



TP2 : COMMANDE ET OBSERVATION MODALE

SYSTEME CONTINU

Objectifs :

- ◆ Modélisation et simulation d'un processus non linéaire sous Matlab – Simulink,
- ◆ Linéarisation au voisinage d'un point d'équilibre,
- ◆ Synthèse de loi de commande par retour d'état estimé.

Résultats : Les résultats obtenus seront consignés dans un compte rendu à l'issue des 2 séances de TP.

1 ANALYSE ET SIMULATION DE LA COMMANDE PAR RETOUR D'ETAT ESTIME D'UN PROCEDE CHIMIQUE

On considère un procédé chimique (Figure 1) constitué d'une cuve dans laquelle un produit est dissout par un solvant. Cette cuve est alimentée par deux arrivées, l'une contenant le produit de concentration constante c_1 et de débit réglable $f_1(t)$, l'autre, le solvant de concentration constante c_2 et de débit réglable $f_2(t)$. La solution résultante s'écoule par le fond de la cuve selon une concentration de produit $c(t)$ et un débit proportionnel à la racine carrée de la hauteur de solution dans la cuve :

$$f(t) = k \sqrt{\frac{V(t)}{S}} \quad (1-1)$$

où S est la section de la cuve, $V(t)$ le volume de solution dans la cuve et k un paramètre déterminé expérimentalement.

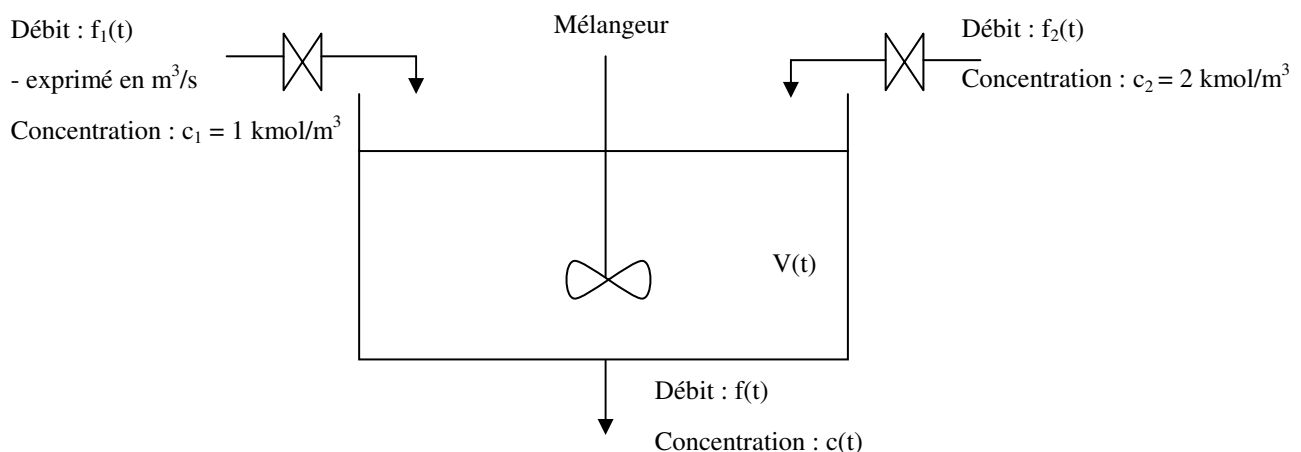


Figure 1: Mélangeur

L'objectif est de réguler autour d'un point d'équilibre du procédé à la fois le débit et la concentration de la solution en sortie de la cuve et ce en agissant sur le débit du solvant et du produit à traiter (grâce aux deux électrovannes).

