

---

MASTER ISTR-RODECO-E2-SIA

---

# Systemes linéaires à Temps Continu

## Travaux pratiques

---

Année scolaire 2018-2019

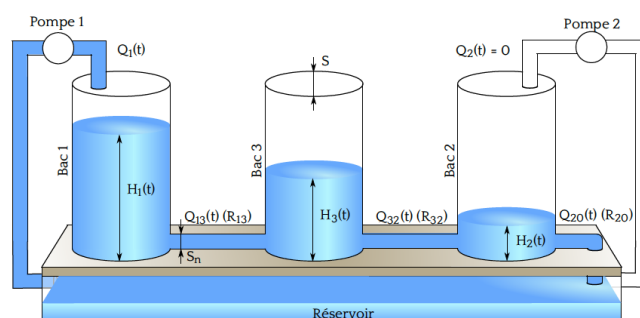
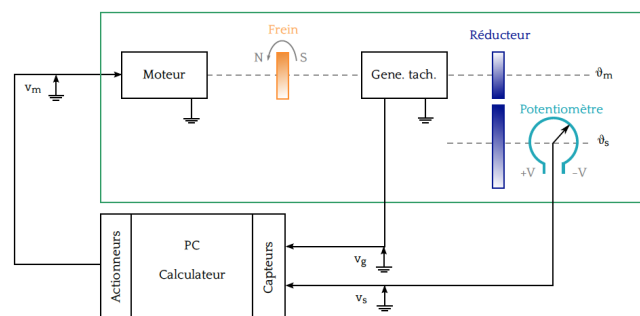
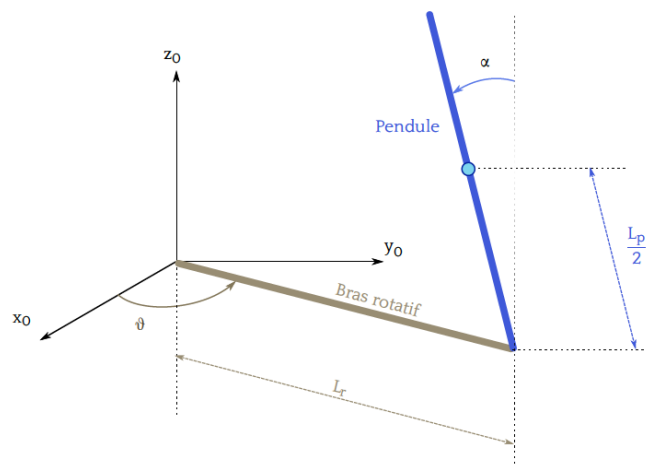
# 1. Présentation des TP et Organisation

Les Travaux pratiques des modules de Systèmes Linéaires 1 & 2 se composent de 4 séances dédiées à un thème précis:

1. Analyse et commande d'un pendule inverse.
2. Analyse et commande d'un moteur à courant continu.
3. Commande basée observateur d'un système à trois bacs d'eau.
4. Commande basée observateur d'un moteur à courant continu.

Les deux premiers constituent les TP du premier module **obligatoire** pour les étudiants d'ISTR-RODECO et E2, **optionnel** pour SIA. Les deux derniers TP constituent les TP du second module, **obligatoire** pour les étudiants d'ISTR-RODECO seulement.

Les travaux pratiques s'effectueront en binômes en respectant la composition des groupes de TP. Les TP doivent être préparés dans cet ordre pour l'ensemble de la promotion et un compte rendu devra être rédigé en LATEX et rendu à l'encadrant de la séance une semaine après les TP.



MASTER ISTR-RODECO / E2 / SIA

---

## SLI 1 - I4

### TP 1: PENDULE

---

#### RÉGULATION D'UN PENDULE INVERSÉ

Année scolaire 2016-2017

Jour de TP:	
Date limite envoi CR:	
Enseignant(s):	
E-mail(s) enseignant(s):	

## 1. BUT DE LA MANIPULATION

Dans cette manipulation on se propose de réaliser un asservissement permettant la régulation du pendule inversé représentée par la Figure 1.1 en position haute comme le montre la Figure 1.2. On utilisera pour cela les techniques d'espace d'état à temps continu. Cette manipulation est constituée de quatre grandes étapes: modélisation, analyse, régulation et implémentation.

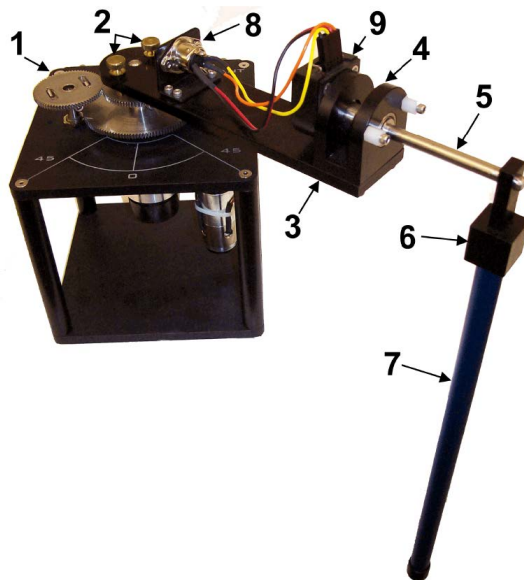


Figure 1.1: Composants du pendule.



Figure 1.2: Pendule régulé en position haute.

Les objectifs du TP sont les suivants:

- Savoir établir un modèle linéarisé autour d'un point d'équilibre.
- Savoir établir les propriétés structurelle d'un système linéaire.
- Savoir synthétiser une commande par retour d'état.
- Savoir faire la simulation de modèles linéaires et non linéaires.
- Savoir utiliser les différentes commandes MATLAB pour étudier un système.
- Savoir utiliser Matlab et sa toolbox simulink.

Avant d'arriver en séance, vous devez impérativement:

1. avoir obtenu le modèle linéaire dans l'espace d'état;
2. avoir réalisé toute l'étude théorique de l'analyse et de la commande.

Les fonctionnalités de MATLAB et de SIMULINK seront exploitées au mieux lors de la séance de Travaux Pratiques. Les principales fonctions utilisées font partie de la *CONTROL-TOOLBOX*, pensez à utiliser l'aide de Matlab et la fonction `help`.

Les tests expérimentaux sur le procédé réel se feront en fin de séance: toutes les parties théoriques seront complétées et les tests en simulation MATLAB-SIMULINK seront réalisés et validés par les enseignants.

## 2. PRÉSENTATION DU PENDULE INVERSÉ

Le pendule inversé de l'entreprise QUANSER est composé d'un moteur SVR02, qui entraine un bras disposant à son extrémitée un pendule (tige) lié l'un à l'autre par une liaison pivot. Ce système possède deux capteurs (une

roue codeuse chacune, voir N9 de Table 2.1) qui nous renseignent sur l'angle du pendule par rapport au bras et sur l'angle du bras par rapport au bâti. Les différents éléments du pendule sont décrits par la Figure 2.1.

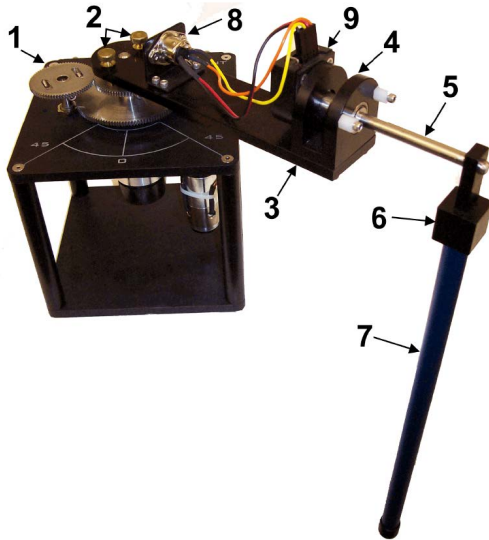


Figure 2.1: Composants du pendule.

N <sup>o</sup>	Composant
1	SRVO2
2	Vis
3	Bras
4	Logement d'arbre
5	Arbre
6	Raccord
7	Pendule
8	Connecteur de l'encodeur
9	Encodeur

Table 2.1: Nomenclature des composants de la maquette.

### 3. MODÉLISATION

La Figure 3.1 donne une représentation schématique du pendule. Le bras est relié au moteur, qui entraîne la rotation du bras et donc du pendule. Il a une longueur  $L_r$  et un moment d'inertie  $J_r$ . Nous noterons son angle  $\theta$ .

Le pendule, quant à lui, est fixé au bout du bras. Il a une longueur  $L_p$ . Son moment d'inertie au centre de masse est noté  $J_p$ . On note l'angle que fait l'axe du pendule avec l'axe verticale  $\alpha$  (en radian).

Tous les calculs de la modélisation se feront en prenant pour origine, la position haute du pendule. Pour les descriptions et les valeurs numériques des constantes se reporter à la Table 3.1.

