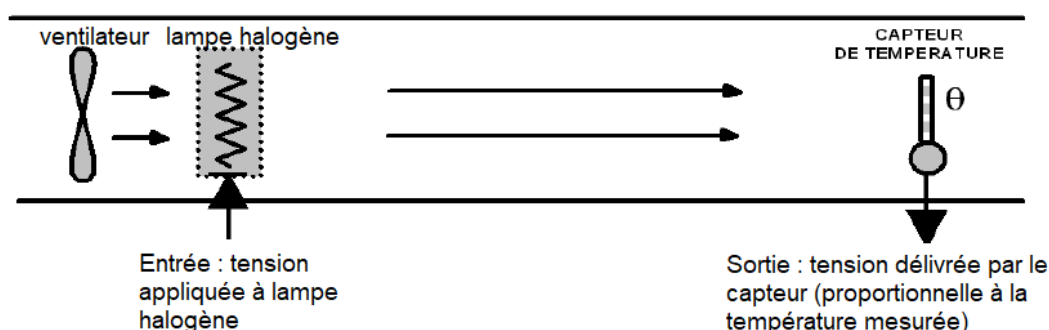


# IDENTIFICATION ET COMMANDE PAR RETOUR D'ÉTAT D'UN PROCESSUS THERMIQUE

## 1 Description du système et des données expérimentales

Le système étudié est une maquette de processus thermique composée :

- d'une lampe halogène : l'entrée du système est la tension appliquée à cette lampe (la chaleur dégagée est proportionnelle à cette tension).
- d'un capteur de température : la sortie du système est la différence de potentiel fournie par ce capteur (proportionnelle à la température mesurée)
- d'un ventilateur dont on peut régler l'intensité à partir d'une molette : ils permet de modifier le flux d'air entre la lampe et le capteur, c'est-à-dire la façon dont l'entrée va influencer sur la sortie, ou encore les paramètres du système.



On interagit avec ce système depuis un PC muni d'une carte d'entrées/sorties. Pour résumer, on se place dans le schéma classique de l'étude d'un système continu *via* un ordinateur représenté sur la Figure 1.

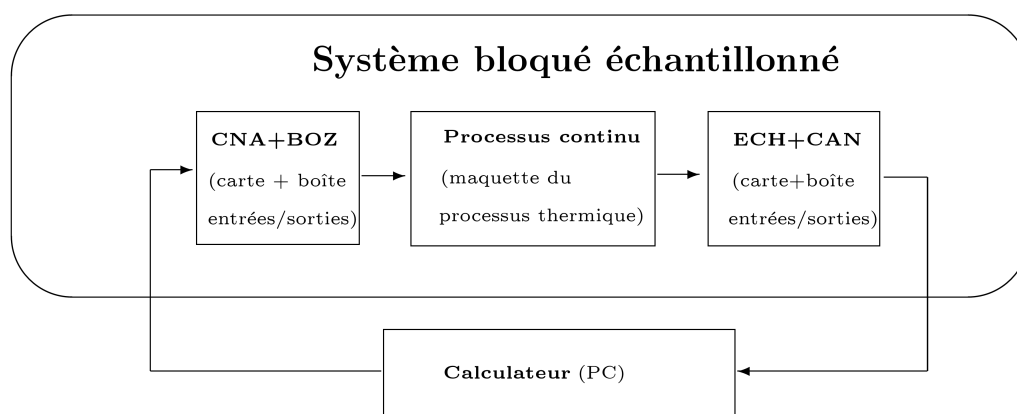


FIGURE 1 – Schéma d'étude du processus thermique *via* un ordinateur

Plusieurs expérimentations (excitation du système par un échelon ou un Signal Binaire Pseudo-Aléatoire et mesure de sa réponse) ont été réalisées avant la séance de TP et on utilisera les fichiers d'acquisitions récupérés lors de celles-ci. Plus précisément :

- Le fichier appelé `StepResponse.sod` contient les données mesurées sur le système lorsque celui-ci est excité à l'aide d'un échelon.
- Les fichiers appelés `PBRResponse1.sod` et `PBRResponse2.sod` contiennent les données mesurées sur le système lors de deux expérimentations où le système a été excité à l'aide d'un signal binaire pseudo-aléatoire (SBPA ou *PBR* pour

*Pseudo-Binary Random Signal*).

