

TP SYSTEMES ASSERVIS : PRISE EN MAIN DU LOGICIEL *MATLAB SIMULINK*

Durée approximative: 30 minutes

| Programme - Compétences | | |
|-------------------------|-----------|---|
| A31 | ANALYSER | Architectures fonctionnelle et structurelle : - diagrammes de définition de blocs - chaîne directe - système asservi - commande |
| B24 | MODELISER | Systèmes linéaires continus et invariants: - Modélisation par équations différentielles - Calcul symbolique - fonction de transfert; gain, ordre |
| B25 | MODELISER | Signaux canoniques d'entrée: - échelon - rampe |
| C21 | RESOUDRE | Réponses temporelle: - systèmes du 1er ordre |
| C23 | RESOUDRE | Rapidité des SLCI: - temps de réponse à 5% |



OBJECTIFS DU TP :

- Prise en main du logiciel *Matlab Simulink*.
- Visualisation des résultats vus en cours et en TD concernant le 1^{er} ordre.
- Déterminer ses caractéristiques dans divers cas de sollicitations- tests.

1. PRESENTATION DU LOGICIEL

Le logiciel *MATLAB Simulink* est un logiciel de modélisation de systèmes multi-physique. Il permet au moyen d'une interface graphique de réaliser (entre autres) des schémas blocs dans le domaine de Laplace et de simuler les réponses temporelles et fréquentielles de circuits formés de blocs fonctionnels et soumis à un ou plusieurs signaux d'entrée canonique.

🌀 *MATLAB Simulink* est basé sur l'utilisation de schémas blocs. Le mode opératoire est le suivant :

- Composer le modèle de votre application et définir les paramètres liés à chaque bloc ;
- Placer les points d'entrée pour les signaux d'excitation et les points de sortie pour observer le comportement des variables qui vous intéressent ;
- Nommer l'ensemble des blocs et variables ;
- Lancer la simulation en définissant les variables calculées et le domaine temporel sur lequel doit porter la simulation ;
- Afficher les courbes obtenues par le calcul et interpréter les résultats.

2. ETUDE D'UN ELEMENT DU PREMIER ORDRE

Lancer *MATLAB* en double cliquant sur l'icône présente sur le bureau :




Lancer le module de calculs graphique de *MATLAB* nommé *Simulink* en cliquant sur l'onglet « *Simulink Library* » situé en haut de l'écran :



Dans la fenêtre *Simulink Library Browser* qui s'ouvre alors, cliquer sur l'icône « *Nouveau Modèle* » :



Une fenêtre de travail redimensionnable s'ouvre alors. C'est dans cette fenêtre que vous allez réaliser le schéma bloc modélisant votre système. Commencer par sauver (« *Fichier* » puis « *Save as...* ») sous le nom « **Essai** ».

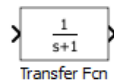
Dans l'arborescence du module *Simulink* de la fenêtre **Simulink Library Browser** , vous avez accès à des entrées (« *Sources* »), des sorties (« *Sinks* ») et des fonctions continues (« *Continuous* ») : très utiles pour modéliser des automatismes continus).

Nous allons élaborer un schéma bloc simple permettant de tester un moteur électrique à courant continu :

- Créer un **signal d'entrée de type échelon** en sélectionnant à la souris une entrée « *Step* » dans l'option « *Sources* » du **Simulink Library Browser**, puis en le plaçant par glisser-déposer dans la fenêtre de travail « *Essai* » ;



- Créer un **bloc fonctionnel** en sélectionnant « *Transfer fcn* » dans l'onglet « *Continuous* » de la fenêtre **Simulink Library Browser**;

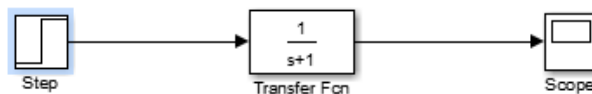


- Créer une **sortie** en sélectionnant « *Scope* » dans l'onglet « *Sinks* » de la fenêtre **Simulink Library Browser**;