Les équations du mouvement, déterminées à l'aide de la mécanique Lagrangienne, sont données par les deux équations différentielles suivantes:

$$\begin{aligned} & \left( m_p L_r^2 + \frac{1}{4} m_p L_p^2 - \frac{1}{4} m_p L_p^2 \cos(\alpha)^2 + J_r \right) \ddot{\theta} - \left( \frac{1}{2} m_p L_p L_r \cos(\alpha) \right) \ddot{\alpha} + \left( \frac{1}{2} m_p L_p^2 \sin(\alpha) \cos(\alpha) \right) \dot{\theta} \dot{\alpha} + \\ & \left( \frac{1}{2} m_p L_p L_r \sin(\alpha) \right) \dot{\alpha}^2 = \frac{\eta_g K_g \eta_m k_t (V_m - K_g k_m \dot{\theta})}{R_m} - B_r \dot{\theta} \\ & - \frac{1}{2} m_p L_p L_r \cos(\alpha) \ddot{\theta} + \left( J_p + \frac{1}{4} m_p L_p^2 \right) \ddot{\alpha} - \frac{1}{4} m_p L_p^2 \cos(\alpha) \sin(\alpha) \dot{\theta}^2 - \frac{1}{2} m_p L_p g \sin(\alpha) = -B_p \dot{\alpha} \end{aligned} \quad (3.1)$$

#### 3.1. Étude succincte du modèle non linéaire

Dans un premier temps, nous allons étudier le modèle non linéaire représenté par les deux équations du mouvement (3.1).

1. Proposez les variables d'état. Combien d'états choisissez-vous?
2. Déterminer les points d'équilibre en fonction de  $V_m$  (tension) constante. Combien y-en-a-t-il? Pourquoi? Montrer qu'il n'existe pas de point d'équilibre lorsque la tension appliquée au moteur est non nulle.

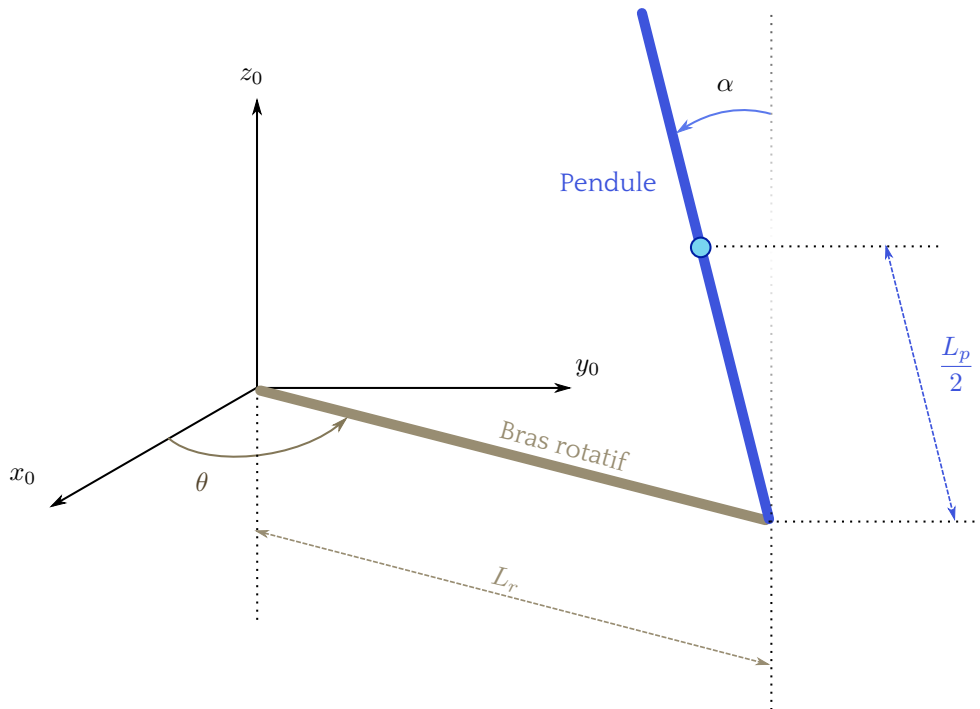


Figure 3.1: Schéma du pendule rotatif.

3. Les valeurs des points d'équilibre vous semblent-elles en accord avec le dispositif réel?
4. En utilisant le simulateur non linéaire proposé, simuler le dispositif non linéaire aux points d'équilibre. Que se passe-t-il?
5. En utilisant le simulateur non linéaire proposé, simuler pour des conditions initiales  $\{\alpha = +1, \theta = 0, \dot{\alpha} = 0, \dot{\theta} = 0\}$  et  $\{\alpha = -1, \theta = 0, \dot{\alpha} = 0, \dot{\theta} = 0\}$ . Obtenons-nous les mêmes valeurs finales?

### 3.2. Linéarisation et obtention d'un modèle linéaire

Nous choisissons comme variable d'état

$$x(t) = \begin{bmatrix} \theta(t) & \alpha(t) & \dot{\theta}(t) & \dot{\alpha}(t) \end{bmatrix}^T.$$

6. Linéariser le système non linéaire autour du point d'équilibre  $x_e = [0 \ 0 \ 0 \ 0]^T$ . On pourra à cet effet utiliser une décomposition en série de Taylor arrêtée à l'ordre 1:

$$\tilde{f}(z) = f(z_0) + \sum_{i=1}^n \frac{\partial f}{\partial z_i} \Big|_{z_0} (z_i - z_0).$$

Montrer que l'on obtient pour le modèle linéarisé en position haute le quadruplet  $(A_{haut}; B_{haut}; C_{haut}; D_{haut})$ .

Symbole	Description	Valeur
$B_p$	Coefficient de frottement visqueux du pendule	$0.0024 \text{ kg/m}^2$
$B_r$	Coefficient de frottement visqueux du bras	$0.0024 \text{ kg/m}^2$
$\eta_g$	Rendement du réducteur	0.9
$\eta_m$	Rendement du moteur	0.69
$K_g$	Rapport de vitesse	70
$k_m$	Gain f.e.m	$7.68 \cdot 10^{-3} \text{ V/(rad/s)}$
$k_t$	Gain couple moteur/courant	$7.68 \cdot 10^{-3} \text{ N} \cdot \text{m/A}$
$J_p$	Moment d'inertie du pendule au centre de masse	$0.0012 \text{ kg/m}^2$
$J_r$	Moment d'inertie du bras au centre de masse	$9.98 \cdot 10^{-4} \text{ kg/m}^2$
$L_m$	Inductance du moteur	$0.18 \text{ mH}$
$L_p$	Longueur du pendule	$0.337 \text{ m}$
$L_r$	Longueur du bras, du pivot à l'extrémité	$0.216 \text{ m}$
$m_p$	Masse du pendule	$0.127 \text{ kg}$
$m_r$	Masse du bras	$0.257 \text{ kg}$
$R_m$	Résistance du moteur	$2.60 \Omega$
$V_m$	Tension envoyé au moteur	À régler

Table 3.1: Liste des données spécifiques au système.

$D_{haut}$ ) (vous pourrez faire les calculs numériquement pour plus de simplicité):

$$A_{haut} = \frac{1}{J_t} \begin{bmatrix} 0 & 0 & J_t & 0 \\ 0 & 0 & 0 & J_t \\ 0 & \frac{1}{4} m_p^2 L_p^2 L_r g & \left( J_p + \frac{1}{4} m_p L_p^2 \right) (-B_r - K_g k_m \gamma) & -\frac{1}{2} m_p L_p L_r B_p \\ 0 & \frac{1}{2} m_p L_p g (J_r + m_p L_r^2) & \left( \frac{1}{2} m_p L_p L_r \right) (-B_r - K_g k_m \gamma) & -(J_r + m_p L_r^2) B_p \end{bmatrix} \quad (3.2)$$

$$B_{haut} = \frac{1}{J_t} \begin{bmatrix} 0 \\ 0 \\ \left( J_p + \frac{1}{4} m_p L_p^2 \right) \gamma \\ \frac{1}{2} m_p L_p L_r \gamma \end{bmatrix} \quad C_{haut} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

avec

$$\gamma = \frac{\eta_m \eta_m K_g k_t}{R_m} \quad J_t = J_p m_p L_r^2 + J_r J_p + \frac{1}{4} J_r m_p L_p^2.$$

7. Définir le système linéaire sous Matlab à l'aide de la commande `ss(A,B,C,D)`.

8. Simuler la réponse à des conditions initiales et comparer avec le simulateur non linéaire. Discuter de la validité du modèle linéarisé.

## 4. ANALYSES TEMPORELLE ET STRUCTURELLE DU MODÈLE LINÉARISÉ

---

1. Analyser la stabilité asymptotique du point d'équilibre haut. (eig).
2. Calculer la matrice de commandabilité. Donner son rang. (ctrb, rank).
3. Calculer la matrice d'observabilité. Donner son rang. (obsv, rank).
4. Conclure sur la commandabilité et l'observabilité du modèle linéarisé.
5. Facultatif. Déterminer la fonction de transfert du modèle linéarisé. Donner les ordres des fonctions de transfert. Est-ce logique ?

## 5. RÉGULATION DU MODÈLE LINÉARISÉ

---

On rappelle que l'on ne dispose pas des mesures de vitesse angulaire. Dans le cadre du TP de SLI1, ces données seront supposées déjà reconstituées afin de pouvoir faire un retour d'état.

