

**IDENTIFICACION DE PROCESOS POR EL
METODO RECURSIVO DE PROBABILIDAD MAXIMA**

ING. PATRICIO BURRANO
FACULTAD DE INGENIERIA ELECTRICA
ESCUELA POLITECNICA NACIONAL

ING. EDISON RICARDO CRUZ LORA
FACULTAD DE INGENIERIA ELECTRICA
ESCUELA POLITECNICA NACIONAL

RESUMEN:

En el presente trabajo se desarrolla un algoritmo de identificación de procesos, basado en el Método de Probabilidad Máxima, utilizando el modelo de ecuación de diferencias conocido como ARMAX. Los resultados obtenidos permiten verificar la bondad del método escogido y de los programas desarrollados para determinar el modelo matemático de la dinámica del proceso. Las pruebas se realizan con diferentes tipos de excitación, poniendo énfasis en la señal binaria pseudo aleatoria (p.r.b.s.) que posee características de excitación persistente. Se considera perturbación de tipo aleatorio a la salida. El software desarrollado se ha codificado en lenguaje C por sus características de adaptabilidad al hardware para una aplicación de tiempo real.

ABSTRACT:

This paper develops a process identification algorithm based on Maximum Likelihood Method using an ARMAX model. The results verify selected method and developed programs that give the process dynamics mathematical model. Tests deal with several excitation signals, supporting theory on pseudo-random binary signal (p.r.b.s.) due to its characteristics like persistence excitation. It considers perturbation in the output signal also. Software is coded in C programming language because of its adaptability to the hardware for real time applications.

1. MODELO MATEMATICO UTILIZADO

La identificación de un sistema consiste en la obtención de un modelo matemático que caracteriza la dinámica y se puede predecir el comportamiento de la planta. El modelo utilizado es discreto mediante ecuaciones de diferencias de sistemas de parámetros concentrados. Dentro de este tipo de modelos, el escogido para desarrollar el algoritmo de identificación de procesos es el modelo ARMAX (AUTOREGRESSIVE MOVING-AVERAGE MODEL with auxiliary input), y consiste en una ecuación de diferencias en términos de las entradas y salidas del sistema. El vector de salidas actual se expresa como una combinación lineal de las salidas y entradas anteriores, y se expresa de la siguiente manera:

$$Aey(t) = - \sum_{j=1}^{n1} A_j y(t-j) + \sum_{j=0}^{m1} B_j u(t-j) + \sum_{j=1}^{h1} C_j w(t-j) \quad ; t \geq 0$$

donde: A_1, B_1, C_1 , son parámetros constantes;
 y , salida del sistema;
 u , entrada del sistema;

y este modelo nos proporciona una descripción válida de las propiedades entrada-salida del sistema bajo estudio como se verá más adelante.

2. FUNCION DE PROBABILIDAD MAXIMA

El principio de probabilidad máxima se lo puede expresar de la siguiente manera:

"Sea y una variable aleatoria cuya densidad de probabilidad, $p(y, \theta)$, depende de un parámetro desconocido θ , para proceder a estimar θ , la variable " y " toma aquel valor de θ que maximiza la función real:

$$L(y, \theta) = p(y, \theta)$$

llamada función de probabilidad. En el caso de ser varios los parámetros desconocidos, θ se constituye en vector para que sea fácilmente interpretable.

Aunque se considerará la distribución normal o gaussiana, el método no restringe o limita la forma de la función de densidad.

Para el caso en que se tengan n variables aleatorias, también con distribución normal, el valor medio será un vector de valores medios, y la covarianza será una matriz (R) no singular.

Sea x el vector de variables aleatorias, u el vector de valores medios; la función de probabilidad será

$$f_x(\tau) = 1/[(2\pi)^n \det R]^{1/2} \exp\{-\frac{1}{2} (\tau - u)^T R^{-1} (\tau - u)\}$$

Consideremos el modelo:

$$y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_1 u(k-1) + \dots + b_n u(k-n) + v(k)$$

A $v(k)$ se le considera ruido blanco no correlacionado, con distribución normal. Se quiere determinar a_1 y b_1 a partir de las observaciones $y(k)$ para lo cual se necesita su función de densidad de probabilidad. Para facilitar la escritura, se define:

$$V(N) = [v(n) \dots v(N)]^T$$

$$Y(N) = [y(n) \dots y(N)]^T$$

$$\begin{aligned} \Phi(k) &= [-y(k-1) \dots -y(k-n) \ u(k-1) \dots u(k-n)]^T \\ \Phi(N) &= [\Phi(n) \dots \Phi(N)]^T \\ \theta^* &= [a_1 \dots a_n \ b_1 \dots b_n]^T \end{aligned}$$

Luego el modelo se puede escribir también en la forma:

$$Y(N) = \Phi^* \theta^* + V(N)$$

Asumiendo todos los elementos conocidos, se puede encontrar la función de densidad de probabilidad de Y, puesto que la de V es conocida (ruido blanco, distribución normal), y si se la despeja, esta se la puede expresar como:

$$V(N) = Y(N) - \Phi(N) \theta^*$$

Se puede escribir, entonces, en forma inmediata que:

$$f(y|\theta^*) = (2\pi\sigma^2)^{-m/2} \exp\left\{-\frac{(Y - \Phi\theta^*)^T(Y - \Phi\theta^*)}{2\sigma^2}\right\}$$

donde: $m (=N-n+1)$ es el número de muestras de Y.

