

MODELADO E IDENTIFICACIÓN DE SISTEMAS 5º INGENIERÍA INDUSTRIAL

PRÁCTICA 5

IDENTIFICACIÓN POR MÍNIMOS CUADRADOS RECURSIVO

Curso 2002-2003

1. Objetivos

En el desarrollo teórico del tema de identificación se ha estudiado el método de identificación paramétrica basado en el algoritmo de Mínimos Cuadrados Recursivo. Este algoritmo es de gran importancia debido a que permite realizar la identificación de los parámetros de un sistema “en línea” lo cual resulta imprescindible para abordar el estudio de estructuras de control adaptativo. Es por esto por lo que, por medio de esta práctica, se va a desarrollar los aspectos vistos en teoría dentro de un marco práctico que ayude a comprender el funcionamiento del mismo. Así, en esta práctica se pretenden alcanzar los siguientes objetivos:

- Comprensión práctica de la identificación de sistemas “en línea” mediante el algoritmo de Mínimos Cuadrados Recursivo.
- Conocimiento y aplicación de la técnica de Mínimos Cuadrados Recursivo para la obtención de modelos discretos
- Utilización de la aplicación Matlab/Simulink para realizar los procesos de identificación y de validación mediante simulación.

Aunque la práctica se realiza con un conjunto de datos suministrados por el profesor, el desarrollo de la misma debe permitir comprender cómo se realizaría una identificación en línea en tiempo real con un sistema físico real.

2. Identificación de Sistemas. Método de Mínimos Cuadrados Recursivo

El algoritmo de Mínimos Cuadrados Recursivo permite estimar en tiempo real los parámetros de un determinado sistema. Obviamente, esta variante del algoritmo de Mínimos Cuadrados está sujeta a las mismas restricciones que su homólogo iterativo (conocimiento a priori de la estructura de la función de transferencia del modelo, y de su retardo, etc.).

El algoritmo de Mínimos Cuadrados Recursivo se puede resumir matemáticamente en las siguientes fórmulas:

$$P_{k+1} = \left[P_k - \frac{P_k x_{k+1} x_{k+1}^T P_k}{1 + x_{k+1}^T P_k x_{k+1}} \right]$$

$$K_{k+1} = \frac{P_k}{1 + x_{k+1}^T P_k x_{k+1}}$$

$$e_{k+1} = y_{k+1} - x_{k+1}^T \mathbf{q}_k$$

$$\mathbf{q}_{k+1} = \mathbf{q}_k + K_{k+1} x_{k+1} e_{k+1}$$

donde:

θ_{k+1} denota el vector de parámetros estimados en un instante

θ_k denota el vector de parámetros estimados en el instante anterior

K_{k+1} es una matriz de ganancias

x_{k+1} es el vector de regresión en ese instante

e_{k+1} es el error de predicción

Así pues, dadas las fórmulas expuestas anteriormente, los pasos a seguir para la utilización de este algoritmo se describen a continuación:

1. Definir los valores iniciales del vector θ y de la matriz P
2. Tomar medidas de las señales de entrada y salida
3. Formar el vector de regresión x
4. Calcular la matriz de ganancias K
5. Calcular el error de predicción e
6. Calcular los parámetros estimados
7. Calcular la matriz P para el instante siguiente
8. Volver al punto 2 en el siguiente muestreo

Como se puede observar, se necesita a priori dar un valor inicial a la matriz P y al vector de parámetros estimados. La matriz P se suele inicializar como $P = aI_{(m+n) \times (m+n)}$ y θ , si no se tiene ningún tipo de conocimiento acerca de los parámetros, a 0.

El problema que presenta este algoritmo es el apagado del mismo (incapacidad para detectar cambios en parámetros) a lo largo del tiempo debido a que la traza de la matriz P se hace muy pequeña. Esto se puede solucionar empleando el algoritmo de Mínimos Cuadrados Recursivo con Factor de Olvido. Este algoritmo introduce un factor de olvido, λ , de forma que se pondera más a las muestras más recientes, permitiendo así detectar un posible cambio de parámetros cuando el algoritmo lleva un tiempo considerable funcionando y la traza de la matriz P es muy pequeña.

Matemáticamente, dicho algoritmo se puede resumir en las siguientes fórmulas, siendo su procedimiento de aplicación el mismo que en el caso anterior:

$$P_{k+1} = \frac{1}{\lambda} \left[P_k - \frac{P_k x_{k+1} x_{k+1}^T P_k}{\mathbf{I} + x_{k+1}^T P_k x_{k+1}} \right]$$

$$K_{k+1} = \frac{P_k}{\mathbf{I} + x_{k+1}^T P_k x_{k+1}}$$

$$e_{k+1} = y_{k+1} - x_{k+1}^T \mathbf{q}_k$$

$$\mathbf{q}_{k+1} = \mathbf{q}_k + K_{k+1} x_{k+1} e_{k+1}$$

Dependiendo del valor de λ , $0 < \lambda < 1$, se obtienen los siguientes comportamientos:

- λ pequeño: Se descartan datos pasados demasiado pronto, los elementos de las matrices P y K se hacen grandes, lo cual supone la posibilidad de detección de posibles cambios en parámetros
- λ grande: (cercano a la unidad) se presentan problemas a la hora de detectar cambios de parámetros.

Normalmente se suele elegir λ entre 0.9 y 0.98 pero la elección depende bastante de los valores asociados a los parámetros del sistema.

