Parcial 2 – Modelación Experimental– Punto 1

Andrés Grimaldos Echavarría

Andrés Grimaldos Echavarría:

i) Dado que tenemos un arx 141:

$$y(t) = -a_1 y(t-1) + b_1 u(t-1) + b_2 u(t-2) + b_3 u(t-3) + b_4 u(t-4)$$

Luego, tomando $Z\{\cdot\}$ a ambos lados:

$$G(z) = \frac{Y(z)}{U(z)} = \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 + a_1 z^{-1}}$$

$$= \frac{b_1 z^3 + b_2 z^2 + b_3 z + b_4}{z^3 (z + a_1)}$$

Donde los parámetros $\theta = [a_1 \;\; b_1 \;\; b_2 \;\; b_3 \;\; b_4]^T$ serán estimados en MATLAB, usando:

$$\phi = \begin{bmatrix} \varphi^T(4) \\ \varphi^T(5) \\ \varphi^T(6) \\ \varphi^T(7) \end{bmatrix} = \begin{bmatrix} y(3) & u(0) & u(1) & u(2) & u(3) \\ y(4) & u(1) & u(2) & u(3) & u(4) \\ y(5) & u(2) & u(3) & u(4) & u(5) \\ y(6) & u(3) & u(4) & u(5) & u(6) \end{bmatrix}$$

$$Y = \begin{bmatrix} y(4) \\ y(5) \\ y(6) \\ y(7) \end{bmatrix}$$

Por mínimos cuadrados por $\theta = (\phi^T \phi)^{-1} \phi^T Y$.

Utilizamos para esto la función de mínimos cuadrados:

```matlab
function [phi, Y, param, cova] = MCO(data,na, nb, nk)
% Function for Least Squares.
% Arguments:
% - data: matrix as [t, u, y].
% - nk: delay of the system. nk = d + 1.
% Output:
% - phi: matrix with the autorregressive model in each time.
% - Y: measured observations from t = na + nk - 1 to t = N.
% - param: vector with the estimation of the parameters.

y = data(:,3);
u = data(:,2);
N = length(data);
Q = max(na, nb + nk - 1);
size_ = N - Q;
phi = zeros(size_, na + nb);
count_aux = 0;
aux = max(na, nb);
for i = 1:size_
    count = 1;
    for j = 1:na
        phi(i,count) = -y(Q - j + 1 + count_aux);
        count = count + 1;
    end
    for j = 1:nb
        phi(i,count) = u(nb - j + 1 + count_aux);
        count = count + 1;
    end
    count_aux = count_aux + 1;
end
```

```matlab
Y = data((Q+1):end,3);
param = inv(phi'*phi)*(phi')*Y;
V = 0.5*(Y'*Y - Y'*phi*inv(phi'*phi)*phi'*Y);
lambda = 2*V/(N-nk-na-nb);
cova = lambda*inv(phi'*phi);
end
```

Sin embargo, la matriz phi obtenida es:

```
phi =

   -1.1868    1.0000    1.0000    1.0000    1.0000
   -1.2544    1.0000    1.0000    1.0000    1.0000
   -1.2402    1.0000    1.0000    1.0000    1.0000
   -1.2463    1.0000    1.0000    1.0000    1.0000
```

Que tiene 4 columnas iguales, lo que provoca que phi'*phi no sea invertible, Además la matriz $R = \frac{\Phi^T \Phi}{N+1}$, tiene determinante 0, lo cual hace que sea no invertible:

```
R =

    0.6748   -0.5475   -0.5475   -0.5475   -0.5475
   -0.5475    0.4444    0.4444    0.4444    0.4444
   -0.5475    0.4444    0.4444    0.4444    0.4444
   -0.5475    0.4444    0.4444    0.4444    0.4444
   -0.5475    0.4444    0.4444    0.4444    0.4444
```

Y así, no haya solución, pues el orden de persistencia excitada de la entrada escalón sería de 1 y, en este caso, podríamos aprovecharnos del teorema de Ljung y convertirlo en orden de excitación 2.

Ahora, para solucionarlo, podemos optar por tomar valores en el pasado, entonces, consideremos un instante del tiempo anterior y en caso de valores de tiempo negativo tomaremos u = 0, así, tendríamos nuestra matriz phi:

$$\Phi = \begin{bmatrix} \phi^T(1) \\ \phi^T(2) \\ \phi^T(3) \\ \phi^T(4) \\ \phi^T(5) \\ \phi^T(6) \\ \phi^T(7) \end{bmatrix} = \begin{bmatrix} y(0) & u(0) & u(-1) & u(-2) & u(-3) \\ y(1) & u(1) & u(0) & u(-1) & u(-2) \\ y(2) & u(2) & u(1) & u(0) & u(-1) \\ y(3) & u(3) & u(2) & u(1) & u(0) \\ y(4) & u(4) & u(3) & u(2) & u(1) \\ y(5) & u(5) & u(4) & u(3) & u(2) \\ y(6) & u(6) & u(5) & u(4) & u(3) \end{bmatrix}$$

```
phi =

    0.0052    1.0000         0         0         0
    2.1190    1.0000    1.0000         0         0
    1.4082    1.0000    1.0000    1.0000         0
    1.1868    1.0000    1.0000    1.0000    1.0000
    1.2544    1.0000    1.0000    1.0000    1.0000
    1.2402    1.0000    1.0000    1.0000    1.0000
    1.2463    1.0000    1.0000    1.0000    1.0000
```

Que tiene orden de excitación en su entrada sería 2, lo cual nos permitiría por el teorema Ljung, estimar un modelo de orden 1 de arx:

```
>> data = iddata(Y,phi(:,end),1);
orden_excit = pexcit(data)

orden_excit =

     2
```

