

Práctica 2: Identificación de Procesos

Las ecuaciones en diferencia del motor responden tiene como entrada la tensión, aplicada a través un canal de salida de la tarjeta del ordenador, y salida la velocidad angular de giro del motor, medida por un tacómetro.

A continuación se describen los pasos a seguir en la práctica. Tras arrancar el ordenador

1º) Copiar localmente en el disco duro los ficheros correspondientes a la práctica (acordaros de borrarlos al acabar). Estos ficheros se encuentran en el servidor de prácticas, concretamente en la carpeta \\vmisalabos\practicas (hay un acceso directo al servidor de prácticas en el escritorio)

2º) Abrir el programa *Matlab*, cliqueando en el acceso directo disponible en el escritorio.

3º) Desde el *Matlab* y con la fuente de alimentación del motor apagada, ejecutar el programa *EJECUTAR_AL_PRINCIPIO.m*

4º) Descripción de ficheros:

- *Prueba.m*: Se aplica al motor un escalón de tensión

Vectores resultantes: $u1 \rightarrow$ Entrada, $Vel1 \rightarrow$ Salida, $t1 \rightarrow$ Tiempo

- *Prac.m*: Aplica una señal excitante para la identificación de Mínimos Cuadrados

Vectores resultantes: $u2 \rightarrow$ Entrada, $Vel2 \rightarrow$ Salida, $t2 \rightarrow$ Tiempo

Ambos ficheros trabajan con un periodo de muestro de 0.01 segundos. Este es el periodo con el que se adquiere la señal de velocidad angular proveniente del sensor.

Ejecutar varias veces *Prueba.m* con distintos valores para el escalón (ej. entre 2 y 7). Antes de ejecutar dicho programa hay que asignarle un valor a la variable v_ent . El programa *Prueba.m* visualiza la entrada (escalón) y la salida (velocidad).

De todas formas, desde la ventana de comando de Matlab, se puede visualizar dichas gráficas ejecutando los comandos: `plot(t1, Vel1); hold; plot(t1, u1, 'r')`. En la misma ventana aparecería la velocidad a la que ha girado el motor en azul y la entrada escalón en rojo.

Ejecutar *Prac.m* y visualizar los resultados. Este programa no necesita ninguna asignación de valores y visualiza la entrada proporcionada (excitación) y la correspondiente salida. Utilizando el comando *plot* de Matlab también se puede visualizar dichas señales.

5º) A continuación se describe parte de la teoría del algoritmo de Mínimos Cuadrados:

Se tiene la siguiente la siguiente ecuación discreta o en diferencias:

$$y(k) = -a_1 \cdot y(k-1) - \dots - a_{na} \cdot y(k-na) + b_0 \cdot u(k-d) + \dots + b_{nb-1} \cdot u(k-d-nb+1)$$

Control por Computador (Escuela Técnica Superior de Informática)

Pasos para la Práctica 2 de Identificación

donde y es la salida, u la entrada, d el retardo, nb el número de coeficientes acompañando a valores de entrada en instantes anteriores al actual; na el número de coeficientes que acompañaban a valores de salida en instantes previos.

Particularizando la ecuación en diferencias anterior del instante M al instante N y poniendo el resultado en forma matricial:

$$\begin{pmatrix} y(M) \\ \vdots \\ y(N) \end{pmatrix} = \begin{pmatrix} -y(M-1) & \cdots & -y(M-na) & u(M-d) & \cdots & u(M-d-nb+1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -y(N-1) & \cdots & -y(N-na) & u(N-d) & \cdots & u(N-d-nb+1) \end{pmatrix} \cdot \begin{pmatrix} a_1 \\ \vdots \\ a_{na} \\ b_0 \\ \vdots \\ b_{nb} \end{pmatrix}$$

$$Y = \phi \cdot \theta$$

El instante N será el último de los vectores de entrada y salida, ej. $\text{length}(y)$, mientras que el instante M será aquel que garantice que no hay ningún elemento con índice menor o igual que H (H son los elementos iniciales de los vectores de entrada y salida que se quieren despreciar), es decir:

$$M = H + 1 + \max(na, nb - 1 + d)$$

La ecuación de Mínimos Cuadrados, para estimar los parámetros del modelo, es:

$$\hat{\theta} = (\phi^T \cdot \phi)^{-1} \cdot \phi^T \cdot Y \rightarrow \text{en Matlab} \rightarrow \text{theta} = \text{inv}(f_i' \cdot f_i) \cdot f_i' \cdot Y$$