1. Proposer un schéma de loi de commande permettant de placer les valeurs propres du système bouclé en  $\begin{bmatrix} -108 & -0.3 & -4.3 + 2.2i & -4.3 - 2.2i \end{bmatrix}^T$ . Justifiez le choix des valeurs propres.
2. Simuler la réponse aux conditions initiales sur le modèle linéarisé. (on pourra prendre des conditions initiales proches du point d'équilibre).
3. Simuler la réponse aux conditions initiales sur le modèle non linéaire en utilisant le simulateur non linéaire proposé. Comparer avec le modèle linéaire.
4. Que se passe-t-il si nous choisissons des conditions initiales trop éloignées du point d'équilibre?
5. En utilisant le reconstruteur d'état proposé, vérifier en simulation que le système en boucle fermée est bien stabilisé.

## 6. IMPLANTATION LOI DE COMMANDE SUR LE DISPOSITIF RÉEL

---

La dernière section nous a permis de simuler et donc de valider le choix d'un retour d'état stabilisant pour la position haute du pendule. Nous pouvons désormais implanter le retour d'état sur le dispositif réel. Le fichier `t_maintient_pendule_en_haut.mdl` est un fichier analogue au fichier Simulink que vous avez utilisé pour la simulation. Par contre, le bloc espace d'état est remplacé par un bloc dont les entrées et les sorties sont celles d'une carte E/S pilotant le pendule inverse réel. Une toolbox de Matlab génère à partir du modèle un code exécuté en temps réel par le PC et dialoguera avec le système réel au travers de la carte.

À FAIRE AVEC L'ENSEIGNANT.

1. Ouvrir le fichier `t_maintient_pendule_en_haut.mdl` et entrer les valeurs du retour d'état dans la boîte prévue à cet effet.
2. Compiler le fichier, connecter et laisser l'interrupteur sur 0, monter le pendule à la main et switcher la commande.



3. Que constatez vous sur l'angle  $\theta$ ?

## 7. APPROCHE ROBUSTE: COMMANDE INTÉGRALE

Supposons que les perturbations affectant le système en boucle fermée puissent être vu comme une perturbation constante agissant sur la sortie  $\theta$ . On considère le dispositif de commande de la Figure 7.1.

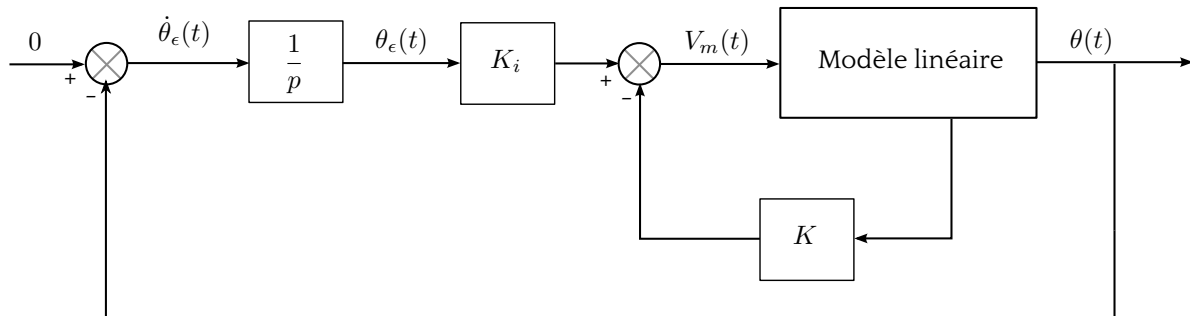


Figure 7.1: Retour d'état avec commande intégrale.

1. Écrire la représentation d'état de ce système en considérant le nouveau vecteur d'état

$$x_{int}(t) = \begin{bmatrix} \theta(t) & \alpha(t) & \dot{\theta}(t) & \dot{\alpha}(t) & \theta_e(t) \end{bmatrix}^T.$$

Le retour d'état considéré sur la Figure 7.1 conduit à la loi de commande  $V_m = -K_{int} \cdot x_{int}$ , avec  $K_{int} = [K \ K_i]$ .

2. Calculer le retour d'état  $K_{int}$  en plaçant les valeurs propres du système bouclé avec intégrateur en

$$\begin{bmatrix} -108 & -4.3 + 2.25i & -4.3 - 2.25i & -0.4 + 0.35i & -0.4 - 0.35i \end{bmatrix}^T. \text{ A FAIRE AVEC L' ENSEIGNANT.}$$

3. Réaliser l'implémentation sur le dispositif réel.

Que constatez vous?

## 8. MONTÉE DU PENDULE

On se propose maintenant de partir du point d'équilibre bas pour arriver au point d'équilibre haut.

1. En sachant que les équation non linéaires ont pour origine le point d'équilibre haut, comment feriez vous pour obtenir un modèle linéarisé au point d'équilibre bas?
2. Proposez une loi de commande permettant de réaliser cette application (aucun calcul n'est demandé).

*Remerciements: Une grande partie de ce TP a été réalisé par des étudiants du M1 ISTR 2012/2013, Messieurs Gelas, Maitre, Miral, suite à un TER sur le même sujet. Merci à eux pour leur contribution efficace.*

## 9. ANNEXE: DÉTERMINATION DES ÉQUATIONS DU MOUVEMENT

Les équations du mouvement, déterminées à l'aide de la mécanique Lagrangienne. Une des grandes forces de la description lagrangienne est de pouvoir faire abstraction des contraintes d'un système, qu'il s'agisse de l'angle formé par un pendule ou de l'abscisse d'un mobile, toutes les coordonnées sont traitées sur un pied d'égalité: on parle de *coordonnées généralisées* (appelée aussi degrés de libertés). Ici nous avons deux coordonnées généralisées l'angle du bras et l'angle du pendule:

$$q(t) = \begin{bmatrix} \theta(t) \\ \alpha(t) \end{bmatrix}.$$

Le centre de la description lagrangienne de la mécanique est une quantité, appelée *lagrangien*. Il s'agit d'une énergie, définie par:

$$L(q, \dot{q}) = T(q, \dot{q}) - V(q);$$

où  $T$  est l'énergie cinétique totale du système (qui fait intervenir les vitesses) et  $V$  est l'énergie potentielle totale (qui fait intervenir les positions).

On peut maintenant obtenir alors un jeu d'équations différentielles appelées équations d'Euler-Lagrange. Pour un système possédant  $n$  degrés de liberté spatiale, donc décrit par  $n$  coordonnées généralisées, il y a  $n$  équations d'Euler-Lagrange. Le procédé possédant deux degrés de liberté, on a donc deux équations:

$$\begin{aligned} \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} &= Q_1 \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\alpha}} \right) - \frac{\partial L}{\partial \alpha} &= Q_2; \end{aligned}$$

où  $Q_1, Q_2$  sont les forces généralisées (forces et couples) externes au système ou ne dérivant pas d'un potentiel. Ici, pour les forces qui concernent le bras ( $Q_1$ ), il y a des frottements visqueux et le couple résultant du moteur à courant continu. Pour le pendule ( $Q_2$ ), il y a seulement des frottements visqueux qui sont pris en compte.

C'est ici qu'apparaissent les avantages de cette description: peu importe le système de coordonnées ou la complexité du système, ce sont toujours ces mêmes équations qui le régissent. On arrive ainsi toujours à établir les équations du mouvement.

Finalement, après des calculs on obtient un jeu de deux équations différentielles qui décrivent le mouvement du procédé:

$$\begin{aligned} \left( m_p L_r^2 + \frac{1}{4} m_p L_p^2 - \frac{1}{4} m_p L_p^2 \cos(\alpha)^2 + J_r \right) \ddot{\theta} - \left( \frac{1}{2} m_p L_p L_r \cos(\alpha) \right) \ddot{\alpha} + \left( \frac{1}{2} m_p L_p^2 \sin(\alpha) \cos(\alpha) \right) \dot{\theta} \dot{\alpha} + \\ \left( \frac{1}{2} m_p L_p L_r \sin(\alpha) \right) \dot{\alpha}^2 = \frac{\eta_g K_g \eta_m k_t (V_m - K_g k_m \dot{\theta})}{R_m} - B_r \dot{\theta} \\ - \frac{1}{2} m_p L_p L_r \cos(\alpha) \ddot{\theta} + \left( J_p + \frac{1}{4} m_p L_p^2 \right) \ddot{\alpha} - \frac{1}{4} m_p L_p^2 \cos(\alpha) \sin(\alpha) \dot{\theta}^2 - \frac{1}{2} m_p L_p g \sin(\alpha) = -B_p \dot{\alpha} \end{aligned}$$

MASTER ISTR-RODECO / E2 / SIA

## SLI 1 - I4

### TP 2: MOTEUR I

SYNTHESE D'UN ASSERVISSEMENT DE POSITION PAR RETOUR D'ETAT  
ET TACHYMETRIQUE

IMPLEMENTATION AVEC MATLAB/SIMULINK ET RTW TARGET

Année scolaire 2017-2018

Jour de TP:	
Date limite envoi CR:	
Enseignant(s):	
E-mail(s) enseignant(s):	

# 1. INTRODUCTION

Cette manipulation se propose de réaliser un asservissement de position en mettant en œuvre les techniques d'espace d'état continu. Le procédé utilisé est la platine "asservissement de position" qui apparaît dans la Figure 1.1, déjà présentée dans les textes des manipulations des TP de licence EEA (2ème année) dont le principe est rappelé sur le schéma présenté dans la Figure 1.2.