De esta manera obtenemos a_1, b_1, b_2, b_3 y b_4 respectivamente de arriba hacia abajo:

```
param =

   -0.1861
    2.1200
   -0.3174
   -0.3537
    0.0268
```

Y así, la función de transferencia sería:

```
G =

  2.12 z^3 - 0.3174 z^2 - 0.3537 z + 0.02682
  ------------------------------------------
               z^4 - 0.1861 z^3

Sample time: 1 seconds
Discrete-time transfer function.
```

Y la matriz de covarianza sería:

```
cova =

   0.001323477544603  -0.000006882083232  -0.002797566833782   0.000940727838704   0.000233296004175
  -0.000006882083232   0.000003763371263   0.000010819763105  -0.000004891784761  -0.000001213139222
  -0.002797566833782   0.000010819763105   0.005920951942110  -0.001992238089883  -0.000493141093625
   0.000940727838704  -0.000004891784761  -0.001992238089883   0.000676124516612   0.000162099215337
   0.000233296004175  -0.000001213139222  -0.000493141093625   0.000162099215337   0.000045783733674
```

Y así las desviaciones estándar de los parámetros serían las raíces cuadradas de los elementos de la diagonal, obteniendo:

$$a_1 = -0.1861 \pm 0.0364$$

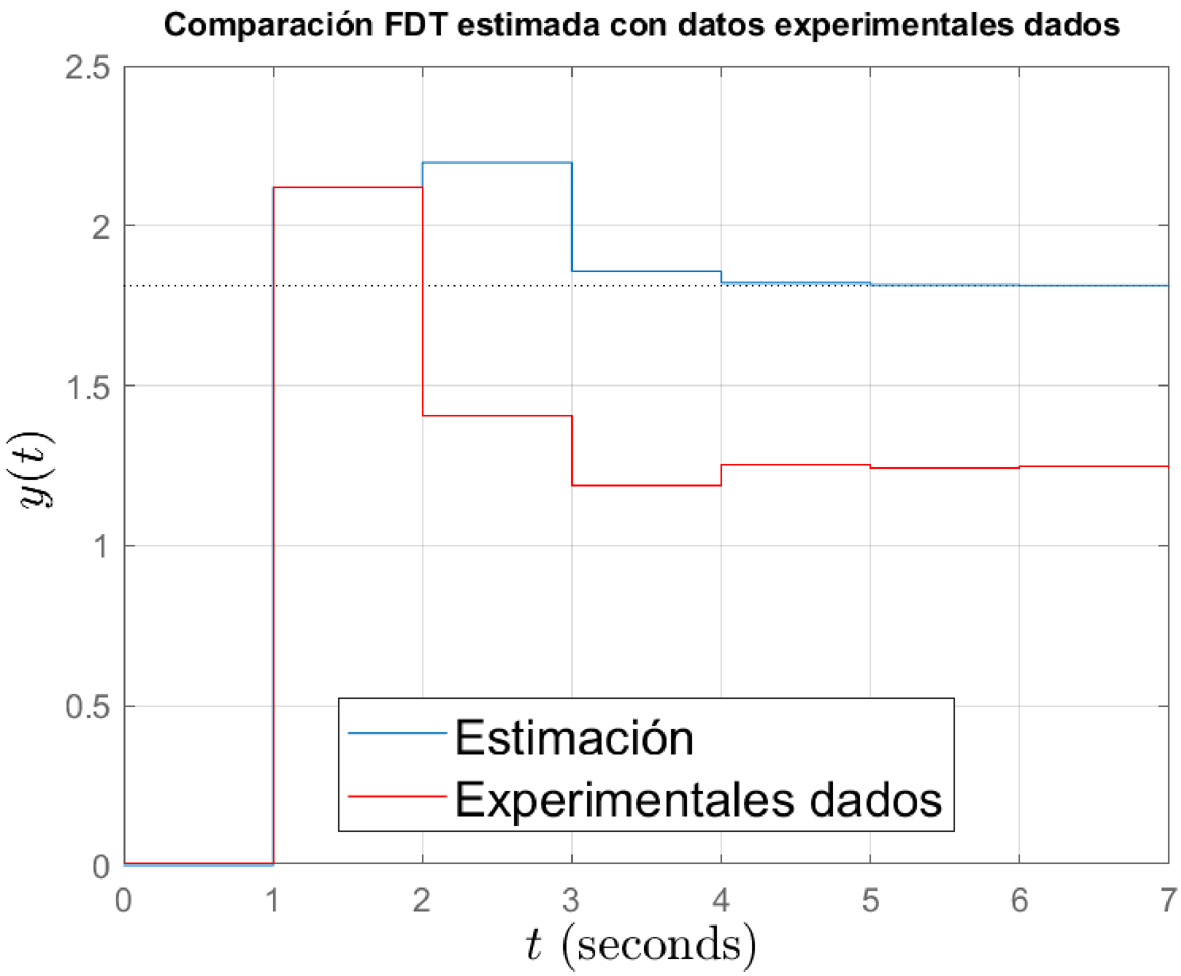
$$b_1 = 2.12 \pm 0.0019$$

$$b_2 = -0.3174 \pm 0.0769$$

$$b_3 = -0.3537 \pm 0.0260$$

$$b_4 = 0.0268 \pm 0.0068$$

Y ahora, comparando la respuesta de la función de transferencia hallada a la entrada escalón con respecto a los datos experimentales sería:



De donde se ve que la estimación no es tan buena, lo que nos puede dar pie a conjeturar que la estructura del modelo de estos datos experimentales adecuada no sería arx141 o que incluso el valor de 2.1190 esté teniendo un peso sobre la estimación, siendo un posible outlier, puesto que este método es sensible a outliers.