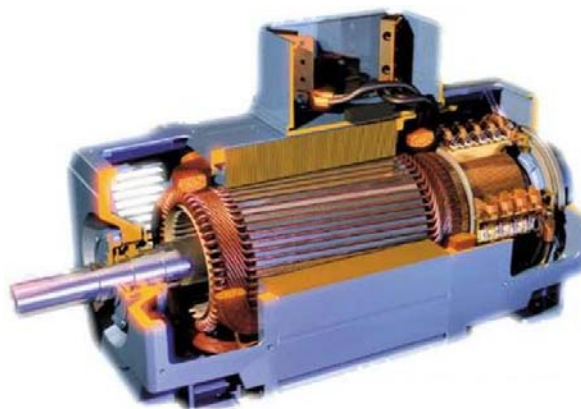


- Relier les entrées et sorties des blocs en cliquant sur les flèches correspondantes puis en créant les liens sur l'espace de travail « *Essai* ».

Une fois les blocs reliés, vous obtenez le schéma ci-dessous :



On va par la suite simuler la **commande en vitesse d'un moteur à courant continu** modélisé par une fonction de transfert du premier ordre : $F(p) = \frac{\Omega(p)}{U(p)} = \frac{10}{1 + 0.2p}$, avec une commande en tension $U(p)$ en entrée et une vitesse de sortie $\Omega(p)$. Le moteur est piloté par un signal échelon de valeur 1.5 V.



On va maintenant renseigner les blocs d'entrée et de fonction de transfert (système) :

- Double cliquer sur l'entrée « *Step* ».
 - Renseigner « *Step Time* » (instant de début d'échelon) : 0 (en secondes) ;
 - « *Initial Value* » (valeur initiale) : 0 ;
 - « *Final Value* » (valeur finale) : 1.5 ;
 - Laisser « *Sample Time* » à 0.

- Double cliquer sur le bloc « *Transfer fcn* » afin de modéliser le moteur. Rentrer la valeur des coefficients des polynômes situés au numérateur et dénominateur de $F(p)$, en prenant soin de commencer par les coefficients situés devant les termes de plus haut degré (cf. exemple ci-dessous).

Numerator coefficients:
[10]

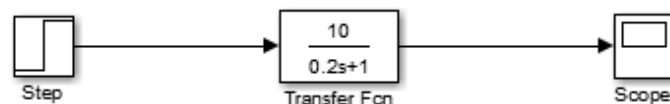
Denominator coefficients:
[0.2 1]

Absolute tolerance:
auto

State Name: (e.g., 'position')

NB : Vous aurez remarqué que le paramètre de Laplace (p) se nomme ici « s » : ce sont les notations U.S.

- Préciser la durée de simulation de 2 secondes dans le rectangle en haut de la fenêtre « Essai ». (NB : par défaut, valeur de 10 s)



- Lancer la simulation en cliquant sur la flèche verte en haut de la fenêtre « Essai » :
- Double cliquer sur la sortie « Scope » enfin sur afin d'obtenir le tracé de la variable de sortie en fonction du temps.
- Pour visualiser la courbe dans de meilleures conditions, vous pouvez agrandir la fenêtre. On peut utiliser les outils de « Zoom » et faire les mesures grâce à l'outil « Règle » :



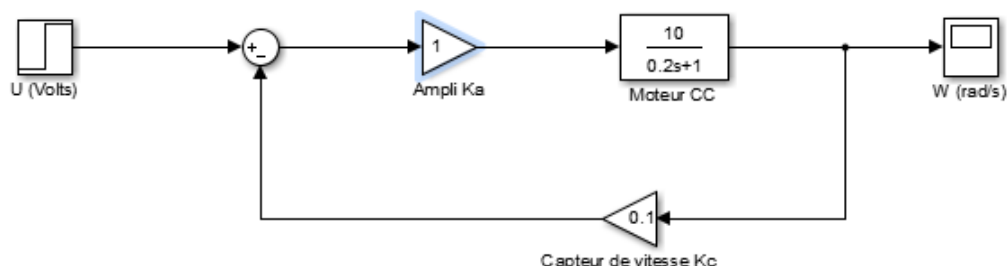
Cette icône permet un « Zoom au mieux sur la courbe », ce qui est souvent très utile :




Question : Trouver graphiquement la valeur du temps de réponse du moteur. Conclure (comparaison avec la valeur théorique de $3 \cdot \tau$).