3. Realización de la práctica

La práctica a realizar consiste en:

- Identificar un modelo de segundo orden cuya función de transferencia corresponde a

$$G(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \cdot z^{-d}$$

mediante los datos de entrada salida que suministrará el profesor. El periodo de muestreo empleado ha sido de $T = 0.1$ seg.

- Realizar la experiencia anterior pero con un conjunto de datos de entrada – salida donde se produce una variación de parámetros. Dicha experiencia va a permitir comprobar el apagado del algoritmo y cómo se evita éste mediante la utilización del Algoritmo de Mínimos Cuadrados Recursivo con Factor de Olvido.

Recordemos, por si puede servir de ayuda, que el modelo matricial de regresión para esta estructura de segundo orden viene dado por:

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+N) \end{bmatrix} = \begin{bmatrix} -y(k) & -y(k-1) & u(k+1-d) & u(k-d) \\ -y(k+1) & -y(k) & u(k+2-d) & u(k+1-d) \\ \vdots & \vdots & \vdots & \vdots \\ -y(k+N-1) & -y(k+N-2) & u(k+N-d) & u(k+N-1-d) \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ a_2 \\ b_0 \\ b_1 \end{bmatrix}$$

Notemos, que aunque la experiencia se va a realizar mediante el algoritmo recursivo fuera de línea, la mecánica del trabajo a realizar se adapta perfectamente a su realización en línea.

Los datos para realizar ambas experiencias se suministran en el fichero Matlab datos.mat que contiene las siguientes variables:

Variable	Nº Datos	Descripción
U1	1001	Entrada para experiencia sin cambio de parámetros
Y1	1001	Salida para experiencia sin cambio de parámetros
Yesc1	1001	Salida ante entrada escalón para experiencia sin cambio de parámetros (para validar el modelo obtenido)
U2	1001	Entrada para experiencia con cambio de parámetros
Y2	1001	Salida para experiencia con cambio de parámetros
Yesc2	1001	Salida ante entrada escalón para experiencia con cambio de parámetros (para validar el modelo obtenido)

Lo primero que se debe hacer es un programa Matlab que implemente el algoritmo de Mínimos Cuadrados Recursivo. Dicho programa tendrá la siguiente forma:

```
% Identificación mediante RLS

% Inicialización de variables
theta = ...;
alpha = ...; % Para inicializar la matriz P. Por ejemplo 10
lambda = ...; % En el caso de tener factor de olvido
                % Ojo, en este caso también cambia la formulación
                % de las matrices P y K

P = ...;

% Puesto que los vectores en Matlab comienzan por
% la componente 1, conviene realizar la primera
% estimación antes de entrar en el bucle

x = ...; % Vector de regresión de la primera estimación
K = ...; % Matriz de ganancias de la primera estimación
e = ...; % Error de predicción de la primera estimación
theta = ...; % Primera estimación

% Guardamos los parámetros estimados en cada iteración
% así como la traza de la matriz P
params = theta;
traza = trace(P);

% El valor final de este bucle depende de las medidas que tengamos
for i=1:Nmuestras
    x = ...; % Calculamos el nuevo vector de regresión
    P = ...; % Calculamos la nueva matriz P
    K = ...; % Calculamos la nueva matriz K
    e = ...; % Calculamos el nuevo error de predicción
    theta = ...; % Estimamos los nuevos parámetros
    % Guardamos traza y parámetros
    params = [params theta];
    traza = [traza trace(P)];
end
```

```

% Representación gráfica de parámetros y traza
figure;
plot(params(1,:));
hold;
plot(params(2,:), '--');
plot(params(3,:), 'r');
plot(params(4,:), 'r--');

figure;
plot(traza);

```

Obviamente, nuestro algoritmo utilizará las variables de entrada – salida que tenemos en el workspace de Matlab.

Así pues, las tareas a realizar son:

- a) Determinación del retardo que presenta el sistema.
- b) Identificación del modelo del proceso mediante Mínimos Cuadrados Recursivo utilizando los datos Y1, U1. Utilizar $a = 10$ y repetir la experiencia con $a = 50$.
- c) Validación del los modelos obtenido en a) mediante simulación ante entrada escalón y comparando con la salida Yesc1.
- d) Identificación del modelo del proceso mediante Mínimos Cuadrados Recursivo utilizando los datos Y2, U2.
- e) Validación del modelo obtenido en d) mediante simulación ante entrada escalón y comparando con la salida Yesc2.
- f) Identificación del modelo del proceso mediante Mínimos Cuadrados Recursivo con Factor de Olvido utilizando los datos Y2, U2.
- g) Validación del modelo obtenido en f) mediante simulación ante entrada escalón y comparando con la salida Yesc2.
- h) Repetir el apartado f) para distintos valores del factor de olvido comprobando el efecto sobre la identificación obtenida y sobre la traza de la matriz P. ¿Qué sucede para valores pequeños del factor del olvido?

4. Informe de prácticas a entregar

En el informe de prácticas se debe detallar:

- Procedimiento detallado de la identificación de los sistemas, comentando y justificando todos los aspectos y valores de parámetros empleados en el algoritmo.
- Gráficas de convergencia de parámetros y traza de la matriz P comentando los aspectos relevantes de éstas para cada identificación.
- Justificación de los distintos comportamientos del algoritmo para los distintos valores de parámetros del mismo (inicialización de P, factor de olvido, etc.).
- Suponiendo que se dispone de las funciones *aplicar_entrada()*, *leer_salida()* y *retardo(t)*, esbozar el programa que permitiría realizar la identificación en línea.
- Cualquier aspecto relevante de la práctica que se quiera hacer notar.