Teniendo en cuenta la teoría anterior, se pide que se programe una función genérica de identificación a la que se le pase (y , u , na , nb , d , H) (vector de salida, vector de entrada, número de parámetros incógnitas a_i , número de parámetros incógnitas b_j , retardo del modelo, primeros instantes de la excitación y salida a despreciar

Una vez desarrollada la función llamarla con:

- $y = \text{Vel2}$, $u = u2$, $na = 1$, $nb = 1$, $d = 1$, $H=7$, $N=\text{length}(\text{Vel2})$. El programa que diseñaremos nos devolverá dos parámetros a_1 y b_0 .
- $y = \text{Vel2}$, $u = u2$, $na = 2$, $nb = 1$, $d = 1$, $H=7$, $N=\text{length}(\text{Vel2})$. El programa que diseñaremos nos devolverá dos parámetros a_1 , a_2 y b_0 .
- $y = \text{Vel2}$, $u = u2$, $na = 2$, $nb = 2$, $d = 1$, $H=7$, $N=\text{length}(\text{Vel2})$. El programa que diseñaremos nos devolverá dos parámetros a_1 , a_2 , b_0 y b_1 .
- Finalmente, llamar a la función con $y = \text{Vel2}$, $u = u2$, $na = 3$, $nb = 3$, $d = 1$, $H=7$, $N=\text{length}(\text{Vel2})$. El programa que diseñaremos nos devolverá dos parámetros a_1 , a_2 , a_3 , b_0 , b_1 , y b_2 .

Control por Computador (Escuela Técnica Superior de Informática)

Pasos para la Práctica 2 de Identificación

6º) VALIDACIÓN DEL MODELO. Diseñar un programa en Matlab para observar la salida que proporciona cada modelo que acabamos de identificar. A cada modelo se le pasará la entrada u_1 y su salida será comparada con la salida que dio el motor real (Vel_1)

Observa, las salidas de cada modelo, y observa cuál de ellos se parece más a la respuesta que dio el motor real cuando se le proporcionó la entrada u_1 (hecho por el programa *Prueba.m*)

7º) Repetir todos los pasos seguidos anteriormente para identificar un nuevo sistema. Los datos se encuentran en el fichero "*datos_experimento.mat*".

En primer lugar, se cargará el anterior fichero ejecutando, desde la ventana de comandos de Matlab, el siguiente comando:

```
>> load datos_experimento.mat
```

Aparecerán 5 variables nuevos en la ventana del workspace de Matlab: T , in_sys , out_sys , $salida_ante_escalon$ y $tiempo$. in_sys contiene la excitación que se le ha proporcionado al sistema con periodo de muestreo T seg. out_sys es la salida que ha proporcionado el sistema ante la entrada in_sys . La variable $salida_ante_escalon$ contiene la salida que ha dado el sistema ante una entrada escalón unitario. La variable $tiempo$ puede utilizarse para visualizar las señales anteriores, por ejemplo, ejecutando comandos como:

```
>> plot(tiempo,salida_ante_escalon)
>> plot(tiempo,out_sys)
```

Utilizando el código realizado, identificar el modelo (ecuación en diferencias) del sistema. Para validar compare la salida del modelo encontrado ante una entrada escalón unitario con la señal $salida_ante_escalon$

8º) Al final de la práctica/sesión:

- Subir al Espacio Compartido del PoliformaT un fichero zip con todos los programas diseñados, y un fichero Word/pdf con todos los cálculos/ejecuciones realizados, insertar las gráficas en formato jpg y cualquier análisis o comentarios sobre las gráficas o resultados obtenidos.