3. Etude de l'asservissement en vitesse du moteur

Vous allez maintenant insérer le moteur dans une boucle d'asservissement en vitesse comme ci-dessous :



- Vous pouvez partir du schéma précédent, effacer les liens (cliquer sur les liens puis « suppr »).
- L'amplificateur et le capteur seront représentés par des gains pur (onglet «*Math Operations*» puis «*Gain*» dans la fenêtre **Simulink Library Browser**).
Afin de mettre le bloc « *Gain* » du capteur dans le bon sens, vous pouvez le retourner en faisant clic droit sur le bloc, « *Rotate & Flip* » puis « *Flip* » (ou ctrl+l)
- Pour former le comparateur, aller dans « *Commonly Used blocks* » puis « *Sum* ». Ensuite, transformer ce sommateur en comparateur en double cliquant dessus et en renseignant « + - » dans la rubrique « *List of signs* ».
- Pour former la séparation située avant le « *Scope* », il faut partir de l'entrée du capteur et remonter vers le lien préalablement créé entre la sortie du moteur et le « *Scope* ».
- Le capteur de vitesse sera de gain $K_c = 0.1 \text{ V.s.rad}^{-1}$
- L'amplificateur aura dans un premier temps un gain $K_a=1$.
- Vous pouvez renommer les blocs en leur donnant un sens physique en double cliquant sur leur nom (cf. ci-dessus).
- Lancer la simulation par appui sur la flèche verte.

 **Question :** Déterminer les temps de réponse ($Tr_{5\%}$) du système pour trois valeurs de K_a : $K_a=1$, $K_a=3$ et $K_a=5$.

 **Conclure** quant à l'influence du gain de l'amplificateur sur la rapidité du système bouclé ;

TP SYSTEMES ASSERVIS : SYSTEME DE REGULATION DE TEMPERATURE D'UN FOUR INDUSTRIEL

La durée est de *1 heure 30 minutes* et les activités feront l'objet d'un *compte-rendu*.

OBJECTIFS DU TP :

-
- **Modéliser** le système sous forme de schéma bloc.
- **Manipuler** le logiciel *Matlab Simulink* afin de **simuler** le fonctionnement du système
- **Déterminer** ses caractéristiques dans divers cas de sollicitations- tests.

1. PRESENTATION DU SYSTEME : FOUR INDUSTRIEL DE MICRO MECANIQUE

| |
|--|
| « requirement » Contrôler la température de l'enceinte chauffée |
| Id= « 3.5 » Text= « Le système doit être précis sous toutes conditions et le temps de préchauffage inférieur à 25 s » |



PROBLEMATIQUE : Régulation de température du four

Afin d'améliorer les propriétés mécaniques de petites pièces en acier (robotique, horlogerie, micro mécanique), on procède à des traitements thermiques modifiant certaines propriétés cristallines du matériau. Ces procédés nécessitent le **maintien à une température** donnée des pièces à traiter.

Les phases de **traitement thermique** des aciers peuvent être nombreuses :

- Le **recuit** consiste à :
 - chauffer la pièce à une température déterminée dite température de recuit (le choix de la température dépend des objectifs, elle peut aller de 450 et 1100°C)
 - maintenir cette pièce à cette température pendant un temps donné
 - refroidir à la vitesse adéquate afin d'obtenir après retour à la température ambiante un état structural du métal proche de l'état d'équilibre stable.

Cette définition très générale est habituellement complétée par une formulation précisant le but du traitement.

Le recuit permet notamment :

- d'éliminer ou réduire les contraintes résiduelles du métal liées à une action antérieure (déformation, soudure, etc.) ou un traitement thermique antérieur,
- ou d'obtenir la formation d'une structure favorable à une action ultérieure (déformation, usinage, etc.) ou un traitement thermique ultérieur.
- La **trempe** (ou durcissement par trempe) consiste à :
 - chauffer la pièce à une température appropriée (austénitisation ou mise en solution)
 - refroidir à une vitesse adéquate la pièce en la plongeant par exemple dans de l'eau (trempe à l'eau) ou de l'huile (trempe à l'huile) ou par insufflation d'un gaz tel que l'air. On peut évoquer

également les émulsions eau-polymères ainsi que certains sels (nitrates, nitrites) comme milieux de trempe.