La función de probabilidad queda determinada en su forma general si se sustituye θ^* por θ , y se tiene: $f(y|\theta)$. Sacando el logaritmo natural, se tiene:

$$\begin{aligned} -\log f(y|\theta) &= \log (2\pi\sigma^2)^{-m/2} \\ &+ \frac{(y - \Phi\theta)^T(y - \Phi\theta)}{2\sigma^2} \\ &= \frac{m}{2} \log 2\pi + \frac{m}{2} \log \sigma^2 \\ &+ \frac{(y - \Phi\theta)^T(y - \Phi\theta)}{2\sigma^2} \end{aligned}$$

Los estimados de θ y de σ^2 que hagan que la función crezca lo mayor posible, se determinan por derivación parcial e igualando cada uno a cero; derivando primero respecto de σ^2 , se tiene:

$$\frac{\partial}{\partial \sigma^2} \left[-\frac{m}{2} \log \sigma^2 + \frac{(y - \Phi\theta)^T(y - \Phi\theta)}{2\sigma^2} \right] = 0$$

Haciendo $S = (y - \Phi\theta)^T(y - \Phi\theta)$, la ecuación se puede escribir como:

$$\begin{aligned} \frac{m}{2\sigma^2} - \frac{S}{2\sigma^4} &= 0 \\ \frac{m}{\sigma^2} - \frac{S}{\sigma^4} &= 0 \\ \hat{\sigma}^2 &= S/m \end{aligned}$$

Derivando, ahora, respecto de θ :

$$\frac{\partial}{\partial \theta} \left[-\frac{(y - \Phi\theta)^T(y - \Phi\theta)}{2\sigma^2} \right] = 0$$

$$\frac{1}{2\sigma^2} [(y - \Phi\theta)^T(-\Phi) + (y - \Phi\theta)(-\Phi^T)] = 0$$

$$\frac{1}{\sigma^2} (\Phi^T \Phi \theta - \Phi^T y) = 0$$

(El método de máxima probabilidad da el mismo resultado para θ estimado que el de mínimos cuadrados como se puede verificar).

3. ALGORITMO DE PROBABILIDAD MAXIMA

Considerando el modelo ARMAX:

$$y(t) + \sum_{i=1}^A a_i y(t-i) = \sum_{i=1}^B b_i u(t-i) + v(t) + \sum_{i=1}^C c_i v(t-i) \quad (3.1)$$

Se procede a estimar sus parámetros a_i , b_i y c_i ; $v(t)$ se considera ruido blanco, es decir, una secuencia independiente de variables aleatorias con valor medio cero.

Los parámetros a ser estimados forman el vector:

$$\theta = (a_1 \dots a_A \ b_1 \dots b_B \ c_1 \dots c_C)^T$$

Se procede a determinar la predicción de $y(t)$ basándose en las observaciones $y(s)$, $u(s)$, $\theta \leq s \leq t-1$, y también en el hecho de que se simula estas señales y por lo tanto (3.1) en verdad las produce. Se denomina a esta predicción por: $yest(t|\theta)$.

Al tiempo t , $y(t-i)$, $u(t-i)$ son conocidas. Una buena aproximación de $v(t-1)$ puede calcularse de $y(s)$, $u(s)$, utilizando (3.1) y se notará como $v_s(t-1)$. El valor de $v(s)$ no se puede predecir de los datos previos puesto que es independiente de todo lo que sucedió hasta $t = t-1$. Luego, la predicción natural de $y(t)$ es:

$$yest(t|\theta) = -a_1 y(t-1) - \dots - a_A y(t-A) + b_1 u(t-1) + \dots + b_B u(t-B) + c_1 v_s(t-1) + \dots + c_C v_s(t-C) \quad (3.2)$$

donde $v_s(t-i)$ se calcula en forma recursiva de (3.1).

A continuación se procede con el siguiente artificio:

$$D = \sum_{i=1}^C c_i$$

$$\begin{aligned} \sum_{i=1}^C c_i yest(t-i|\theta) &= -D \sum_{i=2}^A a_{i-1} y(t-i) \\ &+ D \sum_{i=2}^B b_{i-1} u(t-i) \quad (3.3) \\ &+ D \sum_{i=2}^C c_{i-1} v_s(t-i) \end{aligned}$$

Sumando miembro a miembro (3.1) y (3.3):

$$yest(t|\theta) + \sum_{i=1}^C c_i yest(t-i|\theta) = YEST$$

$$\begin{aligned} YEST &= -\sum_{i=1}^A a_i y(t-i) - D \sum_{i=2}^A a_{i-1} y(t-i) \\ &+ \sum_{i=1}^B b_i u(t-i) + D \sum_{i=2}^B b_{i-1} u(t-i) \quad (3.4) \\ &+ \sum_{i=1}^C c_i v(t-i) + D \sum_{i=2}^C c_{i-1} v_s(t-i) \end{aligned}$$

