**UMMTO, FGEI, Département Automatique, Année Universitaire : 2019−2020**
**Spécialité : Master 2 Académique, Options : AS et AII.**
**Module : Automatique Avancée, Travaux pratiques**

## TP 1. Simulation d'une perturbation interne (variation paramétrique)

Le comportement dynamique d'un système est décrit par le modèle suivant :

$$Y(s) = \frac{K\,e^{-\theta\,s}}{T\,s+1}\,U(s) + \frac{10}{20\,s+1}\,P(s) \tag{1}$$

$Y$, $U$, $P$ sont respectivement la sortie, la commande et la perturbation. $K$, $T$ et $\theta$ sont les paramètres du système qui désignent respectivement le gain, la constante du temps et le retard.

Pour la simulation, il est demandé de prendre un temps final de simulation est égale à 400 s et le solveur de Matlab doit être paramétré avec un pas fixe de l'ordre de 0, 01 s.

### Partie 1 : Poursuite et régulation

On suppose que les paramètres du système sont constants avec $K = 1$, $T = 20\,\mathrm{s}$ et $\theta = 5\,\mathrm{s}$. On désire corriger ce système en utilisant un correcteur du type PI de fonction de transfert

$$C(s) = K_c\left(1 + \frac{1}{T_i\,s}\right) \tag{2}$$

$K_c$ et $T_i$ sont respectivement le gain et la constante intégrale du correcteur.
Pour le réglage du PI, on propose d'utiliser la méthode de **Cohen-Coon** qui conduit au réglage suivant :

$$K_c = \frac{T}{K\,\theta}\left(\frac{10,8\,T+\theta}{12\,T}\right) \qquad T_i = \theta\left(\frac{30\,T+3\,\theta}{9\,T+20\,\theta}\right) \tag{3}$$

**1.** Calculer les paramètres du correcteur,

**2.** Pour évaluer, par simulation en utilisant Simulink, les performances du système en boucle fermée, on demande d'imposer à $t = 0\,\mathrm{s}$ une consigne désirée du type échelon d'amplitude 60 suivie d'une perturbation de même nature et d'amplitude 3 à $t = 200\,\mathrm{s}$. Commenter le résultat obtenu.

### Partie 2 : Simulation d'une perturbation interne (variation paramétrique)

Pour cette partie, on considère une perturbation nulle. On demande d'imposer à $t = 0\,\mathrm{s}$ une consigne désirée du type échelon d'amplitude 60 et de réaliser à $t = 200\,\mathrm{s}$ les variations paramétriques données dans les Tables **1** et **2**.

| Test | I | II |
|---|---|---|
| Variation de $T$ | −75% | 100% |
| Variation de $K$ | 0% | 0% |

Table 1 – Variations du paramètre $T$.

| Test | I | II | III |
|---|---|---|---|
| Variation de $T$ | 0% | 0% | 0% |
| Variation de $K$ | −50% | 30% | 50% |

Table 2 – Variations du paramètre $K$.

Tirer des conclusions sur l'influence de chaque paramètre sur les performances du système corrigé.

**TP 2. Évaluation des performances d'une commande adaptative indirecte**

On désire concevoir une commande adaptative indirecte pour le système à paramètres $a$ et $b$ variables dans le temps décrit par la fonction de transfert suivante :

$$G(s) = \frac{a}{s+b}, \quad a \neq 0 \text{ et } b \neq 0 \tag{4}$$

en utilisant un correcteur du type PI discret dont la fonction de transfert est :

$$C(z) = k_c \left(1 + \frac{T_e}{T_i\,(z-1)}\right) \tag{5}$$

Le réglage des paramètres du correcteur $C(z)$ est réalisé par placement de pôles.

Les équations aux différences du système $G(s)$, muni d'un bloqueur d'ordre 0, et du correcteur PI discret sont données respectivement comme suit :

$$y(k+1) = c\,y(k) + f\,u(k) \tag{6}$$

$$u(k) = u(k-1) + k_c\,e(k) + k_c\left(\frac{T_e}{T_i} - 1\right) e(k-1) \tag{7}$$

avec

$$c = e^{-b\,T_e} \text{ et } f = \frac{a\,(1-c)}{b} \tag{8}$$

et l'erreur de poursuite $e(k) = y^d(k) - y(k)$.

