Travail pratique N°03

a) Méthode des moindres carrés

La méthode la plus utilisée pour déterminer les paramètres a_i et b_i du modèle consiste à évaluer l'écart quadratique moyen entre la sortie réelle du processus *y(k)* et celle du modèle $y_m(k)$ et à ajuster les paramètres du modèle pour minimiser cet écart quadratique.

La sortie y_m du modèle à l'instant k connaissant la suite des entrées peut s'écrire (en supposant que $a_0 = 1$):

$$y_m(n) = -a_1\, y_m(n-1) - a_2\, y_m(n-2) - \ldots - a_p\, y_m(n-p) + b_0\, u(n) + b_1\, u(n-1) + \ldots + b_q u(n-q)$$

Soit $v(i)$ l'écart entre la sortie réelle y du processus et la sortie y_m du modèle à l'instant i:

$$v(i) = y(i) - y_m(i)$$

L'équation précédente devient:

$$y(n) = -a_1\, y(n-1) - a_2\, y(n-2) - \ldots - a_p\, y(n-p) + b_0\, u(n) + \ldots + b_q u(n-q) + e(n)$$

avec $e(n) = -\left(v(n) + a_1\, v(n-1) + a_2\, v(n-2) - \ldots - a_p v(n-p)\right)$

e(n) est appelé résidu ou erreur de prédiction. C'est l'écart entre la sortie réelle et la sortie prédite à l'instant k.

Supposons que l'on fasse N mesures successives sur le processus du couple entrée-sortie. On peut écrire *(N-p)* fois l'équation (2) , l'ensemble des relations est regroupé sous forme matricielle:

$$\begin{bmatrix} y(N) \\ y(N-1) \\ \ldots \\ y(p+1) \end{bmatrix} = \begin{bmatrix} -y(N-1) & y(N-2) & \ldots & -y(N-p) & u(N) & \ldots & u(N-q) \\ -y(N-2) & y(N-3) & \ldots & -y(N-p-1) & u(N-1) & \ldots & u(N-q-1) \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ -y(p) & -y(p-1) & \ldots & -y(1) & u(p+1) & \ldots & u(p+1-q) \end{bmatrix} \begin{bmatrix} a_1 \\ \ldots \\ a_p \\ b_0 \\ \ldots \\ b_q \end{bmatrix} + \begin{bmatrix} \varepsilon(N) \\ \varepsilon(N-1) \\ \ldots \\ \varepsilon(p+1) \end{bmatrix}$$

de la forme : $Y = H\theta + \varepsilon$ avec θ vecteur des paramètres à estimer.

b) Estimation de θ

La méthode d'estimation des meilleures valeurs des paramètres est la méthode des moindres-carrés dont les étapes principales sont rappelées ici:

Pour estimer θ mise un critère quadratique J somme des carrés des erreurs de prédiction:

$$J(\theta)=\sum_{i=p+1}^{N}\varepsilon_i^2=\varepsilon^T\varepsilon=(Y-H\theta)^T(Y-H\theta)$$

Le minimum de J est obtenu en recherchant la valeur θ qui annule les dérivées partielles par rapport à chacune des composantes de θ soit $\frac{\partial J}{\partial\theta}=0$ et l'optimum de θ est donné par :

$$\hat{\theta}=(H^TH)^{-1}H^TY$$

EXEMPLE DE PROGRAMME SOUS MATLAB

```
%   Exemple d'identification des paramètres d'un modèle récursif  du 2ème ordre du type :
%                    y(n) + a1.y(n-1) + a2.y(n-2) = b1.u(n-1)
%   Ce modèle correspond celui d'un système analogique du 2nd ordre dont l'entrée est échantillonnée-bloquée.
%  Le calcul des paramètres est fait dans une première étape par la méthode non-récursive des moindre-carrés;
%  il est fait ensuite par la méthode récursive.
%  Les résultats finaux doivent concorder entre les deux méthodes !

clear all;

% La dimension de theta est celle du nombre total de paramètres inconnus (a1, a2 et b1 dans cet exemple).
thetan= [0 0 0]';

Pn=1000*eye(size(thetan,1));

% Vecteur des mesures (vous pouvez placer vos propres points de mesures) ,
% le programme tient compte automatiquement du nombre de mesures définies dans Y .
% L'exemple initial est celui d'un second ordre avec dépassement.

Y=[0 .1 1.8 .9 1.1 .95 .97 .99 1.02 1.01 ]';

% L'entrée est supposée être un échelon. Le vecteur d'entrée peut être remplacé par
% une autre fonction des valeurs mesurées.

U=ones(size(Y),1);

% Détermination du vecteur-temps associé aux mesures.

Te=1;
T=[0:Te:(size(Y,1)-1)*Te];

stem(T,Y)
AXIS([0 Te*size(Y,1) 0 max(Y)*1.1])
hold on

% Méthode directe: construction de la matrice H puis calcul de la pseudo-inverse theta

for i = 3:size(Y,1)
    H(i,:) = [-Y(i-1) -Y(i-2) U(i-1)];
end
theta = inv(H'*H)*H'*Y

% Méthode récursive

for i = 3:size(Y,1)
    hn1=[-Y(i-1) -Y(i-2) U(i-1)]';
    Kn1 = Pn*hn1/(1+hn1'*Pn*hn1);
    thetan1=thetan + Kn1*(Y(i) - hn1'*thetan);
    Pn1 = Pn - Kn1*hn1'*Pn;
    Pn=Pn1;
    thetan=thetan1;
end
thetan

% Simulation avec les valeurs finales de thetan

Ysim=zeros(size(Y,1));

for i =3:size(Y,1)
    Ysim(i)= - Ysim(i-1)*thetan(1) - Ysim(i-2)*thetan(2) + U(i-1)*thetan(3);
  end
plot(T,Ysim,'r')
```