Pero:

$$v_m(t-j) = y(t-j) + \sum_{i=1}^A a_{1i} y(t-j-i) - \sum_{i=1}^B b_{1i} u(t-j-i) - \sum_{i=1}^C c_{1i} v_m(t-j-i) \quad (3.5)$$

$$j = 1, \dots, t-1$$

Reemplazando (3.5) en (3.4):

$$\begin{aligned} YEST = & - \sum_{i=1}^A a_{1i} y(t-i) - D \sum_{i=2}^A a_{1-i} y(t-i) \\ & + \sum_{i=1}^B b_{1i} u(t-i) + D \sum_{i=2}^B b_{1-i} u(t-i) \\ & + D y(t-j) + D \sum_{i=1}^A a_{1i} y(t-j-i) \\ & - D \sum_{i=1}^B b_{1i} u(t-j-i) - D^2 v_m(t-j-i) \\ & + D \sum_{i=2}^C c_{1-i} v_m(t-i) \end{aligned}$$

$$j = 1, \dots, t-1$$

Desarrollando y reduciendo términos semejantes se tiene:

$$\begin{aligned} YEST = & (c_1 - a_1) y(t-1) + (c_2 - a_2) y(t-2) \\ & + \dots + (c_c - a_A) y(t-A) \\ & + \sum_{i=1}^B b_{1i} u(t-i) \\ = & \sum_{i=1}^K (c_i - a_i) y(t-i) + \sum_{i=1}^B b_{1i} u(t-i) \quad (3.6) \end{aligned}$$

Para cualquier θ se puede calcular la predicción $yest(t|\theta)$ utilizando (3.6) y los datos coleccionados hasta el tiempo $t-1$. De la misma forma, se puede calcular, entonces, el error de predicción:

$$\epsilon(t, \theta) = y(t) - yest(t|\theta) \quad (3.7)$$

Un criterio razonable de cuan bien el modelo de parámetros θ se ha estimado constituye la suma de los cuadrados de los errores de predicción:

$$V_m(\theta) = \frac{1}{2} \sum_{t=1}^M \epsilon^2(t, \theta) \quad (3.8)$$

Se notará que esta función es parte del logaritmo natural negativo $(-\log)$ de la función de probabilidad, para cuando las variables $\{v(t)\}$ son gaussianas. Por otro lado, los parámetros estimados de θ , δ , (off-line) se obtienen por la minimización de $V_m(\theta)$, pero (3.6) no es lineal para θ , por lo tanto (3.7) no puede minimizarse analíticamente. En su lugar, una rutina de minimización numérica que requiere varios pasos iterativos a lo largo de los datos almacenados en el intervalo de tiempo $[1, N]$. No es posible la implementación de

un algoritmo recursivo puesto que no puede establecerse un vector de dimensión predeterminada para memorizar estos datos, por lo tanto se construye un método que calcula aproximaciones de δ en una forma recursiva.

Sea $\delta(t-1)$ el estimado de θ al tiempo $t-1$. Se desea obtener un $\delta(t)$ que aproximadamente minimice $V_t(\theta)$. Por medio de una expansión en Series de Taylor de $V_t(\theta)$ alrededor de $\delta(t-1)$ se obtiene:

$$\begin{aligned} V_t(\theta) = & V_t(\delta(t-1)) \\ & + V'_t(\delta(t-1)) [\theta - \delta(t-1)] \\ & + \frac{1}{2} [\theta - \delta(t-1)]^T V''_t(\delta(t-1)) [\theta - \delta(t-1)] \\ & + g(|\theta - \delta(t-1)|^2) \quad (3.8) \end{aligned}$$

donde las derivadas son respecto de θ ; $g(x)$ es una función tal que:

$$\frac{g(x)}{|x|} \rightarrow 0$$

tanto como $|x| \rightarrow 0$. Esto último se cumple puesto que el proceso de expansión se lleva a cabo en la vecindad de θ , y precisamente el módulo debe ser lo más pequeño posible.

La minimización de V respecto de θ da:

$$\begin{aligned} \delta(t) = & \delta(t-1) - [V''_t(\delta(t-1))]^{-1} V'_t(\delta(t-1))^T \\ & + g(|\delta(t) - \delta(t-1)|) \quad (3.9) \end{aligned}$$

Definiendo:

$$\Phi(t, \theta) \equiv \left[- \frac{\delta}{\delta \theta} \epsilon(t, \theta) \right]^T$$

se tiene:

$$\begin{aligned} [V'_t(\theta)]^T = & - \sum_{k=1}^t \Phi(k, \theta) \epsilon(k, \theta) \\ = & [V'_{t-1}(\theta)]^T - \Phi(t, \theta) \epsilon(t, \theta) \quad (3.10) \end{aligned}$$

Diferenciando una vez más:

$$\begin{aligned} V''_t(\theta) = & V''_{t-1}(\theta) + \Phi(t, \theta) \Phi^T(t, \theta) \\ & + \epsilon''(t, \theta) \epsilon(t, \theta) \quad (3.11) \end{aligned}$$

donde $\epsilon''(t, \theta)$ es la matriz de las segundas derivadas de $\epsilon(t, \theta)$ respecto de θ . Para poder evaluar (3.9) se introduce las siguientes aproximaciones:

1) Se asume que el siguiente $\delta(t)$ estimado será encontrado en un pequeño vecindario de $\delta(t-1)$, siempre y cuando t sea lo suficientemente grande. De esto se deduce que se despreciará la función:

$$g(|\delta(t) - \delta(t-1)|) \quad (3.12)$$

y además se considerará:

$$V''_t(\delta(t)) = V''_t(\delta(t-1)) \quad (3.13)$$

2) Es posible asumir, entonces, que $\delta(t-1)$ es en verdad el estimado óptimo al tiempo $t-1$, con lo cual

$$V'_{t-1}(\delta(t-1)) = 0 \quad (3.14)$$

3) Se asume que:

$$\epsilon''(t, \theta(t-1)) \epsilon(t, \delta(t-1)) = 0 \quad (3.15)$$

puesto que cerca del valor verdadero θ^* , $\{\epsilon(t, \theta)\}$, será casi ruido blanco; considerándolo así, es posible que se le asuma tener un valor medio cero y ser independiente de lo sucedido hasta $t = t-1$.