Chacun de ces fichiers contient trois vecteurs au format attendu par Scilab contenant les instants d'échantillonnage (vecteur nommé  $\mathbf{t}$ , en dixième de seconde), les échantillons de l'entrée (vecteur nommé  $\mathbf{u}$ , en  $V$ ) et les échantillons de la réponse en sortie (vecteur nommé  $\mathbf{y}$ , en  $V$ ). Ces données peuvent être chargées dans l'environnement de Scilab *via* la commande `load("NomFichier.sod")`.

## 2 Identification du système

### 2.1 Caractérisation

- Qu. 1 Charger les données de la réponse indicielle dans l'environnement de travail de Scilab et tracer la courbe représentant la réponse du système  $y$  à l'échelon en fonction du temps. Exploiter la courbe obtenue pour tenter de définir la structure de votre modèle : préciser (en justifiant) le type du système (ordre,...).
- Qu. 2 Utiliser votre script Scilab qui met en œuvre le test du rapport des déterminants instrumentaux en utilisant les données de `PBRResponse1.sod`. Confronter votre conclusion à la question précédente aux résultats obtenus avec ce script.
- Qu. 3 Proposer une structure du modèle du système échantillonné sous la forme d'une équation aux différences.

### 2.2 Estimation des paramètres

- Qu. 4 Utiliser votre script Scilab qui met en œuvre les moindres carrés ordinaires pour estimer les paramètres de votre équation aux différences en utilisant les données de `PBRResponse1.sod`.
- Qu. 4 Utiliser votre script Scilab qui met en œuvre les moindres carrés récursifs en utilisant les données de `PBRResponse1.sod` pour vous prononcer quant à la stationnarité du système durant cette expérimentation.

## 3 Commande par retour d'état du système

### 3.1 Obtention de la représentation d'état

Pour une fonction de transfert

$$H(z) = \frac{b_1 z + b_0}{z^2 + a_1 z + a_0},$$

on peut utiliser une représentation d'état sous la forme canonique de commande donnée par :

$$\begin{cases} x(k+1) &= \begin{pmatrix} 0 & 1 \\ -a_0 & -a_1 \end{pmatrix} x(k) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(k) \\ y(k) &= \begin{pmatrix} b_0 & b_1 \end{pmatrix} x(k) \end{cases}$$

On peut obtenir cette représentation d'état en introduisant une variable  $w$  comme suit

$$H(z) = \frac{Y(z)}{U(z)} = \frac{Y(z) W(z)}{W(z) U(z)} = \frac{b_1 z + b_0}{z^2 + a_1 z + a_0},$$

avec

$$\frac{Y(z)}{W(z)} = b_1 z + b_0, \quad \frac{W(z)}{U(z)} = \frac{1}{z^2 + a_1 z + a_0}$$

puis en faisant la transformée en  $Z$  inverse et en posant

$$x(k) = \begin{pmatrix} w(k) \\ w(k+1) \end{pmatrix}.$$

Notez que la variable  $w(k)$  est "abstraite" dans le sens où sa valeur ne reflète pas directement une grandeur physique au sein du système. Pour simplifier la suite, on considérera tout de même que les variables d'état  $w(k)$  et  $w(k+1)$  peuvent être mesurées sur le système.

Dans la suite, on utilise l'application Xcos avec Scilab (équivalent à Simulink avec Matlab) pour réaliser les expérimentations. Pour vous faire la main, construisez le modèle Xcos représenté sur la Figure 2 pour simuler la réponse indicielle du système. Paramétrer comme suit :

- saisir les matrices  $A$ ,  $B$ ,  $C$  et  $D$  ainsi que la valeur initiale du vecteur d'état dans le bloc de représentation d'état ;
- paramétrer le bloc Clock : Période=1 et Temps d'initialisation=0
- paramétrer le bloc Scope : Refresh Period=300, Ymin=-1, Ymax=3
- configurer le temps de simulation (Simulation>Configurer) à 300.

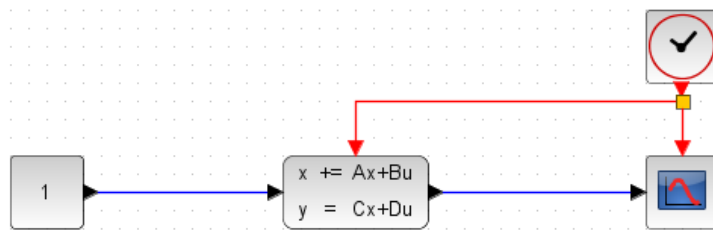


FIGURE 2 – Modèle Xcos pour simuler la réponse indicielle du modèle.