Une trempe a pour objet de durcir le métal. Elle permet d'obtenir des aciers très durs mais dans la plupart des cas peu ductiles. Elle est donc généralement suivie d'un revenu.

Remarque : Il est possible de réaliser des trempes "locales" ne s'appliquant qu'à une partie d'une pièce.

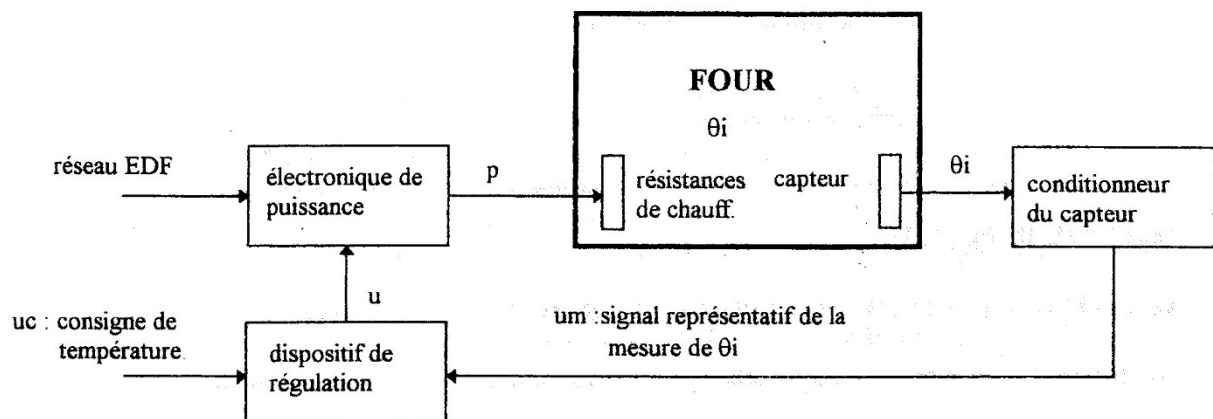
- Le **revenu** consiste :
 - en un ou plusieurs chauffages de la pièce à une température déterminée (inférieure à la température de début de transformation de phase de l'acier $\sim 700^\circ\text{C}$)
 - un maintien de la pièce à cette température pendant un temps donné
 - un refroidissement approprié.

Un revenu permet d'atténuer les effets de la trempe en rendant la pièce plus ductile et plus tenace. Il peut dans des cas particuliers permettre un durcissement secondaire. L'emploi du terme revenu sous-entend que le traitement est réalisé sur un état préalablement trempé. [Un traitement à 500°C sur un acier trempé est un revenu, sur un acier non traité est un recuit].

- Le **traitement de normalisation** (équivalent d'un recuit à haute température suivi d'un refroidissement à l'air) est un traitement d'affinage structural permettant un adoucissement et l'élimination des contraintes apparues avec les opérations précédentes. Outre l'obtention de structures homogènes recherchées dans certaines applications mécaniques il conduit à un état reproductible idéal pour la trempe. L'état dit "normalisé" est un état de livraison des produits sidérurgiques.
- Il faut également noter qu'avec l'évolution technologique on réalise à présent des produits avec des **traitements thermomécaniques** à hautes températures, sans oublier les **traitements thermochimiques** (cémentation, nitruration...).

On va par la suite s'intéresser à la **phase de revenu** à 700°C .