Con estas tres aproximaciones expuestas, y tomadas en cuenta en (3.11) se procede a evaluar la matriz de las segundas derivadas, la cual se nota: $R(t)$. Luego se tiene:

$$R(t) = R(t-1) + \Phi(t, \delta(t-1)) \Phi^T(t, \delta(t-1)) \quad (3.16)$$

Análogamente, (3.10) se puede expresar como:

$$[V^T \epsilon(\delta(t-1))]^T = -\Phi(t, \delta(t-1)) \epsilon(t, \delta(t-1)) \quad (3.17)$$

Reemplazando (3.16) y (3.17) en (3.8):

$$\delta(t) = \delta(t-1) + R^{-1}(t) \Phi(t, \delta(t-1)) \epsilon(t, \delta(t-1)) \quad (3.18)$$

Aunque (3.16) y (3.18) pueden darse en una forma recursiva, a continuación se verá cómo se puede obtener $\epsilon(t)$ y $\Phi(t)$, partiendo de (3.7) y de la definición de Φ :

$$\Phi^T(t, \theta) = -\frac{d}{d\theta} \epsilon(t, \theta) = -\frac{d}{d\theta} yest(t|\theta) \quad (3.19)$$

Aplicando esto a (3.8) se tiene que:

$$\begin{aligned} \frac{d}{da_1} yest(t|\theta) + D \frac{d}{da_1} yest(t-i|\theta) &= -y(t-i) \\ \vdots & \vdots \\ \frac{d}{db_1} yest(t|\theta) + D \frac{d}{db_1} yest(t-i|\theta) &= u(t-i) \\ \vdots & \vdots \\ \frac{d}{dc_1} yest(t|\theta) + \sum_{i=1}^n c_i \frac{d}{dc_1} yest(t-i|\theta) &= \epsilon(t-i) \end{aligned}$$

Se debe aclarar que el número de parámetros a , b o c es independiente uno de otro. Por otro lado, las tres expresiones generales anteriores se pueden escribir también en la forma:

$$\Phi(t, \theta) + \sum_{i=1}^c c_i \Phi(t-i, \theta) = \begin{bmatrix} -y(t-1) \\ \vdots \\ -y(t-n_a) \\ u(t-1) \\ \vdots \\ u(t-n_b) \\ \epsilon(t-1) \\ \vdots \\ \epsilon(t-n_c) \end{bmatrix} \quad (3.20)$$

De esta manera se puede calcular $\epsilon(t, \delta(t-1))$ y $\Phi(t, \delta(t-1))$ resolviendo (3.6), (3.7) y (3.20), respectivamente. Se utilizarán todos los datos para el ruido y para Φ hasta el instante $t-1$, pero no en forma

recursiva, sino para calcular los valores de estos, para considerarse como iniciales, puesto que estos se almacenan (para posteriores cálculos) en un vector de dimensión predeterminado. Esto obliga a que se busque un método de aproximación para que pueda hacerse cálculos en forma recursiva.

Una de estas aproximaciones constituye el hacer una sola iteración con el θ estimado existente y usando los valores previos de ϵ y Φ como valores iniciales. Esto significa que $\epsilon(t, \delta(t-1))$ es aproximado por $\epsilon(t)$, el mismo que se calcula por:

$$\epsilon(t) = y(t) - yest(t)$$

y $yest(t)$ por (3.6), teniendo en cuenta que los parámetros a, b y c son los estimados al instante $t-1$, e.d., $\delta(t-1)$.

Introduciendo el vector:

$$\phi(t) = \begin{bmatrix} -y(t-1) \\ \vdots \\ -y(t-n_a) \\ u(t-1) \\ \vdots \\ u(t-n_b) \\ \epsilon(t-1) \\ \vdots \\ \epsilon(t-n_c) \end{bmatrix}$$

es posible escribir $\epsilon(t)$ como:

$$\epsilon(t) = y(t) - \delta^T(t-1) \phi(t) \quad (3.21)$$

Análogamente, se ve que la aproximación natural de $\Phi(t, \delta(t-1))$ es $\Phi(t)$ si se calcula por medio de:

$$\Phi(t) = -\sum_{i=1}^c c_i \Phi(t-i) + \phi(t) \quad (3.22)$$

Esta última expresión constituye una típica aproximación del gradiente, como se demuestra por (3.20).

Se obtiene, entonces el algoritmo recursivo que nos permite estimar los parámetros del modelo matemático seleccionado:

- $\epsilon(t)$ se calcula por (3.21). (3.23a)
- $\Phi(t)$ se calcula por (3.22). (3.23b)
- $R(t) = R(t-1) + \Phi(t) \Phi^T(t)$ (3.23c)
- $\delta(t) = \delta(t-1) + R^{-1}(t) \Phi(t) \epsilon(t)$ (3.23d)

Como se puede observar en (3.23c), la matriz de covarianzas debe ser invertida en cada iteración, en forma previa, para poder calcular el nuevo estimado de θ . Aplicando el lema de inversión de matrices se puede mejorar y agilizar los cálculos en el algoritmo si se hace:

$$p(t) = R^{-1}(t) \quad (3.24)$$

con lo cual se mejora $p(t)$ en forma directa. Para no dejar en vacío alguna inquietud del lector, el lema de inversión de matrices dice:
"Sean A, B, C y D matrices de dimensiones compatibles, tales que el producto BCD y la suma $A + BCD$ existen. Luego:

$[A+BCD]^{-1} = A^{-1} - A^{-1}B [DA^{-1}B + C^{-1}]^{-1}DA^{-1}$
 Haciendo:
 $A = p(t-1)$ $B = \Phi(t)$ $C = I$ $D = \Phi^T(t)$
 da: $p(t) = [p^{-1}(t-1) + \Phi(t) \Phi^T(t)]^{-1}$

$$p(t) = p(t-1) - \frac{p(t-1)\Phi(t)\Phi^T(t)p(t-1)}{1 + \Phi^T(t)p(t-1)\Phi(t)}$$

En el cálculo del error de predicción $\epsilon(t)$, se utiliza el estimado de θ al tiempo $(t-1)$, es decir, $\hat{\theta}(t-1)$, basándose en las medidas realizadas hasta el tiempo $t-1$, inclusive. Esto es necesario debido a que se usa $\epsilon(t)$ para mejorar θ en (3.23d). Sin embargo se utiliza $\epsilon(t)$ en $\hat{\theta}(t+1)$ para hallar $\Phi(t+1)$ en (3.22). Esta variable es necesaria sólo después de que se ha actualizado $\hat{\theta}(t)$. Se puede mejorar el error de predicción tomando la nueva información de $y(t)$ en cuenta, esto es, considerando $\hat{\theta}$ al instante t :

$$e(t) = y(t) - \delta^T(t)\hat{\theta}(t) \quad (3.25)$$

Este error se conoce con el nombre de residual y puede ser utilizado para formar $\hat{\theta}(t)$ con $e(t-1)$. El residual al instante t difiere del error de predicción puesto que utiliza los estimados de los parámetros del tiempo anterior a su determinación.

Introduciendo todas estas mejoras al algoritmo descrito anteriormente se tiene un nuevo algoritmo cuyo efecto es el de conseguir una mejor convergencia de parámetros, como se verá en el siguiente capítulo. A continuación se detalla el definitivo

ALGORITMO	
1)	Evaluar $y(t)$.
2)	Formar $\hat{\theta}(t)$ con y , u , e calculados al tiempo $t = t-1; i \geq 1$.
3)	$\epsilon(t) = y(t) - \delta^T(t-1)\hat{\theta}(t)$
4)	$\hat{\theta}(t) = -\sum_1^{i-1} c_i \hat{\theta}(t-i) + \hat{\theta}(t)$
5)	$p(t) = p(t-1) - \frac{p(t-1)\Phi(t)\Phi^T(t)p(t-1)}{1 + \Phi^T(t)p(t-1)\Phi(t)}$
6)	$\delta(t) = \delta(t-1) + p(t)\Phi(t)\epsilon(t)$
7)	$e(t) = y(t) - \delta^T(t)\hat{\theta}(t)$

4. PAQUETE DE PROGRAMAS

Las aplicaciones en línea requieren cierto tipo de lenguaje, en cambio, aquellas aplicaciones que son fuera de línea pueden incluso ser escritas en otro lenguaje. Por las proyecciones que tiene este algoritmo para su aplicación en línea se ha seleccionado al Lenguaje C como lenguaje de programación para el trabajo de la presente tesis.

El Lenguaje de Programación C nace como una constante de selección dentro del ámbito descrito arriba, pues constituye un programa de aplicación general que fue inicialmente diseñado para la programación de sistemas (compiladores, sistemas operativos, editores de textos, etc.), pero desde 1.872 ha dado pruebas satisfactorias en otras aplicaciones como por ejemplo: sistemas de bases de datos, sistemas de conmutación telefónica, análisis numérico, procesadores de textos, y para lo que nos interesa en el presente trabajo como son programas de ingeniería.

4.1 DIAGRAMA DE FLUJO DEL PROGRAMA

La estructura del paquete de programas desarrollado está de acuerdo al esquema que se presenta en la figura 3.2.1 (a), que se muestra a continuación:

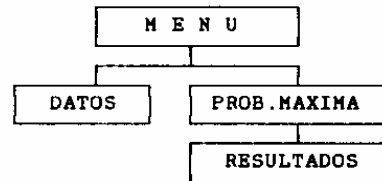


Fig. 3.2.1 (a)
 ESTRUCTURA DEL PAQUETE DE PROGRAMAS

Como se puede observar, el paquete se halla formado básicamente por cuatro programas entrelazados para brindar versatilidad al manejo de la información. Todos se hallan realizados en C, más el soporte que nos brinda el utilitario LOTUS 123, versión 2.x, o más reciente, para recuperar los resultados en forma gráfica.

La comunicación se realiza a través de archivos creados por cada programa y que se colocan en la unidad de almacenamiento del computador personal, sea esta disco duro o disco flexible según el caso. Cada programa recupera estos archivos para dar el procesamiento respectivo a esos datos. Se los puede reconocer a estos archivos de comunicación del paquete, por su extensión *.DAT. El formato de los datos es variable puesto que se almacenan caracteres como son números enteros, de punto flotante, etc., los cuales solamente pueden ser interpretados por el programa que los necesite. A todos estos se los puede denominar como archivos de datos de entrada, ya que son condiciones de inicialización que se deben cumplir en forma previa a la ejecución de uno de los programas.

Los archivos de los datos de salida se los puede reconocer por su extensión *.PRN, y son utilizados para la comunicación con el programa de resultados. Estos son recuperados por el utilitario LOTUS y luego de ser procesados se puede mirar un conjunto de curvas que corresponden a la convergencia de parámetros, cuya interpretación es objeto de estudio en esta tesis. Cabe anotar, que para ser posible mirar un gráfico en la pantalla de un PC, se requiere que este tenga la capacidad de generar gráficos, opción que no existe en todos los computadores personales. En tal caso, el resultado se podrá mirar

previamente en pantalla, y luego obtenerlo por impresor, si el hardware lo permite.