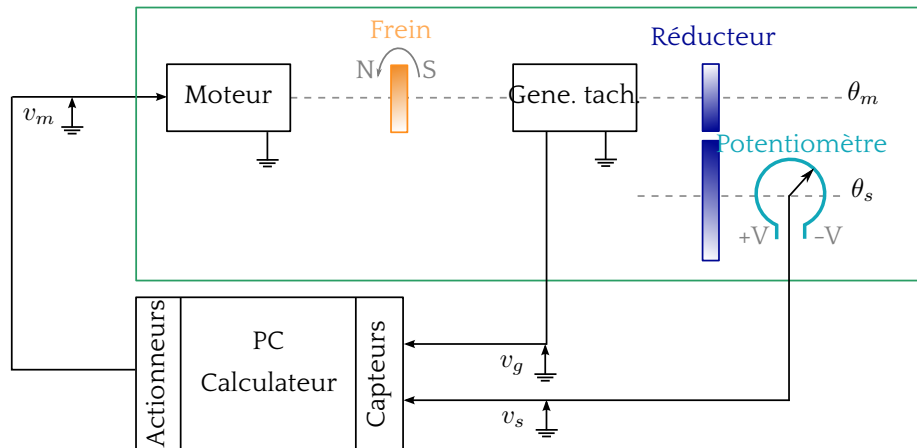


Figure 1.1: Asservissement de position par ordinateur.

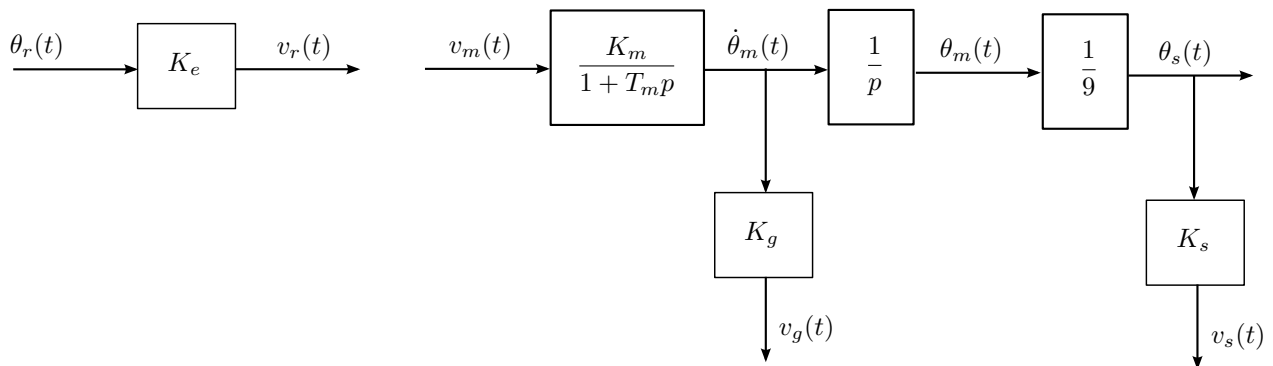


Figure 1.2: Eléments de la platine "asservissement de position".

- $v_m(t)$  est la tension d'entrée du moteur.
  - $\theta_s(t)$  est la position de l'axe secondaire du moteur.
  - $\theta_m(t)$  est la vitesse de rotation de l'axe principal du moteur.
- Les paramètres  $K_m$  et  $\tau_m$  seront déterminés dans la section 2.3 . Les coefficients connus sont  $K_e$ ,  $K_s$  et  $K_g$ , leurs valeurs numériques sont données comme suit:

$$K_e = 10(V \cdot tr^{-1})$$

$$K_s = 10(V \cdot tr^{-1})$$

$$K_g = 0.105(V \cdot s \cdot tr^{-1})$$

Il est possible d'obtenir une mesure de  $\theta_s$  et de  $\theta_m$  via un potentiomètre et une génératrice tachymétrique.

$$v_s(t) = K_s \theta_s(t) \quad \text{et} \quad v_g(t) = K_g \dot{\theta}_m(t)$$

Avant d'arriver en séance, vous devez impérativement:

1. avoir obtenu les modèles linéaires dans l'espace d'état pour les systèmes  $S_{v_m \rightarrow v_s}$  et  $S_{v_m \rightarrow v_g}$ ,
2. avoir réalisé toute l'étude théorique de l'analyse et de la commande (choisir par défaut  $K_m = 7$  et  $T_m = 0.3$  pour votre préparation théorique).

Sans ces réalisations, il est illusoire de traiter le sujet.

- Les fonctionnalités de MATLAB et de MATLAB-SIMULINK seront exploitées au mieux lors de la séance de Travaux Pratiques pour valider la théorie ainsi que pour toutes les simulations.
- Les tests expérimentaux sur le procédé réel se feront en fin de séance, une fois que la partie théorique et la partie simulation MATLAB-SIMULINK seront finalisées et approuvées par les enseignants. Pour obtenir ces tests expérimentaux (c'est-à-dire commander en temps réel le procédé via MATLAB-SIMULINK), on utilisera 1 PC assisté d'une carte entrée-sortie (DAQ6024E) et d'un boîtier de raccordement (PC, procédé). Dans cette manipulation, nous chercherons à n'utiliser qu'un PC pour effectuer la commande du procédé en temps réel, par un calculateur. Ainsi, aucun matériel supplémentaire ne sera nécessaire (GBF, oscilloscope, platines dédiées à base d'amplificateurs opérationnels...), comme le montre la Figure 1.1.

## 2. ÉTUDE PRÉLIMINAIRE DU PROCÉDÉ

### 2.1. Représentation et analyse de l'ensemble {moteur, réducteur, potentiomètre, génératrice tachymétrique} pour le système $S_{v_m \rightarrow v_s}$

On considère le système en boucle ouverte  $S_{v_m \rightarrow v_s}$ , qui admet comme entrée le signal  $v_m(t)$  et comme sortie le signal mesurée  $v_s(t)$ .

1. Écrire la représentation d'état de ce système, en choisissant un vecteur d'état défini par:

$$\mathbf{x}(t) = [x_1(t) \ x_2(t)]^T = [v_s(t) \ v_g(t)]^T.$$

2. Nous supposons que  $v_m(t)$  est constante (c'est à dire,  $v_m(t) = v_{m_0}$ ,  $v_{m_0}$  étant constante):
  - a) Déterminer les points d'équilibre du système  $S_{v_m \rightarrow v_s}$ .
  - b) Étudier la stabilité du système  $S_{v_m \rightarrow v_s}$  autour de ses points de fonctionnement, s'ils existent.
3. Quels sont les modes associés au système  $S_{v_m \rightarrow v_s}$ ? Interpréter physiquement le résultat de cette analyse.
4. Le système  $S_{v_m \rightarrow v_s}$ , est-il commandable? Interpréter physiquement le résultat de cette analyse.
5. Le système  $S_{v_m \rightarrow v_s}$ , est-il observable? Interpréter physiquement le résultat de cette analyse.
6. Justifier que ce système admet des représentations compagnes de commande et d'observation. Déterminer ces représentations, ainsi que les changements de base qui les lient à la représentation initiale. On notera
  - $x_c$ : les variables d'état associées à la forme compagne de commande,
  - $x_o$ : les variables d'état associées à la forme compagne d'observation.

## 2.2. Représentation et analyse de l'ensemble {moteur, réducteur, potentiomètre, génératrice tachymétrique} pour le système $S_{v_m \rightarrow v_g}$

On considère le système en boucle ouverte  $S_{v_m \rightarrow v_g}$ , qui admet comme entrée le signal  $v_m(t)$  et comme sortie le signal mesurée  $v_g(t)$ .

1. Écrire la représentation d'état de ce système en conservant la définition du vecteur d'état choisie dans la sous section 2.1.
2. Vérifier que les propriétés de stabilité du système  $S_{v_m \rightarrow v_g}$  sont identiques à celles de  $S_{v_m \rightarrow v_s}$ .
3. Le système  $S_{v_m \rightarrow v_g}$  est-il commandable?
4. Montrer que le système  $S_{v_m \rightarrow v_g}$  n'est pas observable.
  - Déterminer quel état est concerné par cette perte d'observabilité.
  - Pourquoi ce résultat était-il prévisible?

## 2.3. Identification des coefficients $K_m$ et $T_m$

1. Déterminer expérimentalement le gain en vitesse  $K_m$  et la constante de temps mécanique  $T_m$  du moteur à courant continu en indiquant le principe de la mesure ainsi que son mode opératoire.

## 3. MISE EN PLACE D'UN RETOUR D'ÉTAT ET D'UN PRÉ-COMPENSATEUR

L'asservissement de position du système  $S_{v_m \rightarrow v_s}$ , est réalisé selon la loi de commande par retour d'état  $v_m(t) = NK_e \theta_r(t) - \mathbf{K}x(t) = Nv_r(t) - \mathbf{K}x(t)$ , avec  $\mathbf{K} = [k_1 \ k_2]$ .

### 3.1. Calcul des paramètres du retour d'état dans la base initiale

1. Déterminer l'expression du modèle du système asservi par retour d'état nommé  $(A_{bf}, B_{bf}, C_{bf}, D_{bf})$  en fonction de  $\mathbf{K}$  et  $N$ .
2. Donner l'expression de la matrice  $A_{bf}$  en fonction de  $k_1, k_2$ .
3. Justifier la possibilité de placer les valeurs propres de la matrice  $A_{bf}$  en
 
$$v_1, v_2 = -2.4 \pm 5.5j.$$
4. Calculer  $\mathbf{K}$  tel que les valeurs propres du système asservi soient  $v_1$  et  $v_2$ .