Les relations liant les paramètres du correcteur PI discret et les estimés des paramètres du modèle $\hat{c}$ et $\hat{f}$ sont donnés comme suit :

$$k_c = \frac{1 + \hat{c} - (z_1 + z_2)}{\hat{f}} \text{ et } T_i = \frac{T_e\,(1 + \hat{c} - (z_1 + z_2))}{(z_1 - 1)\,(z_2 - 1)} \tag{9}$$

où $z_1$ et $z_2$ sont les pôles désirés en boucle fermée.

**Partie 1 : Simulation du système en boucle ouverte et le système corrigé**

En considérant les paramètres constants, avec $a = 1$ et $b = 0,5$, on demande d'écrire le programme permettant :

**1.** de simuler le système en boucle ouverte pour une entrée du type échelon unité $u(k) = 1$,

**2.** de simuler le système en boucle fermée pour une consigne désirée $y^d(k) = 1$,

**3.** de représenter les évolutions de la consigne $y^d(k)$ et de la sortie $y(k)$ du système non corrigé (boucle ouverte) et du système corrigé (boucle fermée). Conclure.

Pour la simulation, on considère une période d'échantillonnage $T_e = 1$ et les pôles désirés suivants : $z_1 = 0,25 + j\,0,25$ et $z_1 = 0,25 - j\,0,25$

**Partie 2 : Système corrigé avec adaptation des paramètres du correcteur**

On considère maintenant que les paramètres $a$ et $b$ sont variables dans le temps comme suit :

$$a(k) = a(k-1) + \alpha\,\varepsilon_a(k-1), \quad b(k) = b(k-1) + \beta\,\varepsilon_b(k-1) \text{ avec } \alpha,\, \beta \geq 0 \tag{10}$$

où $\varepsilon_a$ et $\varepsilon_b$ sont des variables aléatoires (à générer avec la fonction **randn**).

Pour l'estimation des paramètres $c$ et $f$ à l'instant $k = 0$, on utilise la méthode des moindres carrés en considérant les mesures du Tableau **3**.

| $k$ | $-4$ | $-3$ | $-2$ | $-1$ | $0$ |
|---|---|---|---|---|---|
| $u(k)$ | 1 | 1 | 1 | 1 | ? |
| $y(k)$ | 0 | 0,75 | 1,25 | 1,50 | 1,70 |

TABLE 3 – Mesures entrée-sortie.

et pour $k \geq 1$, on utilise la méthode des moindres carrés récursifs.

Pour la simulation, on considère $\alpha = \beta = 0,05$.

**1.** donner l'organigramme de la stratégie de commande,

**2.** écrire le programme permettant de simuler la stratégie de commande adaptative indirecte, de représenter l'évolution de la sortie $y(k)$ (avec la consigne $y^d(k)$), les variations des paramètres du correcteur $k_c$ et $T_i$, les variations des paramètres du modèle ($c$ et $f$) et leurs estimées ($\hat{c}$ et $\hat{f}$),

**3.** étudier l'influence de $\alpha$ et de $\beta$ sur les performances en boucle fermée en réalisant des tests avec les valeurs du Tableau **4**.

| Test | I | II | III | IV |
|---|---|---|---|---|
| $\alpha$ | $0,025$ | $0,050$ | $0,025$ | $0$ |
| $\beta$ | $0,050$ | $0,025$ | $0,025$ | $0$ |

TABLE 4 – Paramètres $\alpha$ et $\beta$.

## TP 3. Évaluation des performances d'une commande adaptative directe

### Partie 1 : Méthodes de Lyapounov et du gradient (règle MIT)

Soit le système dynamique du second ordre, à deux paramètres $a$ et $b$ variables dans le temps, suivant :

$$\ddot{y}(t) = -a\,\dot{y}(t) - b\,y(t) + u(t) \tag{11}$$

où $y(t)$ et $u(t)$ sont respectivement la sortie et la commande du système.

Notre objectif est d'imposer à la sortie $y(t)$ une trajectoire $y_m(t)$ définie par le modèle de référence suivant :

$$\ddot{y}_m(t) = -2\,\dot{y}_m(t) - y_m(t) + y^d(t) \tag{12}$$

où $y_m(t)$ et $y^d(t)$ sont respectivement la sortie de référence et la consigne désirée.

