdot m$  et  $0,02 N \cdot m$ .**

### c) Influences des paramètres du correcteur

**Q 13 :** Taper "clear all" dans la console Matlab. Relancer le script `data_model_eleve.m`.

**Q 14 :** Réaliser 3 simulations avec 3 valeurs de  $K_p$  différentes ( $\{10^6; 10^5; 5 \times 10^4\}$ ) en modifiant le nom de la variable dans le bloc "workspace" (respectivement (S1, S2 et S3)).

Le fichier "`compare_model_kp.m`" à ouvrir dans Matlab. permet de comparer les trois simulations et trois essais expérimentaux avec les 3 valeurs de  $K_p$  données précédemment.

**Q 15 :** Ouvrir le fichier et exécuter-le.

**Q 16 :** Remplir le tableau suivant.

Correcteur	Stabilité		Rapidité	Précision
$K_p$	Nombre de dépassement	Valeur de 1 <sup>er</sup> dépassement	Temps de réponse à 5%	Erreur statique $\varepsilon_s$
$5 \times 10^4$				
$10^5$				
$10^6$				

## 4 Influence de l'inertie

Une étude dynamique permet de montrer qu'entre la configuration 1 et la configuration 2  $J_{eq}$  est multiplié par 1,4.

**Q 17 :** Mettre en place deux simulations modélisant le comportement pour les configuration 1 et 2 (avec  $K_p = 10^6$ ) en modifiant le nom de la variable dans le bloc "workspace" (respectivement (S1, et S4)).

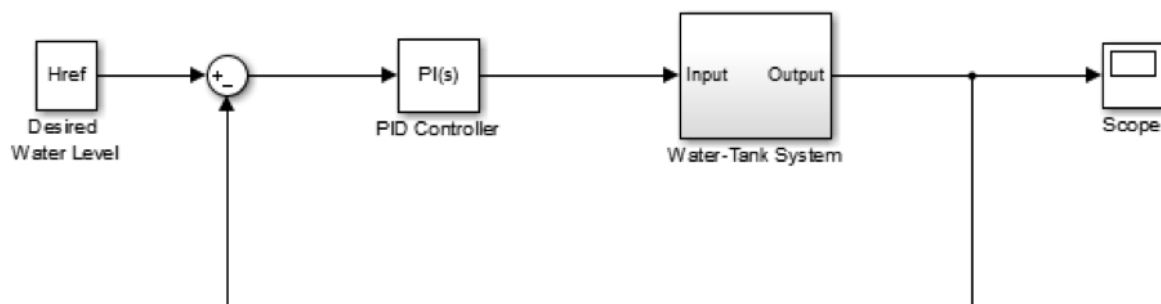
Le fichier "`compare_model_configuration.m`" à ouvrir dans Matlab. permet de comparer les deux simulations et trois essais expérimentaux pour les configuration 1 et 2.

**Q 18 :** Ouvrir le fichier et exécuter-le.

## 5 Analyse des performances fréquentielles du système

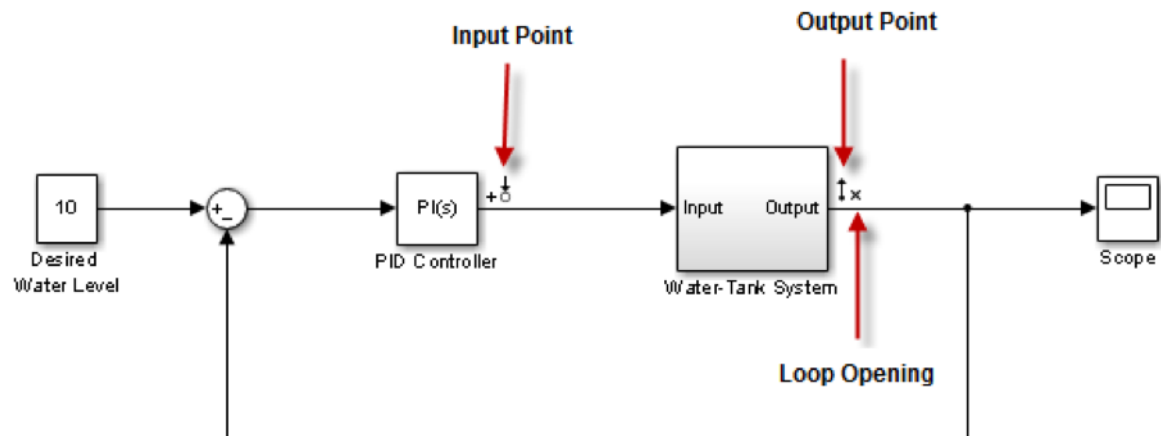
### a) Mise en oeuvre d'une étude fréquentielle en boucle fermée

Cet exemple montre comment utiliser le mode « Linear Analysis Tool » pour linéariser un modèle. Il faut pour cela préciser des points de prélèvements (entrée et sortie).



Copyright 2004-2012 The MathWorks, Inc.

1. Ouvrir "Linear Analysis Tool" pour le modèle.  
Dans le menu de Simulink sélectionner Analysis > Control Design > Linear Analysis.
2. Dans le menu de Simulink, définir la partie du modèle qui doit être linéarisée :
  - (a) Cliquer droit sur le signal de sortie du qui correspond à l'entrée du bloc à étudier. Sélectionner Linear Analysis Points > Input Perturbation.
  - (b) Cliquer droit sur le signal en sortie du " Water-Tank System " et sélectionner Linear Analysis Points > Open-loop Output.
3. Insérer ce point en « boucle ouverte » annule les effets du retour dans la linéarisation.
4. On voit alors apparaître sur le schema bloc les points créés.



5. Il suffit de cliquer sur  pour faire apparaître le diagramme.

**Q 19 : Vérifier cette analyse sur un système du 1<sup>er</sup> ordre.**

**b) Analyse fréquentielle par la simulation**

**Q 20 : Effectuer l'analyse fréquentielle en boucle fermée du système.**

**Q 21 : Effectuer l'analyse fréquentielle en boucle ouverte du système.**

**c) Comparaison entre les modèles réels et simulés**

Le fichier "**tracer\_bode\_exp.m**" à ouvrir dans Matlab. permet de tracer le diagramme de bode en boucle fermé obtenu expérimentalement en configuration 1.

**Q 22 : Ouvrir le fichier et exécuter-le.**