PROGRAMA MENU.*

El programa nos permite seleccionar la ejecución de cualquiera de los programas de datos, o de aplicación del método de probabilidad máxima, según la opción que escoja el usuario, a través de la digitación de una de las teclas propuestas por el programa. Este se ejecutará al inicializar todo el proceso de estudio de simulación. Asimismo permitirá concluir en forma normal dicho proceso.

PROGRAMA DATOS.*

El programa realiza tres tareas: reinicializa los datos para ejecutar el proceso de simulación, puede mirar los datos actuales en caso de ya existir los mismos, o permite salir para volver al menú inicial.

PROGRAMA RML.*

Contiene el desarrollo del algoritmo del método de probabilidad máxima. En él se han desarrollado variadas opciones, tanto las dirigidas para el mejor usuario, como las necesarias para la mejor transferencia de información desde y hacia las variadas del programa; por ejemplo se puede mencionar la facilidad de dosificar tanto la magnitud de la excitación que se haya escogido, como la magnitud del tipo de ruido o perturbación de salida seleccionados. Otro ejemplo sería la facilidad con la cual se ha realizado la implementación del programa, gracias a la versatilidad del lenguaje, al poder encomendar tareas particulares a determinadas funciones definidas en el programa, las cuales permiten no confundir el desarrollo del algoritmo dentro de la codificación en C del mismo.

Los resultados que se obtienen al correr este programa, se obtienen por pantalla según lo desee el usuario. Este entregará los resultados obtenidos en tantas iteraciones como se lo desee. Además existe la opción de mirar la convergencia de los parámetros estimados $\hat{\theta}(t)$, sirviéndonos del utilitario LOTUS, y adicionalmente, se puede entregar los datos por impresor. Todo esto a selección del usuario.

PROGRAMA DE RESULTADOS

En realidad no es un programa en C particularmente realizado para arrojar datos, sino más bien son funciones que permiten hacer esta tarea desde el programa descrito anteriormente. Como complemento de esto, y ahora si, en una forma ajena a estos programas se tiene una rutina de instrucciones realizadas en LOTUS, denominada MACRO, que permiten obtener el gráfico de convergencia de parámetros, en forma automática, como resultado de las iteraciones realizadas previamente. En forma adicional, el usuario queda en libertad de poder generarse más gráficos, puesto que los datos recogidos por el LOTUS no son solamente los que se refieren a los parámetros como tales, sino también que se dispone de la misma información de pantalla, pero ahora en una hoja electrónica, la cual nos permite hacer

algunas tareas adicionales para completar un análisis que el usuario desee culminar, por ejemplo, restar los valores del residual al ruido de salida, esto permitiría conocer cuan cerca, el residual se halla del segundo.

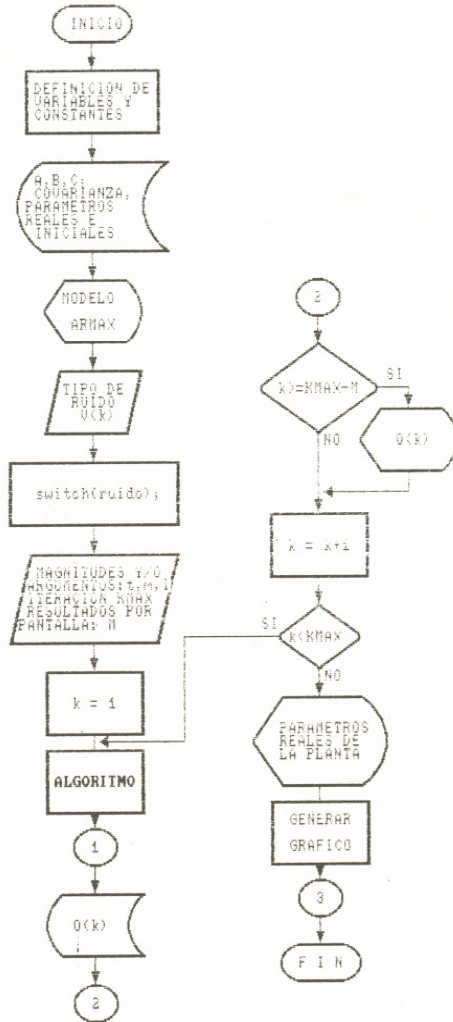


FIGURA 3.2.1 (b)
 DIAGRAMA DE FLUJO DEL PROGRAMA RML.*
 CODIFICACION CON LENGUAJE C

Este lenguaje se caracteriza por su concisión y por poseer un moderno flujo de control y estructuras de datos, además de contener un gran variedad de operadores. No es un lenguaje de muy alto nivel ni grande, sin especializarse en ninguna área particular de aplicación. C es un lenguaje de relativo bajo nivel, lo cual significa que trabaja con la misma clase de objetos que la mayoría de las computadoras (caracteres, números, direcciones), que

pueden ser combinados con los operadores aritméticos y lógicos, utilizados normalmente en las máquinas. Su carencia de restricciones y su generalidad lo hacen cada vez más eficaz y conveniente para muchas tareas que otros lenguajes, supuestamente más potentes. C contiene proposiciones (sentencias) de control de flujo de la información sencillas, secuenciales, de selección, de iteración, de bloques y subprogramas, pero no multiprogramación o corrutinas. Estos detalles hacen del Lenguaje C ser sencillo y compacto, fácil de aprender y de escribirse en poco espacio.