### 3.2. Calcul des paramètres du retour d'état en utilisant la forme compagne de commande

On souhaite utiliser la forme compagne de commande pour calculer le gain du retour d'état  $\mathbf{K}$ .

1. Déterminer la loi de commande dans la forme compagne de commande.
2. Déterminer l'expression du modèle du système asservi par retour d'état nommé  $(A_{cbf}, B_{cbf}, C_{cbf}, D_{cbf})$  en fonction de  $\mathbf{K}_c$  et  $N$ , où  $\mathbf{K}_c = [k_{1c} \ k_{2c}]$  est l'expression du gain du retour d'état dans la forme compagne de commande.
3. Calculer la matrice dynamique  $A_c - B_c \mathbf{K}_c$  en fonction de  $k_{1c}$  et  $k_{2c}$ .
4. Déterminer  $k_{1c}$  et  $k_{2c}$  afin d'obtenir des valeurs propres en boucle fermée égale à

$$v_1, v_2 = -2.4 \pm 5.5j.$$

5. Retrouver la valeur du gain  $\mathbf{K}$  en fonction de  $\mathbf{K}_c$ .

### 3.3. Calcul du pré-compensateur $N$

1. Établir l'expression du gain statique de l'asservissement lorsqu'il est asymptotiquement stable ; en déduire qu'il peut être modifié par action sur  $N$ .
2. Calculer le pré-compensateur  $N$  permettant d'annuler l'erreur de position de l'asservissement.

### 3.4. Simulations du système asservi par retour d'état

1. Simuler à l'aide de MATLAB-SIMULINK le schéma de l'asservissement, en choisissant différentes conditions initiales et plusieurs amplitudes pour  $v_r(t)$ .
2. Tracer différentes courbes permettant d'illustrer le choix des valeurs propres.
3. Donner les caractéristiques temporelles de l'asservissement (dépassement, temps de montée...) et expliquez vos résultats.
4. Que se passe-t-il si  $v_r(t) = t, \forall t > 0$ ? Justifiez votre réponse.

## 4. COMPARAISON AVEC UN ASSERVISSEMENT DE POSITION AVEC CORRECTION PROPORTIONNELLE ET RETOUR TACHYMÉTRIQUE

### 4.1. Retour de sortie avec correction proportionnelle

On considère l'asservissement de position suivant:

$$v_m(t) = k'_1(v_r(t) - v_s(t)) = k'_1 K_e(\theta_r(t) - \frac{K_s}{K_e}\theta_s(t)),$$

avec  $k'_1 > 0$  et ici,  $K_e = K_s$ . Cet asservissement est présenté dans la Figure 4.1.

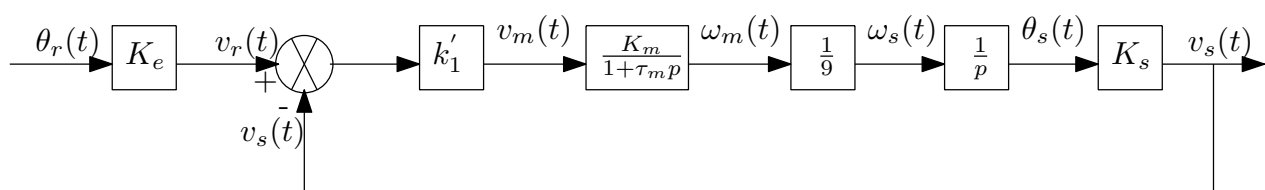


Figure 4.1: Asservissement de position avec correction proportionnelle

1. Écrire la représentation d'état de l'asservissement.
2. Donner l'expression de la fonction de transfert de l'asservissement.
3. Déterminer les expressions analytiques du gain statique, des zéros et des pôles de l'asservissement. En déduire l'amortissement et la pulsation naturelle de ses pôles dominants.
4. Calculer l'erreur de position et l'erreur de traînage à une rampe unitaire.
5. Tracer le lieu sur lequel les pôles de l'asservissement peuvent être placés par action sur  $k'_1 > 0$ .

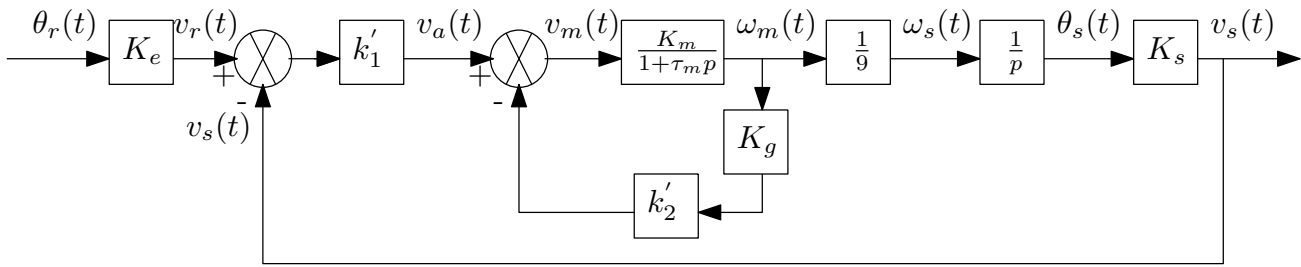


Figure 4.2: Asservissement de position avec contre-réaction tachymétrique.

## 4.2. Mise en place d'une contre-réaction tachymétrique

On ajoute une correction tachymétrique dans le schéma précédent, conformément à la Figure 4.2.

1. Écrire la fonction de transfert de la boucle interne de vitesse ( d'entrée  $v_a(t)$  et de sortie  $\omega_m(t)$  ) sous la forme

$$\frac{K_0}{1 + \tau p}$$

Déduire des calculs analytiques effectués dans la section précédente les nouvelles expressions du gain statique, des zéros et des pôles de l'asservissement en fonction des gains  $\{k'_1, k'_2\}$ .

2. Dessiner les lieux des racines de l'asservissement lorsque  $k'_2$  est fixé à une valeur positive et  $k'_1$  varie de 0 à  $+\infty$ .
3. En déduire les effets des gains  $k'_1$  et  $k'_2$  sur les performances temporelles du système asservi. Indiquer intuitivement dans quelles zones du plan complexe la commande proposée – où les gains  $k'_1$  et  $k'_2$  sont supposés positifs – permet de placer les pôles du système asservi.
4. Calculer les gains  $k'_1$  et  $k'_2$  tels que la réponse indicielle de l'asservissement présente un dépassement de 25% ainsi qu'un temps de réponse de l'ordre de 1.25 s.
5. Calculer l'erreur de position de l'asservissement. Le type de commande mis en œuvre permet-il de satisfaire une spécification supplémentaire sur l'erreur de traînage?
6. Comparer le schéma de commande obtenu par la correction proportionnelle et tachymétrique. Sachant que les gains  $N$  et  $K$  sont réalisés par des amplificateurs analogiques, lequel des deux asservissements est le plus robuste du point de vue des performances en précision?

## 5. PROTOTYPAGE SUR LE PROCÉDÉ: IMPLÉMENTATION DES LOIS DE COMMANDE AVEC MATLAB-SIMULINK ET RTW TARGET

Dans cette partie, on se propose de vérifier les performances du système asservi de manière expérimentale, l'ensemble étant codé dans l'environnement Matlab-Simulink. Pour commander en temps réel le système, il faut utiliser la boîte à outils RTW TARGET.

1. Définir dans l'environnement de travail, une période d'échantillonnage  $T_e$  pour effectuer l'acquisition et l'envoi des données du moteur. Justifier votre choix avec l'aide de l'enseignant.
2. Pour la loi de commande par retour d'état, effectuer le schéma-bloc de l'asservissement, à partir du fichier copié du serveur. Vous devez le copier/coller du répertoire COMMUN (à partir de l'icône poste de travail, puis



connecter un lecteur réseau) dans un de vos répertoires de travail, dans la racine (Z:).

3. Compiler et exécuter votre schéma-bloc de l'asservissement (la démarche est décrite en Annexe). Observer et mesurer les performances de système asservi (caractéristiques dynamiques, erreurs...).
4. Comparer ces résultats réels aux résultats en simulation ainsi qu'à la théorie. Interpréter.
5. Pour toutes les autres lois de commande, suivre la démarche précédemment décrite. Observer et mesurer les performances de système asservi (caractéristiques dynamiques, erreurs...). Comparer ces résultats réels aux résultats en simulation ainsi qu'à la théorie. Interpréter.

## 6. ANNEXE: INTRODUCTION TO SIMULINK AND REAL-TIME WORKSHOP

---

SIMULINK is a powerful GUI simulation environment for the evaluation, prototyping and design of control systems using the MATLAB environment. Real-Time Workshop is an extension to the SIMULINK environment that allows you to compile executable code to run on embedded systems directly from SIMULINK GUI code.

