

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

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

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

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

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

$$u(t) = u(t-1) + \Delta u(t) \quad (21)$$

avec

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

et

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

$$X(t) = [ y(t) \quad \Delta u(t-1) \quad \Delta u(t-2) \quad \cdots \quad \Delta u(t-N_s) ]^T \quad (24)$$

$$y_f^d(t) = \alpha y_f^d(t-1) + (1-\alpha) y^d(t) \quad (25)$$

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 partir 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)')
```

@copyright ahmed maidi 2019-2020

## 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 matrices  $G^*$  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.
%  $y_m^d(t+1) = y_m^d(t+2) = y_m^d(t+3)$ 
u(k) = 0.396*u(k-1)+0.604*u(k-2)-1.371*y_m(k) + 0.805*y_m(k-1)+0.566*y_m_d(k);
% Notons que : 0.566 = 0.133 + 0.286 + 0.147 (somme des trois éléments derniers de
% de la loi de commande puisque la consigne est un échelon  $y_m^d(k) = y_m^d(t+1) = y_m^d(t+2) = y_m^d(t+3)$ )

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

% Application de la commande
ym(k+1) = 0.8*y_m(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)
```