TP5 Techniques d'identification

1- Génération des données soit le système donne par la representation suivantes :

$$G(s) = \frac{K\omega^2 e^{-\tau s}}{s^2 + 2\varepsilon\omega + \omega^2}$$

Avec $\tau = 0,\ \ \omega = 20 * pi\ \ ,\varepsilon = 0.1\ \ ,K = 1$

- Cree un script Matlab sous le nom **systeme_ient_lsr** qui permet de tracer la réponse indiciel par la commande **Step**

2- Génération d'un signal d'excitation PRBS une règle empirique de choix de la période d''echantillonnage donne par $\frac{2\pi}{10\omega}$ utilise la commande **idinput**

t=[0:Te:(ID_LENGTH-1)*Te]';
U=[t,u];

- on simule la reponse du systeme sous simulink sans bruit va=0

disp(' on simule la reponse du systeme sous simulink sans bruit ');
sim('data_generation');

```
disp('synthese du regresseurpar moindre carre simple :');
disp('i=1...length(Y) : PHI(i,1:4)=[-Y(i-1) -Y(i-2) u(i-1) u(i-2)];');
for i=1:length(Y)
    if i==1
        PHI(i,1:4)=[0 0 0 0];
    end
    if i==2
        PHI(i,1:4)=[-Y(1) 0 u(1) 0];
    end
    if i>2
        PHI(i,1:4)=[-Y(i-1) -Y(i-2) u(i-1) u(i-2)];
    end
end
PHI;
disp('calcul des paremetres : ');

 THETA=inv(PHI'*PHI)*PHI'*Y
  GLS=tf(THETA(3:4)',[1 THETA(1:2)'],Te)
```

3- Discrétisation de système, la fonction de transfert échantillonnée théorique est la suivante :

Gz=c2d(G,Te,'zoh')

- Tracer la réponse initiale de **Gz** par la commande **Step** dans la même figure de la réponse continue
- renvoie le numérateur et le dénominateur sous forme vecteur ligne

[BZ,AZ]= tfdata(GZ,'v')

- définition du filtre du bruit

CZ=[1 -0.5];

4. Soit le système donne par le modèle Simulink suivant nommer par (**LSR_data**) , Simule la réponse du système sous Simulink sans bruit puis avec un bruit de **variance 0.1** ;on utilise la commande (**sim)**