Una gran ventaja del C es su portabilidad, es decir, es independiente de la estructura particular de cualquier máquina. Con un pequeño cambio, un mismo programa puede funcionar en diferente computador.

SIMULACION

El trabajo se limita a simular las condiciones en el mismo computador, es decir, que el modelo seleccionado produce los valores de entrada y salida de la planta, para de allí proceder a la aplicación del algoritmo y obtener los resultados que interesa.

En lo referente a la simulación de los datos con el modelo seleccionado, cabe señalar que la excitación puede escogerse entre la función escalón, exponencial y señal binaria pseudo-aleatoria. Esta última, se genera con un registro de desplazamiento de 8 estados, con retroalimentación hacia el primero del resultado de una compuerta OR exclusiva cuyas entradas son los registros 2, 3, 4 y 8, todo esto en el ámbito de la lógica binaria. La implementación se completa transformando esta secuencia de longitud 255, en una señal pseudo-aleatoria (p.r.b.s.: pseudo-random binary signal) de dos niveles +V y -V más comunmente utilizada.

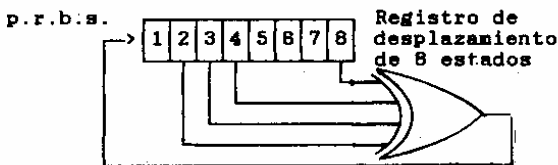


Figura 3.4.1
GENERADOR DE P.R.B.S.

Esta señal es un típico ejemplo de lo que constituye una excitación persistente, es decir, que dentro del dominio de frecuencia, la señal tiene un amplio espectro, en los dos lados del mismo.

La señal de entrada auxiliar del modelo, $v(t)$, también permite ser escogida entre 3 señales para su implementación, siendo estas: ruido blanco, p.r.b.s. y 0 (cero). La primera proviene de un arreglo cuya base es la función `random()` del C, y que consiste en la generación de números enteros en forma aleatoria. La última opción permite simular las condiciones de tipo determinísticas al método, dejando la facilidad de conocer el trabajo del algoritmo cuando un sistema es ideal.

5. RESULTADOS

Teniendo como base los conceptos fundamentales, y considerados necesarios, que se presentan en los dos primeros capítulos, se ha diseñado un paquete de programas, en Lenguaje C. Seleccionando modelos matemáticos, a manera de ejemplos, tanto de primer orden como de orden mayor, se ha probado el comportamiento del algoritmo, tanto para un proceso determinístico como para otros estocásticos.

Ejemplo 5a: La salida de un sistema tiene la forma:

$$y(t) = 0.81 y(t-1) + 0.95 u(t-1) + v(t) + 0.5 v(t-1) + 0.25 v(t-2)$$

Para el caso determinístico este sistema de primer orden converge a un valor de 5 para una entrada tipo escalón. En el caso mostrado en la figura 5a, la perturbación a la salida tiene características de ruido aleatorio en un 100%, y se puede observar que para iteraciones superiores a 200, se tiene una suave aproximación a valores estables de los parámetros estimados. Se debe indicar que la convergencia del parámetro "b" es suave respecto de una similar con excitación tipo escalón, pues en el presente caso la excitación es la señal p.r.b.s..

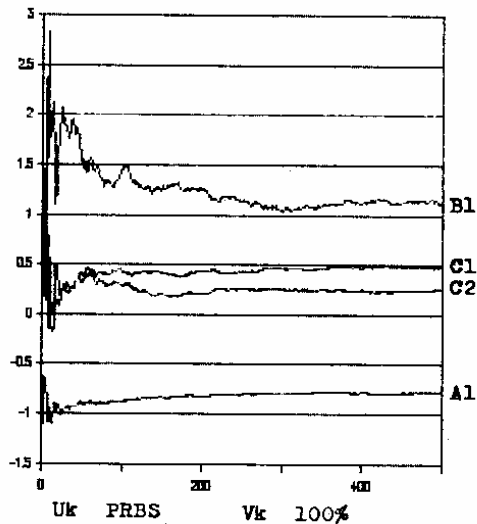


Fig. 5a

Ejemplo 5b: La salida de un sistema tiene la forma:

$$y(t) = -1.1 y(t-1) + 0.37 y(t-2) + 0.55 y(t-3) + 0.05 y(t-4) + 0.75 u(t-1) + v(t) + 0.21 v(t-1)$$

Este modelo corresponde a un sistema de cuarto orden, con ruido correlacionado a la salida. Converge a 0.65 para la entrada escalón, es decir, en el caso determinístico. Para el caso estocástico, con un ruido blanco a la salida del 100%, los parámetros convergen hacia otros

valores, bastante desviados de los parámetros reales de la planta. En cambio, si la señal de excitación se la cambia por la p.r.b.s., la convergencia es como la mostrada en la figura 5b. La misma escala del gráfico nos permite comparar la gran estabilidad que poseen los estimados a partir de la iteración 250, observando que su variación es bastante pequeña.