Qu. 5 Lancer la simulation et vérifier que vous obtenez une réponse indicielle similaire à celle tracée au début du TP.

On va commander le système à l'aide d'une commande par retour d'état. Plus précisément, on va appliquer le principe du placement de pôles, c'est-à-dire fixer une matrice de régulation  $L$  de façon à imposer les pôles du système bouclé, ou encore imposer le polynôme caractéristique du système corrigé  $|z\mathbf{I} - \mathbf{A} + \mathbf{B}\mathbf{L}|$  pour qu'il soit égal à

$$P_{des}(z) = z^2 + \alpha_1 z + \alpha_0.$$

Qu. 6 Vérifier que le système est commandable.

Qu. 7 Exprimer la matrice de régulation  $L$  (en fonction de  $a_0$ ,  $a_1$ ,  $\alpha_1$  et  $\alpha_0$ ).

Qu. 7 On souhaite en outre assurer un gain unitaire entre la consigne et la sortie. Déterminer la valeur du gain  $K_{re}$  à appliquer en amont de la boucle fermée en fonction de (en fonction de  $a_0$ ,  $a_1$ ,  $b_0$ ,  $b_1$ ,  $l_1$  et  $l_0$ ).

Pour une pulsation naturelle  $\omega_{nd}$  et un coefficient d'amortissement  $\xi_d$  (désirés pour notre système du deuxième ordre), les coefficients du polynôme caractéristique  $P_{des}(z) = z^2 + \alpha_1 z + \alpha_0$  sont donnés par

$$\alpha_0 = e^{-2\xi_d\omega_{nd}\Delta},$$

$$\alpha_1 = -2e^{-\xi_d\omega_{nd}\Delta} \cos(\omega_{nd}\sqrt{1 - \xi_d^2}\Delta).$$

avec ici  $\Delta = 1$ .

Qu. 8 Calculez à l'aide d'un script Scilab la matrice de régulation  $L$  et le gain  $K_{re}$  pour  $\omega_{nd} = 0.5$  et  $\xi_d = 0.9$ .

Qu. 8 Utilisez Xcos pour expérimenter la commande par retour d'état que vous avez élaborée. Pour cela, construisez le modèle Xcos représenté sur la figure 3 pour simuler le système équipé du retour d'état :

- Pour la boîte de représentation d'état en temps discret : précisez les matrices  $A$ ,  $B$  pour l'équation d'évolution et une matrice identité comme matrice  $C$  pour pouvoir disposer en sortie du vecteur d'état (attention alors aux dimensions de la matrice  $D$  nulle).
- Préciser  $C$ ,  $L$  et  $K_{re}$  sous la forme de gain.
- Configurer l'horloge avec les paramètres suivants : période = 1 et temps d'initialisation = 0.
- Configurer le temps d'intégration final (Simulation>Configurer) à 100.

Simuler le système corrigé. Comparer la réponse indicielle obtenue avec celle obtenue lors de la première simulation.

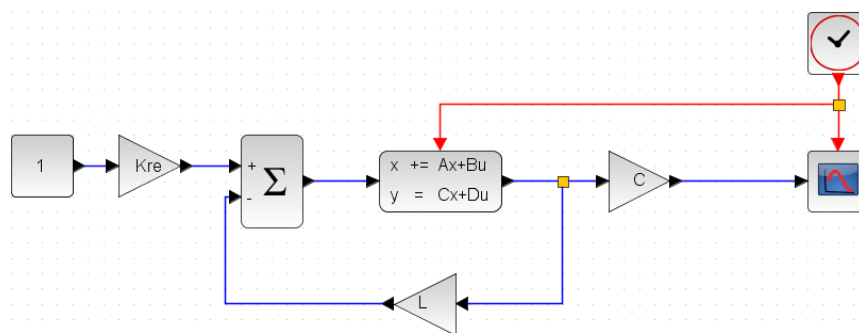


FIGURE 3 – Modèle Xcos pour simuler le système équipé du retour d'état