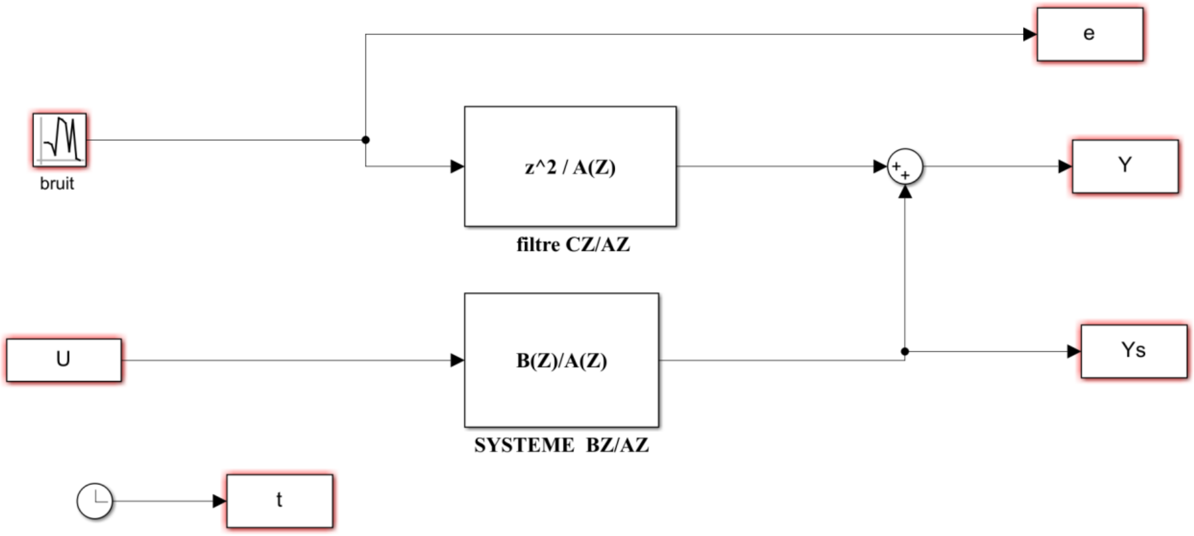

Schéma blok d'un system(LSR_data)

Synthèses d'un estimateur par la méthode de moindre carre récursive

L'estimation de paramètres par la méthode précédente des moindres carrés présente un inconvénient majeur, la nécessité de calculer l'inverse d'une matrice, ce qui est long et parfois impossible sur certains calculateurs. On se propose de déterminer une méthode récursive qui présente les principaux avantages suivants :

- La possibilité de traiter un plus grand nombre de données que dans le cas de la formulation directe (pas de pseudo-inverse à calculer).
- Dans le cas des systèmes variant dans le temps, la forme récursive permet de “suivre” les paramètres du système.

Pour avoir une forme récursive, supposons que nous possédions une estimation des paramètres θ_N à l' instant N :

$$\theta_N = (\Psi_N^T \Psi_N)^{-1} (\Psi_N^T y_N) \qquad (2)$$

À l'instant $N + 1$, la nouvelle estimation est :

$$\theta_{N+1} = (\Psi_{N+1}^T \Psi_{N+1})^{-1} (\Psi_{N+1}^T Y_{N+1}) \qquad (3)$$

Peut réécrire sous la forme :

$$\hat{\theta}_{N+1} = \hat{\theta}_N + K_{N+1} (y_{N+1} - \psi_{N+1} \hat{\theta}_N)^{-1} \qquad (4)$$

Tels que

$$K_{N+1} = P_N\psi_{N+1}^T(I + \psi_{N+1}P_N\psi_{N+1}^T)^{-1} \quad (5)$$
$$P_{N+1} = P_N - K_{N+1}\psi_{N+1}P_N$$

- Complété le scripte Matlab (**systeme_ident_lsr**) pour remplir le vecteur Y et la matrice Ψ et calculer le valeur de $\hat{\theta}$.
- Calculer **yes** . tracer les réponses indicielles G et Gz et GLSR(GLSR : least squares recurssive transfert function)
- Faire les même calcule avec un bruit gaussien de variance 0.1, interpréter les résultats

```
%synthese dy regresseur  par moinf=dre carre recurssive sans bruit de variance 0

sim('LSR_data_NOISE');

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%ù
n=4
theta=THETA
R=eye(n)
for i=1:5000
if i==1
phi=[0 0 0 0]'
end
if i==2
phi=[-Y(1)  0 u(1) 0]';
end
if i>2
phi=[-Y(i-1) -Y(i-2)  u(i-1) u(i-2)]';
end
R=R+phi*phi';   % actualisation de R
ep(i)=Y(i)-phi'*theta; %calcule del'erreur
theta=theta+inv(R)*phi*ep(i); %calcule theta(i+1)
plotheta(i,1:n)=theta'; %stokage de vecteur theta

end


disp('comparaison des reponses indicielles;');
 GLSR=tf(theta(3:4)',[1 theta(1:2)'],Te)
 [ydis,t] = step(GZ,1000);
 [ylsr,t] = step(GLSR,1000);
 ey=ydis-ylsr;
 figure(3)
 plot(t,ey)
 figure(1)
 plot(plotheta),legend('a1','a2','b0','b1')
 figure(2)
%step(G);
 hold on
 step(GZ);
 step(GLS)
 step(GLSR);
 hold off
 legend('Gz','GLS','GLSR')
```