Pour concrétiser cet objectif, on propose d'utiliser un retour d'état

$$u(t) = k_1(t)\,\dot{y}(t) + k_2(t)\,y(t) + y^d(t) \tag{13}$$

Les lois d'adaptation des gains du retour d'état $k_1(t)$ et $k_2(t)$, déterminées en utilisant les méthodes de la fonction **Lyapounov** et du *gradient* (règle **MIT**), sont résumées dans le Tableau 5.

| Méthode de la fonction **Lyapounov** | Méthode du gradient (Règle **MIT**) |
|---|---|
| $\dot{k}_1(t) = -\dot{e}(t)\,\dot{y}(t)$<br>$\dot{k}_2(t) = -\dot{e}(t)\,y(t)$ | $\dot{k}_1(t) = -\lambda\,e(t)\,F(s)\,\dot{y}(t)$<br>$\dot{k}_2(t) = -\lambda\,e(t)\,F(s)\,y(t)$ |

TABLE 5 – Lois d'adaptation.

où $\lambda$ est un paramètre positif (pas de descente), $e(t)$ est l'erreur de poursuite ($e(t) = y(t) - y_m(t)$) et $F(s)$ est le filtre de sensibilité dont la fonction de transfert est

$$F(s) = \frac{1}{s^2 + 2\,s + 1} \tag{14}$$

**1.** réaliser le schéma de simulation de chaque stratégie de commande,

**2.** étudier l'influence de la condition initiale $k(0)$ sur les performances de chaque stratégie de commande en considérant les différents cas du Tableau 6,

| Cas | I | II | III | IV | V |
|---|---|---|---|---|---|
| $k(0) = \begin{bmatrix} k_1(0) \\ k_2(0) \end{bmatrix}$ | $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$ | $\begin{bmatrix} 1.00 \\ 0.50 \end{bmatrix}$ | $\begin{bmatrix} 0.50 \\ 1.00 \end{bmatrix}$ | $\begin{bmatrix} -0.50 \\ 1.00 \end{bmatrix}$ | $\begin{bmatrix} -1.00 \\ -0.50 \end{bmatrix}$ |

TABLE 6 – Condition initiale $k(0)$.

**3.** étudier l'influence du paramètre $\lambda$ en considérant les valeurs du Tableau 7.

| Cas | I | II | III | IV |
|---|---|---|---|---|
| $\lambda$ | 0.001 | 0.01 | $0,5$ | 1 |

TABLE 7 – Valeurs du pas de descente $\lambda$.

**Indications :**

— les conditions initiales du système et du modèle de référence sont nulles ($y(0) = \dot{y}(0) = y_m(0) = \dot{y}_m(0)$),

— comme l'estimation paramétrique dépend de la nature du signal d'excitation, on considère pour la stratégie de commande basée sur la fonction de Lyapounov $y^d(t) = \sin(t)$ et pour la stratégie basée sur la règle MIT $y^d(t) = 1$,

**Partie 2 : Commande à modèle de référence adaptative**

Pour imposer, en boucle fermée, la dynamique du modèle de référence suivant :

$$G_m(s) = \frac{y_m(s)}{y^d(s)} = \frac{s+1}{s^2+2\,s+2} \tag{15}$$

pour le système dynamique instable suivant :

$$G(s) = \frac{y(s)}{u(s)} = \frac{2\,s+4}{s^2+s-6} \tag{16}$$

où $y_m(t)$, $y^d(t)$, $y(t)$ et $u(t)$ sont respectivement la sortie du modèle de référence, la consigne désirée, la sortie du système et la commande du système.

Pour ce faire, on utilise une commande à modèle de référence adaptative qui conduit à la loi de commande suivante :

$$u(s) = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \begin{bmatrix} \dfrac{u(s)}{s+1} \\ \dfrac{y(s)}{s+1} \\ y(s) \\ y^d(s) \end{bmatrix} \tag{17}$$

avec la loi d'adaptation suivante :

$$\begin{bmatrix} \dot{k}_1(t) \\ \dot{k}_2(t) \\ \dot{k}_3(t) \\ \dot{k}_4(t) \end{bmatrix} = -e(t)\,\Gamma \begin{bmatrix} \dfrac{u(s)}{s+1} \\ \dfrac{y(s)}{s+1} \\ y(s) \\ y^d(s) \end{bmatrix} \tag{18}$$

avec $e(t) = y(t) - y_m(t)$ et $\Gamma = \Gamma^T > 0$ (matrice symétrique définie positive).