The ensemble of programs consists of four principal components:

- **SIMULINK:** SIMULINK is an interactive tool for modelling, simulating, and analyzing dynamic systems. It enables you to build graphical block diagrams, evaluate system performance, and refine your designs.
- **Real-Time Workshop (RTW):** Real-Time Workshop generates optimized, portable, and customizable ANSI C code from Simulink models. It automatically builds programs that execute in real time or as stand-alone non real-time simulations. Generated code can run on PC hardware, DSPs or on micro-controllers on bare-board environments, and with commercial or proprietary real-time operating systems. The resulting code accelerates simulations on a wide variety of real-time rapid prototyping targets.
- **Real-Time Window Target (RTWT):** Real-Time Windows Target enables you to run Simulink and Stateflow models in real time on your PC. You can create and control a real-time execution through Simulink. You generate C code, compile it, and start real-time execution on Microsoft Windows while interfacing to real hardware using PC I/O boards. Other Windows applications continue to run during operation and can use all CPU cycles not needed by the real-time task.
- **I/O board:** Signals are passed between the real world and RTWT in real-time mode via a PC based data acquisition board. RT-Workshop supports a wide range of data acquisition boards. The laboratory is equipped with the DAQ6024E Devices.

### 6.1. Creating a Simulink model with DAQ6024E device

1. Start SIMULINK either by typing `simulink3` at the MATLAB command prompt or by double clicking on the appropriate icon.
2. Choosing components:  
At the Simulink Library Browser window, from File menu, choose Open and the file *identif.mdl*. Your workspace window titled *identif.mdl* will appear.
3. Connect components.
4. Save modifications.
5. Before running a RTWT model, it is necessary to compile and link the model.

- Build model: Select Tools menu, Real-Time Workshop, Build Model.  
The build function invokes the c-compiler and the log of the compile process is output to the MATLAB command window.  
If there are errors they are displayed in an error dialogue box.
  - Link Code: To link the code to the SIMULINK process choose external at the drop down menu in the SIMULINK window.  
Choose Simulation, Connect to Target.
  - Run Experiment: Select Simulation, Start real-time code. It will run for the time specified in the Simulation, Simulation parameters, Solver, Start time/Stop-time parameters.
6. Modifying parameters during a real-time experiment: When running a real-time model it is possible to change some parameters online without rebuilding the model.

MASTER ISTR-RODECO / E2 / SIA

---

# SLI 2 - I5

## TP 3: BACS D'EAU

---

ETUDE SIMULÉE DE LA COMMANDE PAR RETOUR DE SORTIE  
D'UN SYSTÈME DE BACS D'EAU

Année scolaire 2016-2017

Jour de TP:	
Date limite envoi CR:	
Enseignant(s):	
E-mail(s) enseignant(s):	

## 1. OBJET DE LA MANIPULATION

Le but de cette manipulation est de réaliser un asservissement de niveau par retour de sortie. En effet, le vecteur d'état du procédé hydraulique n'est pas entièrement accessible par la mesure. On se propose ainsi de mettre en oeuvre l'estimation de l'état par un observateur.

## 2. PRÉSENTATION DU SYSTÈME

Le procédé est le même que celui qui a été utilisé dans le TP du module I4 intitulé "Modélisation, analyse et commande d'une distribution hydraulique à trois bacs d'eau". Nous redonnons ici les caractéristiques essentielles de la manipulation.

### 2.1. Le procédé

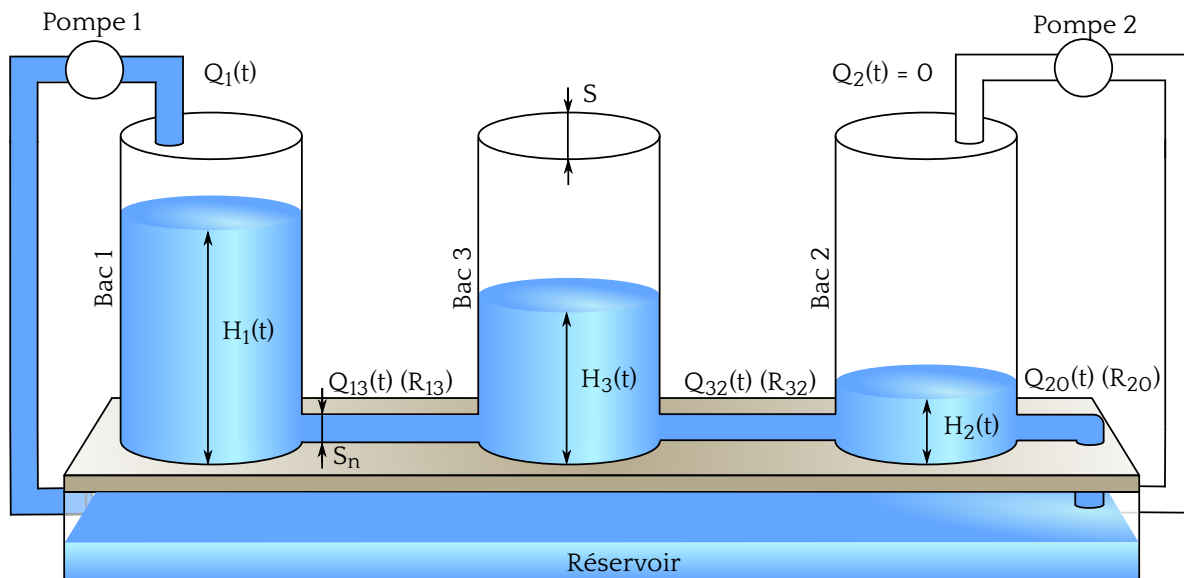


Figure 2.1: Procédé trois bacs.

Le système représenté sur la Figure 2.1 est composé de trois bacs cylindriques en plexiglas de section  $S$ . Ces trois bacs sont disposés en série (de gauche à droite, on trouve les bacs 1, 3 et 2) et sont reliés par des tuyaux d'écoulement de section  $S_n$ .

Le dernier bac 2 se vide par un cylindre, également de section  $S_n$ , dans le réservoir situé sous les bacs. Deux pompes de débit  $Q_1(t)$  et  $Q_2(t)$  permettent de remplir respectivement les bacs 1 et 2 avec l'eau récupérée dans le réservoir, le système fonctionnant en circuit fermé.

Les valeurs données par le constructeur sont  $S = 0,0154\text{m}^2$  et  $S_n = 5 \cdot 10^{-5}\text{m}^2$ .

Les pompes obéissent à la relation  $V_{q_i}(t) = k \cdot Q_i(t) + b$ ,  $i = 1, 2$ , avec  $k = 1,6 \cdot 10^5$  et  $b = -9,2592$ ; où  $V_{q_i}$  et  $Q_i(t)$  représentent respectivement la tension appliquée à la pompe  $i$  et le débit correspondant.

Les pompes sont alimentées par une tension comprise entre  $[-10\text{V}, 10\text{V}]$ . Ainsi le débit maximal,  $Q_{\max}$  délivrée par une pompe est  $12 \cdot 10^{-5}\text{m}^3/\text{s}$  lorsque la tension appliquée est de  $+10\text{V}$ . Les capteurs de niveau d'eau sont supposés linéaires autour du point de fonctionnement. Leur caractéristique est modélisé par l'équation  $H_i(t) = k_i \cdot V_{h_i}(t) + b_i$ , où  $H_i$  est exprimé en mètres.

## 2.2. Le modèle linéarisé

Nous considérons le procédé actionné par la seule pompe n°1. Son débit  $Q_1$  est compris entre  $[0, Q_{\max}]$  suivant sa tension d'alimentation; le débit  $Q_2(t)$  délivré par la pompe 2 sera nul tout au long de la manipulation. Ainsi, les différentes hauteurs  $H_1(t)$ ,  $H_3(t)$  et  $H_2(t)$  respectent par conséquent la condition  $H_1(t) \geq H_3(t) \geq H_2(t)$ . La seule mesure disponible lors de cette manipulation est la mesure de la hauteur d'eau  $H_1$ .

Le travail sera réalisé sur un modèle aux faibles variations autour d'un point d'équilibre  $H_0$  et d'un débit  $Q_{10}$  à ce point d'équilibre de telle sorte que:

$$\begin{cases} Q_1(t) &= q_1(t) + Q_{10} \\ H_1(t) &= h_1(t) + H_{10} \\ H_3(t) &= h_3(t) + H_{30} \\ H_2(t) &= h_2(t) + H_{20} \end{cases}$$

avec  $H_0 = [H_{10} \ H_{20} \ H_{30}]^T = [0.27474 \ 0.0299 \ 0.1368]^T$  et  $Q_{10} = 3,5 \cdot 10^{-5}$ .

Les variations de hauteurs  $h_1$ ,  $h_2$  et  $h_3$  vérifient la condition suivante:

$$h_1(t) \geq h_3(t) \geq h_2(t).$$

En rappelant que la sortie mesurée est la hauteur d'eau  $H_1(t)$ , ce qui permet d'obtenir la variation de hauteur d'eau  $h_1(t)$ , le modèle d'état linéarisé autour du point d'équilibre du procédé est donné par

$$\begin{cases} \dot{h}(t) &= \begin{bmatrix} -\frac{1}{SR_{13}} & \frac{1}{SR_{13}} & 0 \\ \frac{1}{SR_{13}} & -\frac{1}{S} \left( \frac{1}{R_{13}} + \frac{1}{R_{32}} \right) & \frac{1}{SR_{32}} \\ 0 & \frac{1}{SR_{32}} & -\frac{1}{S} \left( \frac{1}{R_{32}} + \frac{1}{R_{20}} \right) \end{bmatrix} h(t) + \begin{bmatrix} \frac{1}{S} \\ 0 \\ 0 \end{bmatrix} q_1(t) \\ y(t) &= [1 \ 0 \ 0]h(t) \end{cases}$$

avec  $h(t) = [h_1(t) \ h_2(t) \ h_3(t)]$ . De manière conventionnelle, les matrices dynamique, de commande et d'observation seront notées A, B, C.