1.1 Modélisation Non – Linéaire (NL)

1.1.1. Sous l'hypothèse naturelle d'incompressibilité des liquides, montrer que les équations de bilan volumique et de matière sont de la forme :

$$\begin{cases} \frac{dV(t)}{dt} = f_v(f_1(t), f_2(t), f(t)) \\ \frac{d(c(t)V(t))}{dt} = f_M(c_1, c_2, c(t), f_1(t), f_2(t), f(t)) \end{cases} \quad (1-2)$$

Rappel : Volume entrant = Volume accumulé + Volume sortant

Masse entrante = Masse accumulée + Masse sortante ⇔ Matière entrante = Mat accumulée + Mat sortante

1.1.2. Mettre le système (1-2) sous la forme du modèle d'état NL suivant :

$$\dot{\underline{x}}(t) = g(\underline{x}, \underline{u}) \quad (1-3)$$

où : $\underline{x}(t) = (V(t) \quad c(t))^T$

1.2 Modélisation Linéaire

1.2.1. Rechercher l'état d'équilibre, la sortie (concentration $c(t)$ et débit $f(t)$) étant régulée autour de l'équilibre c_e, f_e .

1.2.2. Montrer qu'au voisinage de cet état d'équilibre le comportement dynamique du système non linéaire est défini par le système linéarisé tangent suivant :

$$\begin{cases} \frac{\partial \dot{\underline{x}}}{\partial \underline{x}} = \begin{pmatrix} -\alpha & 0 \\ 0 & -2\alpha \end{pmatrix} \frac{\partial \underline{x}}{\partial \underline{x}} + \begin{pmatrix} 1 & 1 \\ \beta_1 & \beta_2 \end{pmatrix} \frac{\partial \underline{u}}{\partial \underline{u}} \\ \frac{\partial \underline{y}}{\partial \underline{x}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \frac{\partial \underline{x}}{\partial \underline{x}} \end{cases} \quad (1-4)$$

où $\underline{u} = \begin{pmatrix} f_1(t) \\ f_2(t) \end{pmatrix}$, $\alpha = \frac{f_e}{2V_e}$, $\beta_1 = \frac{c_1 - c_e}{V_e}$, $\beta_2 = \frac{c_2 - c_e}{V_e}$, $\frac{\partial \underline{x}}{\partial \underline{x}} = \underline{x} - \underline{x}_e$, $\frac{\partial \underline{u}}{\partial \underline{u}} = \underline{u} - \underline{u}_e$ et $\frac{\partial \underline{y}}{\partial \underline{y}} = \underline{y} - \underline{y}_e$ et le volume V et la concentration c sont mesurées.

1.2.3. Application numérique : l'état d'équilibre souhaité est :

$$\begin{pmatrix} f_{1e} = 15 * 10^{-3} \text{ m}^3 / \text{s} \\ f_{2e} = 5 * 10^{-3} \text{ m}^3 / \text{s} \\ V_e = 1 \text{ m}^3 \end{pmatrix} \quad (1-5)$$

D'après le modèle linéaire précédent, donner la représentation d'état autour de ce point d'équilibre et construire le modèle linéaire de simulation sous matlab et/ou simulink.

1.3 Commande par retour d'état

Vérifier la commandabilité du système linéaire (fonctions *ctrb* et *rank*).

Proposer une commande par retour d'état permettant la régulation du système autour de son point d'équilibre. On placera les pôles du système bouclé en -2 en utilisant la fonction Matlab (*place*).

Simuler le comportement du système bouclé avec cette commande lorsque le procédé est dans l'état initial :

$$V_0 = 2 \text{ m}^3; \quad C_0 = 1500 \text{ mol} \tag{1-6}$$

On pourra appliquer les fonctions Matlab (*initial* et/ou *lsim*).

1.4 Commande par retour d'état basé sur observateur identité

On suppose à présent que seul f et c sont mesurés. L'objectif est d'estimer l'état connaissant les mesures, la commande et le modèle. Ici la matrice C est inversible, de ce fait, l'estimé est directement donné par l'inverse de C . On demande cependant de construire l'observateur de façon classique (observateur identité) afin d'en illustrer ses forces et faiblesses.

1.4.1. Montrer après linéarisation autour du point d'équilibre précédent que $C = \begin{pmatrix} \alpha & 0 \\ 0 & 1 \end{pmatrix}$

1.4.2. Proposer à partir du modèle linéarisé tangent une loi de commande basée sur observateur permettant la régulation du système autour de son point d'équilibre défini par (1-5). Les pôles de l'observateur seront placés en -5 et ceux du contrôleur en -2 . Pourquoi?