📖 La **structure** du four est donnée sur la figure suivante :



📖 **Éléments constitutifs du système :**


- La température à maintenir est appelée « **consigne de température** » notée $\theta_c(t)$. Elle est affichée sur un **afficheur/convertisseur** qui délivre une tension $u_c(t)$ image de cette consigne. Le convertisseur est considéré comme un élément proportionnel de gain $K_c = 0.01 \text{ V}/^\circ\text{C}$.
- La **température à l'intérieur du four** est notée $\theta_i(t)$.
- $\theta_i(t)$ est mesurée par un **capteur** de type « thermistance » qui conditionne l'information pour la transformer en une tension $u_m(t)$. Cet ensemble « capteur/conditionneur » peut être assimilé à un élément de type 1^{er} ordre de fonction de transfert :

$$R(p) = \frac{0.01}{1 + 4p}$$

- Les deux tensions, images des températures $\theta_c(t)$ et $\theta_i(t)$ sont comparées par un **dispositif de régulation** fournissant une tension $u(t) = u_c(t) - u_m(t)$ qui commande le dispositif d'alimentation des résistances chauffantes.
- Un **dispositif d'alimentation** de type proportionnel permet de distribuer la puissance électrique $p(t)$ à la résistance chauffante. Le gain de ce dispositif est noté K_a . On prendra dans un premier temps le réglage $K_a = 1000 \text{ W/V}$
- Une étude expérimentale permet de montrer que le **système thermodynamique** (résistances chauffantes + parois du four + pièces à traiter) qui transforme la puissance d'alimentation des résistances $p(t)$ en température $\theta_i(t)$, obéit à l'équation différentielle suivante :

$$\{1\} \quad \theta_i(t) + \tau \frac{d\theta_i(t)}{dt} = K.p(t) \quad \text{avec } K = 1^\circ\text{C/W et } \tau = 60\text{s}$$


2. MODELISATION ET ETUDE DES CARACTERISTIQUES TECHNIQUES DU FOUR INDUSTRIEL

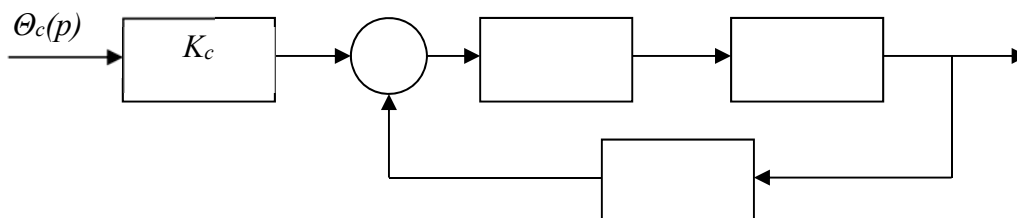
 **QUESTION N°1:** Déterminer la fonction de transfert $F(p)$ modélisant l'action du **système thermodynamique** décrit ci-dessus (équation {1}). De quel ordre est cette fonction de transfert ? En déduire ses constantes caractéristiques.


$F(p) =$

Ordre :

Constantes caractéristiques :

 **QUESTION N°2:** Compléter le schéma- bloc ci-dessous (en Laplace !) modélisant le système de régulation de température du four.




 **QUESTION N°3:** Saisir ce modèle sous le logiciel *Matlab Simulink*. Observer pendant 400s l'évolution temporelle de la température à l'intérieur du four pour une température de consigne de 700 °C. Déterminer alors graphiquement l'**erreur statique** \mathcal{E}_s du système ainsi que son **temps de réponse** à +/- 5% $Tr_{5\%}$:

$\mathcal{E}_s =$

$Tr_{5\%} =$


(NB : Réponse à reporter dans le tableau page suivante.)


Au moyen d'une étude expérimentale paramétrée, on a déterminé les valeurs de K_a permettant d'obtenir graphiquement un $Tr_{5\%}$ **minimum** dans les cas avec et sans dépassement autorisés de la valeur finale.

 **QUESTION N°4:** Faire la simulation pour les deux réglages de K_a donnés dans le tableau ci-dessous. Compléter alors le tableau avec les valeurs de l'**erreur statique** \mathcal{E}_s et du **temps de réponse** à +/- 5% $Tr_{5\%}$:


| Etude Expérimentale | Cas initial (Q n°3) | $T_{r5\%}$ minimal (Q n°4) : Cas sans dépassement de la valeur finale | $T_{r5\%}$ minimal (Q n°4) : Cas avec dépassement de la valeur finale |
|---------------------|---------------------|--|--|
| K_a | 1000 W/V | 430 W/V | 670 W/V |
| $T_{r5\%}$ | | | |
| ε_s | | | |

On choisit finalement par la suite de prendre comme réglage $K_a = 400$ W/V.