On rappelle que  $R_{ij}$  représente la résistance à l'écoulement dans les tuyaux reliant les bacs i et j:

$$R_{ij} = \frac{2\sqrt{|H_{i0} - H_{j0}|}}{a_{ij}}$$

avec les trois coefficients d'écoulement :  $a_{13} = 0.4753 \cdot S_n \sqrt{2g}$ ,  $a_{32} = 0.4833 \cdot S_n \sqrt{2g}$  et  $a_{20} = 0.9142 \cdot S_n \sqrt{2g}$ .

## 3. ANALYSE ET CALCUL D'UNE LOI DE COMMANDE PAR RETOUR D'ÉTAT

1. Le modèle linéarisé est-il commandable? Justifiez votre réponse.

On considère pour cette étape de synthèse de la loi de commande que l'ensemble des états est mesuré. La loi de commande devra répondre aux spécifications du cahier des charges suivant:

- La consigne, notée  $w(t)$ , est un échelon de niveau exprimé en mètre. On pose  $w(t) = 0,05\mathcal{U}(t)$ ;
- le temps de réponse  $t_r$  de la sortie doit respecter  $t_r \leq 90$  s;
- l'erreur doit être nulle en régime permanent;
- le système doit rester dans le cadre d'une étude linéaire, c'est-à-dire que la vanne ne doit pas saturer:  $0 \leq Q_1(t) \leq Q_{\max}$ .

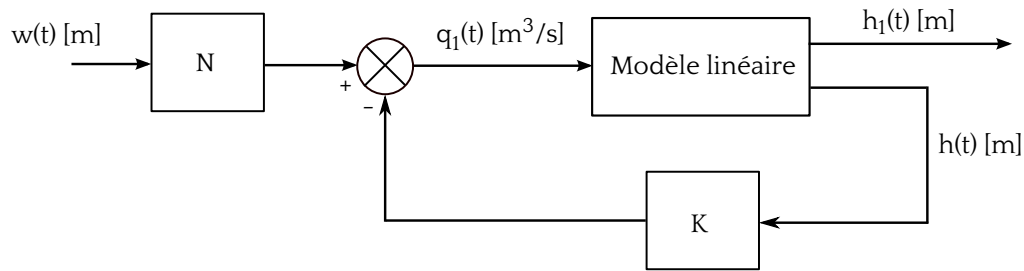


Figure 3.1: Retour d'état.

La loi de commande choisie est un retour d'état classique représenté sur la figure 3.1

2. Calculez les valeurs des gains  $K$  et  $N$  permettant de remplir l'ensemble des conditions.  
Donner les tracés de  $q_1(t)$  et de  $h_1(t)$  ainsi que le schéma bloc.

## 4. IMPLANTATION DE LA LOI DE COMMANDE

Nous considérons maintenant que seul l'accès à la mesure de  $h_1$  est possible. La mise en place d'un estimateur du vecteur d'état est donc nécessaire.

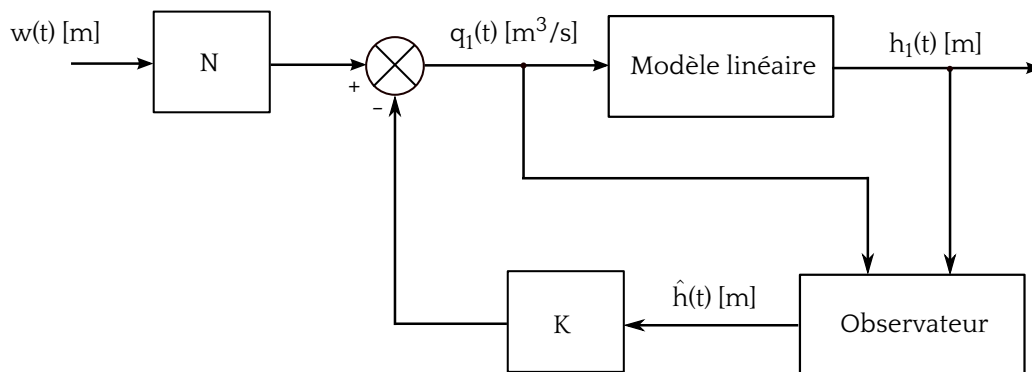


Figure 4.1: Observateur.

### 4.1. Calcul d'un observateur minimal identité

1. Le modèle linéarisé est-il observable? Justifiez votre réponse.
2. Calculer un observateur minimal identité admettant un temps de réponse 5 fois plus rapide que le système bouclé par le retour d'état calculé précédemment.
3. Insérer dans votre schéma Simulink l'observateur sous la forme d'un bloc State-Space.
4. Vérifier, en boucle ouverte, que les états estimés convergent vers les états réels du système linéarisé lorsque les hauteurs initiales sont non nulles.  
Evaluer la vitesse de convergence.
5. Réaliser le bouclage déterminé dans le paragraphe précédent en utilisant les états estimés (cf. figure 4.1).
6. Evaluer l'influence des conditions initiales de l'état de l'observateur sur les caractéristiques dynamiques de la boucle fermée (temps de réponse, dépassement, etc).  
Dédurre une règle générale sur l'influence de l'observateur sur le comportement entrée/sortie de l'asservissement.
7. Donner les 3 courbes suivantes:

- Schéma Simulink de l'asservissement par retour de sortie;
- Evolution de l'erreur d'estimation;
- Evolution de la consigne et de la sortie mesurée

## 4.2. Bruit sur la sortie mesurée

La mesure délivrée par le capteur de niveau du bac 1 est perturbée par un bruit sinusoïdal; la sortie mesurée est alors  $y(t) = Ch(t) + v(t)$  où  $v(t)$  est une sinusoïde d'amplitude 1 mm et de pulsation 10 rad/s;

8. Que constate-t-on sur la sortie du système et sur les états estimés pour le schéma pour le schéma de commande précédent?
9. Calculer un nouvel observateur de dynamique uniquement 2 fois plus rapide que le système bouclé par le retour d'état.  
Quel effet ce changement a-t-il sur la sortie du système et sur ses états estimés?
10. Expliquez le phénomène observé lors des deux dernières questions en vous basant sur une analyse fréquentielle des observateurs.

MASTER ISTR-RODECO / E2 / SIA

---

## SLI 2 - I5

### TP 4: MOTEUR II

---

SYNTHESE D'UN ASSERVISSEMENT DE POSITION PAR RETOUR D'ETAT  
CONTINU AVEC OBSERVATEUR

Année scolaire 2017-2018

Jour de TP:	
Date limite envoi CR:	
Enseignant(s):	
E-mail(s) enseignant(s):	



## 1. BUT DE LA MANIPULATION

Cette manipulation se propose de réaliser un asservissement de position en mettant en oeuvre les techniques d'espace d'état continu. Le procédé utilisé est la platine "asservissement de position", déjà présentée dans les textes des manipulations de licence EEA ainsi que du module Automatique Système Linéaire Invariant I et dont le schéma de principe est rappelé Figure 1.1.

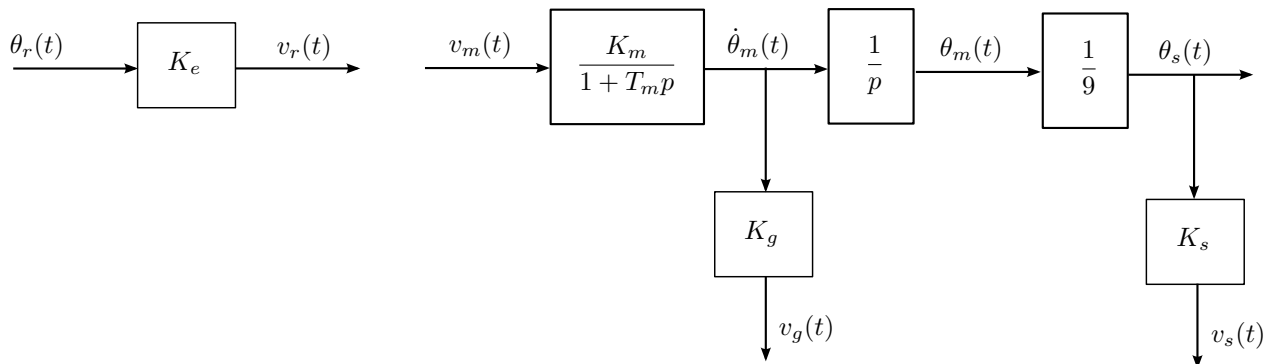


Figure 1.1: Schématisation de la platine "asservissement de position"

Les valeurs numériques des coefficients connus sont

$$K_e = 10 \text{ V.tr}^{-1}$$

$$K_s = 10 \text{ V.tr}^{-1}$$

$$K_g = 0.105 \text{ V.s.tr}^{-1}.$$

## 2. IDENTIFICATION ET ANALYSE DU PROCÉDÉ

1. Déterminer expérimentalement le gain en vitesse  $K_m$  et la constante de temps mécanique  $T_m$  du moteur à courant continu en indiquant le principe de la mesure ainsi que son mode opératoire.