**1.** on considère une matrice de pondération $\Gamma = I_{4\times4}$ (matrice identité de dimensions $4\times4$), étudier l'influence des conditions initiales en considérant les différents cas résumés dans le Tableau 8,

| Cas | I | II | III | IV |
|---|---|---|---|---|
| $k(0) = \begin{bmatrix} k_1(0) \\ k_2(0) \\ k_3(0) \\ k_4(0) \end{bmatrix}$ | $\begin{bmatrix} 0.25 \\ 0.50 \\ 1.00 \\ 1.50 \end{bmatrix}$ | $\begin{bmatrix} 0.50 \\ 0.50 \\ 1.00 \\ 1.50 \end{bmatrix}$ | $\begin{bmatrix} 0.25 \\ 0.50 \\ 1.25 \\ 1.50 \end{bmatrix}$ | $\begin{bmatrix} -0.25 \\ 1.50 \\ 0.75 \\ -1.00 \end{bmatrix}$ |

TABLE 8 – Condition initiale $k(0)$.

**2.** on considère la condition initiale

$$k(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{19}$$

étudier l'influence de la matrice de pondération $\Gamma$ sur les performances en boucle fermée en considérant les cas du Tableau 9.

| Cas | I | II | III | IV |
|---|---|---|---|---|
| $\Gamma$ | $\begin{bmatrix} 7 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 6 \end{bmatrix}$ |

TABLE 9 – Matrice de pondération $\Gamma$.

**TP 4. Commande Dynamique Matricielle d'un système**

On désire concevoir une Commande Dynamique Matricielle (DMC) pour le système suivant :

$$G(z) = \frac{y(z)}{u(z)} = \frac{0,2713}{z - 0,8351} \tag{20}$$

La période d'échantillonnage $\Delta t = 2$.

Rappelons que la commande $u(t)$ à appliquer à l'instant d'échantillonnage $t$ est :

$$u(t) = u(t-1) + \Delta u(t) \tag{21}$$

avec

$$\Delta u(t) = \left[G^T\, Q\, G + R\right]^{-1}\, G^T\, Q\, \left(y_f^d(t) - F\, X(t)\right) \tag{22}$$

et

$$G = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ g_3 & g_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ g_{N_p} & g_{N_p-1} & \cdots & g_{N_p-N_u+1} \end{bmatrix}, \quad F = \begin{bmatrix} 1 & g_2 - g_1 & g_3 - g_2 & \cdots & g_{1+N_s} - g_{N_s} \\ 1 & g_3 - g_1 & g_4 - g_2 & \cdots & g_{2+N_s} - g_{N_s} \\ 1 & g_4 - g_1 & g_5 - g_2 & \cdots & g_{4+N_s} - g_{N_s} \\ \vdots & \vdots & & & \vdots \\ 1 & g_{N_p+1} - g_1 & g_{N_p+2} - g_2 & \cdots & g_{N_p+N_s} - g_{N_s} \end{bmatrix} \tag{23}$$

$$X(t) = \begin{bmatrix} y(t) & \Delta u(t-1) & \Delta u(t-2) & \cdots & \Delta u(t-N_s) \end{bmatrix}^T \tag{24}$$

$$y_f^d(t) = \alpha\, y_f^d(t-1) + (1-\alpha)\, y^d(t) \tag{25}$$

avec $\alpha$ est un paramètre qui permet de fixer la dynamique de la consigne filtrée.

Pour écrire le programme de la commande dynamique matricielle, on propose de procéder par étape. On demande de représenter, sur la même figure, l'évolution de la sortie commandée $y(t)$, de la consigne désirée $y^d(t)$ et la consigne filtrée $y_f^d(t)$, et sur une autre figure l'évolution de la commande $u(t)$.

**1.** générer la réponse indicielle $y(t)$ pour une entrée du type échelon unité,

**2.** choisir une valeur pour $N_s$ et générer les deux matrices $G$ et $F$ pour des valeurs de $N_p$ (horizon de prédiction) et $N_u$ (horizon de commande) données ($N_u < N_p < N_s$),

**3.** générer et appliquer au système la commande $u(t)$ en considérant les valeurs suivantes : $N_u = 5$, $N_p = 10$, $\alpha = 1$, $Q = I_{N_p \times N_p}$ et $R = I_{N_u \times N_u}$.

**Partie 1 : Etude de l'influence de $\alpha$ (filtrage de la consigne désirée $y^d(t)$)**

On prend $N_u = 5$ et $N_p = 10$, on demande d'étudier l'influence du paramètre $\alpha$ sur $y(t)$ et $u(t)$ en considérant les valeurs suivantes : $0,25$ ; $0,5$ et $0,75$.

**Partie 2 : Etude de l'influence de l'horizon de commande $N_u$**

On prend $N_p = 20$ et $\alpha = 0$, on demande d'étudier l'influence de l'horizon de commande $N_u$ sur $y(t)$ et $u(t)$ en considérant pour $N_u$ les valeurs suivantes : 5, 10 et 15.