 **QUESTION N°5:** Pour ce nouveau réglage de K_a , conclure quant à la validation de l'**exigence 3.5** présentée en début de sujet.

 **QUESTION N°6:** Calculer littéralement la fonction de transfert en boucle fermée $G(p)$ équivalente au système modélisé précédemment. Que remarque-t-on ?

$$G(p) =$$

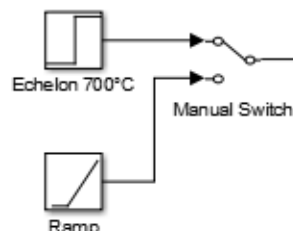
 **QUESTION N°7:** Le réglage de $T_{r5\%}$ n'étant pas suffisant, nous allons chercher à annuler l'erreur statique en jouant sur le gain K_c . Retrouver la valeur de K_c permettant d'obtenir un système précis (formule littérale et application numérique).

$$K_c =$$

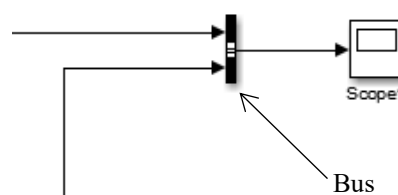
 **QUESTION N°8:** Modifier le modèle *Simulink* en injectant la nouvelle valeur de K_c . Déterminer la nouvelle valeur expérimentale de l'erreur statique ε_s . Conclure quant à la validation de l'**exigence 3.5** présentée en début de sujet.


On désire élever progressivement mais linéairement la température au sein du four à la vitesse de 10 °C/s, avant d'atteindre une température constante de 700 °C. Le système sera considéré comme réglé ($\varepsilon_s = 0$, $K_a = 400$ W/V).

Afin de conserver la commande de type échelon, on peut rajouter un *switch* (« *Signal Routing* » dans *Simulink Library Browser*, puis « *Manual Switch* ») en entrée comme ceci :



Afin de visualiser l'entrée et la sortie dans le « *Scope* », on peut saisir l'information en sortie du « *Switch* » et la rassembler avec la sortie avant d'entrer dans la « *Scope* » au moyen d'un « *Bus* » (« *Signal Routing* » dans *Simulink Library Browser*, puis « *Bus Creator* ») :




 **QUESTION N°9:** Quel type de sollicitation doit-on employer ? Déterminer l'erreur dynamique \mathcal{E}_v du système dans sa phase de montée en température. Retrouver ces résultats par le calcul et remplir le tableau ci-dessous :

| Résultats | Expérimental | Calcul (calcul littéral et application numérique) |
|-----------------|--------------|--|
| Sollicitation | | |
| \mathcal{E}_v | | |

Par la suite, on désire étudier l'influence de la **perturbation** de l'ouverture de la porte lors du chargement du four (durée de chargement : 30 secondes). Celle -ci va générer une perte de puissance P_p de 1200 W pendant 30 secondes. On fait intervenir cette perturbation au bout de 1 minute après démarrage du four. On Considère à nouveau une entrée principale de type échelon de 700 °C.

 **QUESTION N°10:** Quel signal peut-on mettre en place pour modéliser cette perturbation ?

On pourra modéliser cette entrée au moyen d'un module d'entrée « *Signal Builder* » (« *Sources* » dans *Simulink Library Browser*, puis « *Signal Builder* »). Vous pouvez alors sélectionner les points constituant ce signal et les modifier soit par glisser / déposer, soit en entrant leurs coordonnées. Ce signal sera injecté dans la chaîne directe (au bon endroit !) en utilisant un bloc « *Add* » (« *Math Operations* » dans *Simulink Library Browser*, puis « *Add* »)

 **QUESTION N°11:** Modifier alors le modèle *Simulink* comme ci-dessous et **conclure** quant au comportement du système.