1.4.3. Simuler le comportement du système bouclé avec cette commande lorsque le procédé est dans l'état initial donné (1-6). On pourra notamment regarder l'allure de l'erreur de sortie et de l'erreur faite sur l'estimation d'état.

1.4.4. Analyser l'influence des pôles de l'observateur et l'influence de l'initialisation de l'état estimé

1.5 Tracé des fonctions S (sensibilité) et T (sensibilité complémentaire) avec observateur

```
SysK=ltisys(A-G*C-B*K,G,-K,zeros(2,2)) ;
```

```
SysG= ltisys(A,B,C,zeros(2,2)) ;
```

```
Loop=smult(SysK, SysG);
```

```
[AL,BL,CL,DL]=ltiss(Loop);
```

```
loop_=ss(AL,BL,CL,DL)
```

```
figure(1)
```

```
S=inv(eye(2)+loop_);
```

```
sigma(S);
```

```
figure(2)
```

```
T=series(loop_,S);
```

```
sigma(T);
```

```
figure(3)
```

```
sigma(loop_);
```

Analyser l'influence des pôles de l'observateur et du régulateur

1.6 Tracé des fonctions S (sensibilité) et T (sensibilité complémentaire) sans observateur

Que remarquez vous ?

1.7 Simulation Non linéaire

1.7.1. Construire le modèle non-linéaire de simulation sous matlab et/ou simulink

On pourra appliquer une S_function (bloc melange sous simulink). Cette S_function melange sera décrite **sous matlab** et enregistrée sous le même nom "melange.m" :

```
function [sys, x0] = melange (t, x, u, flag)
    % définition de la S-fonction
    k=0.03; fe=20*10^(-3); Ve=1; c1=1; c2=2; S=k*k*Ve*inv(fe*fe);
    Si flag == 0 alors initialisation de sys et x0
    % sys = [nb état continu; nb état discret; nb de sortie; nb d'entrées; nb de commande direct; nb de période d'éch]
    sys=
    x0=
    Sinon si flag == 1 alors
    % calcul de dx/dt
    sys =[u(1)+u(2)- k*sqrt(x(1)/S)]; eq diff2] ;
    Sinon si flag == 3 alors
    % calcul des sorties y
    sys=[k*sqrt(x(1)/S); x(2)];
end
```

Sous simulink ouvrir une boîte S_function qui porte le nom de la fonction melange et l'enregistrer sous un nom différent exemple S_melang

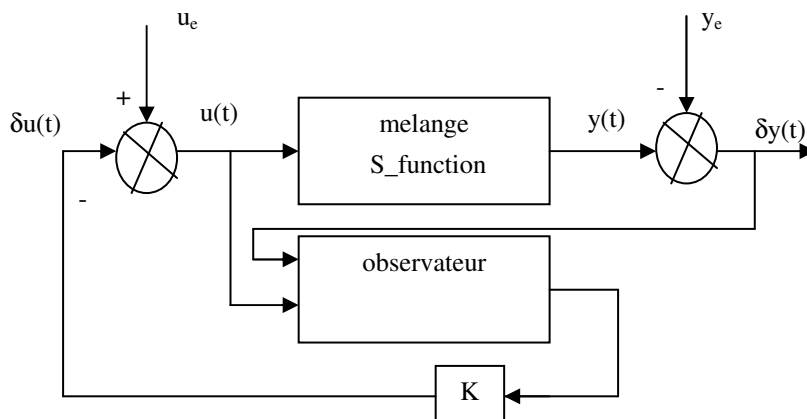


Figure 2: Simulation du modèle non-linéaire associé au Reg/Obs synthétisé sur le modèle linéaire

1.7.2. Sous simulink, remplacer le modèle linéaire par le simulateur non-linéaire (S_function) et analyser les résultats du régulateur et de l'observateur, ceux-ci étant synthétisés à partir du modèle linéaire

On pourra tester pour différents états initiaux du procédé et de l'observateur, $x_0 = (v_0 \quad c_0)^T$ et $\delta \hat{x}_0 = \hat{x}_0 - x_0$.