On considère le système en boucle ouverte d'entrée  $v_m(t)$  et de sortie  $v_s(t)$ , que l'on désigne par  $S_{v_m \rightarrow v_s}$ .

2. Écrire la représentation d'état de ce système, lorsque son vecteur d'état est défini par:

$$\mathbf{x}(t) = (x_1(t), x_2(t))^T = (v_s(t), v_g(t))^T.$$

3. Rappeler les propriétés de commandabilité et d'observabilité de  $S_{v_m \rightarrow v_s}$ .  
Interpréter physiquement les résultats de cette analyse.

## 3. MISE EN PLACE D'UN RETOUR D'ÉTAT

1. Rappeler la justification de la possibilité d'annuler l'erreur de position de l'asservissement par un choix judicieux d'un gain de retour d'état  $K$  et de pré-compensation  $N$ .
2. Calculer les gains  $K$  et  $N$  permettant de placer les valeurs propres du système asservi en  $p_1, p_2 = -2.4 \pm 5.5j$  et d'annuler l'erreur statique.
3. Donner la représentation d'état de la boucle fermée  $S_{v_r \rightarrow v_s}$ .

## 4. COMPENSATION PAR RETOUR D'ÉTAT AVEC OBSERVATEUR IDENTITÉ ET MINIMAL IDENTITÉ

---

On envisage maintenant les cas où l'une des composantes du vecteur d'état est inaccessible ou bien trop bruitée pour être utilisée dans la commande, et on examine le problème de son estimation.

Cette section concerne la mise en place d'un observateur de l'état  $x_2(t) = v_g(t)$  du système  $S_{v_m \rightarrow v_s}$ .

Deux méthodes seront envisagées, la construction d'un observateur identité et la construction d'un observateur minimal identité. Le retour d'état proposé dans la section 3 sera donc effectué.

### 4.1. Observateur identité

Construction d'un observateur identité sur l'état estimé  $\hat{x}(t) = (\hat{x}_1(t), \hat{x}_2(t))^T$ , où  $\hat{x}_1(t)$ ,  $\hat{x}_2(t)$  sont les estimés de  $x_1(t)$ ,  $x_2(t)$  générés à partir de la seule connaissance de l'entrée  $v_m(t)$  et de la tension de sortie  $v_s(t) = K_s \theta_s(t)$ .

1. À partir de la représentation d'état du système  $S_{v_m \rightarrow v_s}$  compris entre  $v_m(t)$  et  $v_s(t)$  et du vecteur d'état  $x(t) = (x_1(t), x_2(t))^T = (v_s(t), v_g(t))^T$ , établir les équations d'un observateur identité permettant de reconstituer  $v_s(t)$ ,  $v_g(t)$ . Vous préciserez les dimensions de l'espace d'état associé à l'observateur.
2. Calculer les gains de l'observateur de manière à modifier ses valeurs propres pour que l'observateur soit 2 fois, 4 fois et 8 fois plus rapide que la dynamique de la boucle fermée.
3. Construire sous Simulink le schéma-bloc de asservissement. Les systèmes sous forme d'état seront représentés par de bloc `State-Space` de la librairie `Continuous` de Simulink. Ces blocs présentent l'avantage de permettre la spécification de conditions initiales. Visualiser sur Simulink les effets du choix des valeurs propres de l'observateur sur la vitesse de convergence de l'estimation pour un état initial du moteur non nul à savoir  $v_s(0) = \frac{\pi}{2}$  et  $\dot{v}_s(0) = 0$ .

### 4.2. Observateur minimal identité

Construction d'un observateur minimal identité sur l'état estimé  $\hat{x}(t) = (x_1(t), \hat{x}_2(t))^T$ , où  $\hat{x}_2(t)$  est l'estimé de  $x_2(t)$  généré à partir de la seule connaissance de l'entrée  $v_m(t)$  et de la tension de sortie  $v_s(t) = K_s \theta_s(t)$ .

4. À partir de la représentation d'état du système  $S_{v_m \rightarrow v_s}$  compris entre  $v_m(t)$  et  $v_s(t)$  et du vecteur d'état  $x(t) = (x_1(t), x_2(t))^T = (v_s(t), v_g(t))^T$ , établir les équations d'un observateur minimal identité permettant de reconstituer uniquement  $v_g(t)$ . Vous préciserez les dimensions de l'espace d'état associé à l'observateur.
5. Calculer les gains de l'observateur de manière à modifier ses valeurs propres pour que l'observateur soit 4 fois plus rapide que la dynamique de la boucle fermée. Justifiez le choix de vos valeurs propres.
6. Calculer les valeurs propres du système en boucle fermée.
7. Construire sous Simulink le schéma-bloc de asservissement. Les systèmes sous forme d'état seront représentés par le bloc `State-Space` de la librairie `Continuous` de Simulink. Ces blocs présentent l'avantage de permettre la spécification de conditions initiales. Visualiser sur Simulink les effets du choix des valeurs propres de l'observateur sur la vitesse de convergence de l'estimation pour un état initial du moteur non nul à savoir  $v_s(0) = \frac{\pi}{2}$  et  $\dot{v}_s(0) = 0$ .
8. Comparer les simulations obtenues avec celle obtenues en utilisant un observateur identité.

## 5. COMPENSATION PAR RETOUR DE SORTIE DYNAMIQUE – OBSERVATEUR FONCTIONNEL

---

Cette section est dédiée à la reconstruction via un observateur fonctionnel permettant de reconstituer la valeur du retour d'état.

1. À partir de la représentation d'état du système  $S_{v_m \rightarrow v_s}$  compris entre  $v_m(t)$  et  $v_s(t)$  et de vecteur d'état  $x(t) = (x_1(t), x_2(t))^T = (v_s(t), v_g(t))^T$ , établir les équations d'un observateur permettant de reconstituer  $Kx(t)$ , où  $K$  est la gain du correcteur par retour d'état calculé précédemment.
2. Calculer les gains de l'observateur de manière à modifier ses valeurs propres pour que l'observateur soit 8 fois plus rapide que la dynamique de la boucle fermée.
3. Calculer les valeurs propres du système en boucle fermée.
4. Construire sous Simulink le schéma-bloc de asservissement. Les systèmes sous forme d'état seront représenté par de bloc State-Space de la librairie Continuous de Simulink. Ces blocs présentent l'avantage de permettre la spécification de conditions initiales. Visualiser sur Simulink les effets du choix des valeurs propres de l'observateur sur la vitesse de convergence de l'estimation pour un état initial du moteur non nul à savoir  $v_s(0) = \frac{\pi}{2}$  et  $\dot{v}_s(0) = 0$ .

## 6. PROTOTYPAGE SUR LE PROCÉDÉ: IMPLÉMENTATION DES LOIS DE COMMANDE AVEC Matlab-Simulink et RTW TARGET

---

Dans cette partie, on se propose de vérifier les performances du système asservi de manière expérimentale. Afin de mettre en place la commande en temps réel de la manipulation, nous exploiterons l'environnement Matlab-Simulink et le moteur temps réel fourni par la boîte à outils Real-Time Workshop. La mise en place de cette démarche est expliqué en annexe.

1. Mettre place la lois de commande par retour d'état et par reconstruction du signal  $v_g(t)$ . L'observateur choisi sera celui qui a permis de converger le plus vite vers l'état du système lors des essais effectués dans la section 4.
2. Comparer les résultats obtenus avec ceux issus de la simulation. Quel est l'origine de ces observations?
3. Proposer et mettez en place une solution permettant de ne pas dégrader les performances initialement constatées lors des simulations.

## 7. ANNEXE: CRÉER ET COMPILER UN MODÈLE Simulink POUR LE PROTOTYPAGE DU BANC MOTEUR

---

1. Copier dans votre espace personnel le fichier `identif_ni.mdl` et l'ouvrir depuis Simulink.
2. Construire le schéma bloc d'asservissement en considérant le bloc bleu comme étant le moteur. Ce bloc est, en fait, composé de sous blocs d'émulation des entrées et sorties de la carte d'acquisition DAQ6024E de National Instrument.
3. Déclarer dans Matlab la période d'échantillonnage du processus temps réel:  $T_e = 0.05$ .
4. Compiler le modèle obtenu après l'avoir sauvegardé. À cet effet, vous sélectionnez la fonction *Build* du menu *Tools/Real-Time workshop*. En cas d'erreurs, une boîte de dialogue s'ouvrira.

5. Démarrer le processus temps réel en chargeant le code compilé sur la carte d'acquisition:
  - a) Choisir *External* dans le menu déroulant de la barre d'outils de la fenêtre du modèle.
  - b) Connecter le modèle à la cible en cliquant *Connect to target* dans le menu *Simulation*.
6. Lancer l'expérimentation avec la fonction *Start real-time code* du menu *Simulation*. Le temps d'expérimentation est spécifié par les paramètres de simulation *Start-time/Stop-time*.
7. REMARQUE: à chaque modification du schéma le modèle Simulink doit être recompilé. Cependant certains paramètres des blocs peuvent être modifié sans avoir à exécuter une nouvelle compilation.