**Partie 3 : Etude de l'influence de l'horizon de prédiction $N_p$**

On prend $N_u = 5$ et $\alpha = 0$, on demande d'étudier l'influence de l'horizon de prédiction $N_P$ sur $y(t)$ et $u(t)$ en considérant les valeurs suivantes : 10, 15 et 20.

**Programme DMC**

```
clc
clear
close all

%pkg load control

dt = 2;
ts = 300;

% Réponse indicielle
gz = tf(0.2713, [1, -0.8351], dt)
yind = step(gz, ts);
figure(1)
stairs(yind(1:end-1));
grid

yind_G = yind(2:end);

Ns = 40; % instant d'échantillonnage à partie dequel
Np = 10; % Horizon de prédiction Nu<Np
Nu = 2; % Horizon de commande

% Génération de la matrice G

for k = 1:Nu
G(:, k) = [zeros(k-1, 1);yind_G(1:Np-k+1)];
end

% Génération de la matrice F

for i = 1: Np;
for j = 1 : Ns
F(i,j) = yind_G(i+j)-yind_G(j);
end
end

Q = eye(Np);
R = 1;
F = [ones(Np,1) F];

% Calcul du gain de la commande
K = inv(G'*Q*G + R*eye(Nu))*G'*Q;

u = zeros(Ns,1);
y = zeros(Ns+1,1);

delta_u = zeros(Ns,1);

Hs = 3*Ns;
alpha = 0;
yd =  zeros(Ns,1);
ydf = zeros(Ns,1);

for k = Ns+1:Hs
X = [y(k); delta_u(k-1:-1:k-Ns)];
f = F*X;
yd(k) = 1;
```

```
ydf(k) = alpha*ydf(k-1)+(1-alpha)*yd(k);
delta_u(k) = K(1,:)*(ydf(k)-f);
u(k) = u(k-1) + delta_u(k);
if k <= 2*Ns
d = 0;
else
d = 0.1;
end
y(k+1) = 0.8351*y(k) + 0.2713*u(k) + d;
end

figure(2)
stairs(yd(Ns:end-1),'r');
hold on
plot(ydf(Ns:end-1),'g');
hold on
plot(y(Ns+1:end-1),'b')
hold on
plot(yind(1:(Hs-Ns)),'k')
hold off
grid
xlabel('t')
ylabel('y(t), y^d(t), y^d_f(t)')

figure(3)
stairs(u(Ns:end-1))
grid
xlabel('t')
ylabel('u(t)')
```

**TP 5. Commande prédictive généralisée**

Écrire le programme qui permet de simuler la commande prédictive généralisée de l'exemple du cours.

Le programme ci-joint ne calcule pas les matrice $G^*$ et $X$ automatiquement. Il est intéressant d'écrire le programme qui calcule directement ces matrices et la loi de commande (on peut s'inspirer du programme de la commande DMC ) .

**Programme GPC**

```
clc
clear



N = 43;
ym = zeros(1, N);
u = zeros(1, N);
ym_d = ones(1, N);

for k = 3:N;
% Calcul de la commande u
% Commande GPC dans le cas d'un échelon Unité.
% ym^d(t+1) = ym^d(t+2) = ym^d(t+3)
u(k) = 0.396*u(k-1)+0.604*u(k-2)-1.371*ym(k) + 0.805*ym(k-1)+0.566*ym_d(k);
% Notons que : 0.566 = 0.133 + 0.286 + 0.147 (somme des trois éléments derniers de
% de la loi de commande pusique la consigne est un échelon ym^d(k) =ym^d(t+1) = ym^d(t+2) = ym^d(t+3)

if k >= 20
d = -0.1;
else
d = 0;
end

% Application de la commande
ym(k+1) = 0.8*ym(k)+0.4*u(k)+0.6*u(k-1)+d;

end

% Résultats

figure(1)
plot(0:N-3, ym(3:N),'r','linewidth',2)
hold on
stairs(0:N-3,ym_d(3:N),'b','linewidth',2)
hold off
grid
set(gca, "fontsize", 20)
legend('y(k)','y_m^d(k)',"location",'southeast')
xlabel('k','fontsize', 20)
ylabel('y_m(k), y_m^d(k)', 'fontsize', 20)

figure(2)
stairs(0:N-3,u(3:N),'linewidth',2)
grid
set(gca, "fontsize", 20)
xlabel('k', 'fontsize', 20)
ylabel('u(k)', 'fontsize', 20)
```