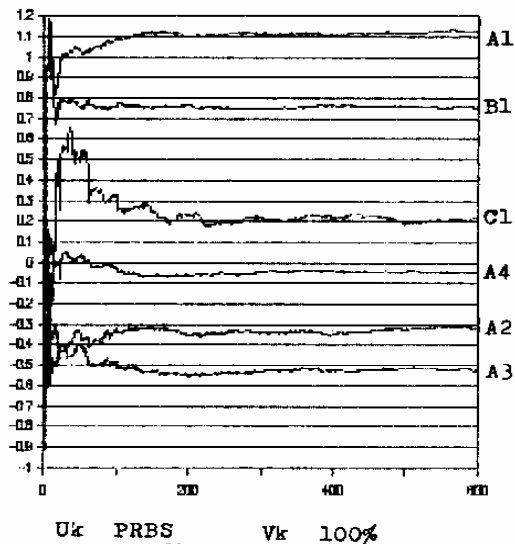


Fig. 5b

A lo largo de las pruebas realizadas, y de los ejemplos más significativos aquí presentados, se inclinan los resultados a seleccionar cada vez más la señal indicada, como señal excitatriz de una planta a simularse, debido a que los parámetros convergen con mayor exactitud hacia los valores reales de los parámetros de un sistema. Con esta afirmación lo que se obtiene es una comprobación de lo que se conoce como señal persistente de excitación de un sistema, siendo la p.r.b.s. la mejor señal para proceder a la identificación de parámetros, en lo que lleva de utilizarse el presente método.

CONCLUSIONES

En el presente trabajo se ha desarrollado un algoritmo de identificación de procesos, basándose en el método de probabilidad máxima. En la etapa de identificación se selecciona el modelo ARMAX, como modelo matemático para que caracterice la dinámica de un proceso, o expuesto de otra manera, para que se pueda predecir el comportamiento del mismo.

Dada la naturaleza estocástica del problema propuesto, la revisión de conceptos probabilísticos contribuye a clarificar el desarrollo del algoritmo de probabilidad máxima. Luego de obtener un algoritmo recursivo que permita su implementación en un ordenador, se procede a su codificación utilizando el lenguaje C de programación. El proceso de simulación del método desarrollado, lleva a resultados que permiten verificar la estructura del modelo y la dinámica del proceso, es decir, se

comprueba el funcionamiento del algoritmo, y el problema de identificación se obtiene en forma recursiva, para lo cual se basa el estudio en los mínimos cuadrados recursivos (determinísticos). En otras palabras, los resultados que se han obtenido son satisfactorios, permitiendo comprobar la necesidad de desarrollar un método que elimine la desviación de parámetros estimados, como se expone en 1.4., pues, el método de mínimos cuadrados tiene este problema en el proceso de identificación cuando el parámetro c_1 es diferente de cero.

Si bien el método recursivo de probabilidad máxima se lo reconoce dentro de los métodos de predicción de error, la utilización de este error de predicción como herramienta para la construcción del algoritmo de probabilidad máxima, impide una rápida y más aproximada convergencia en los parámetros estimados, independientemente del valor de covarianza inicial o de valores iniciales de parámetros estimados, diferentes de cero. En lugar de esto, si el algoritmo trabaja con el residual en la formación del vector $\hat{\theta}$, los resultados se optimizan. Luego, este problema se elimina al introducir un algoritmo mejorado, el cual contiene el residual, constituyéndose en una variable más actualizada dentro del algoritmo, y que sustituye al error de predicción cuando el algoritmo actualiza sus variables para iniciar una nueva iteración. No se elimina el concepto del error de predicción al estimar los nuevos parámetros, lo que si se hace es sustituir a este por el residual, para la redefinición de las demás variables del algoritmo.

Otro detalle importante que se deduce, es la utilización del residual como fuente de información, esto es, que analizando el comportamiento de esta variable se puede tener una idea bastante aproximada de una perturbación que una planta podría tener a su salida, pues del análisis realizado a lo largo de todas las pruebas, el seguimiento de esta variable a las perturbaciones simuladas es prácticamente fiel. Esto constituye de gran importancia, si se enfoca el trabajo del algoritmo para estudios que se lleven a cabo en tiempo real.

La implementación del algoritmo con lenguaje C, permite varias ventajas al método: portabilidad del programa, es decir, puede correrse en muchos ordenadores, sin mayor limitación del hardware; velocidad de ejecución, pues cada día existen procesadores más veloces, lo cual permite agilidad en la obtención de los resultados del programa, y por ende amplía la gama de procesos a identificarse, es decir, procesos más rápidos.

Debe aclararse que los tiempos de ejecución dependen tanto del número de operaciones que deba realizar el algoritmo en la tarea de estimación paramétrica, como de las opciones que el usuario requiera, es decir, cuánta información desea por pantalla, puesto que a mayor procesamiento de información, mayor es la demora para terminar de arrojar resultados en un programa. Mientras más parámetros se estimen, mayor será la cantidad de

operaciones que se deban realizar, pues el orden y la dimensión del modelo y de las matrices se han incrementado, respectivamente.

Resulta interesante presentar este trabajo como base para estudios e implementación de sistemas expertos, aquellos capaces de aprender de sí mismos, y en cuya base se pueden analizar una gran variedad de señales. Esto se traduce en que los datos interpretados convenientemente permiten reconocer el proceso del cual provienen y conseguir una idea bastante aproximada del mismo; además esto redundará en una mejor comprensión del proceso, la posibilidad de influir en él, optimizarlo, etc., etc..

6. BIBLIOGRAFIA

CRUZ MORA, EDISON RICARDO.- Tesis: "Identificación de Procesos por el Método Recursivo de Probabilidad Máxima". E.P.N., 1.989

LJUNG, L. SÖDERSTRÖM, T.- Theory and Practice of Recursive Identification. MIT Press, Londres 1.987

HELSTROM, Carl W.- Probability and Stochastic Processes for Engineers. Macmillan Pub.Co., N.Y. 1.984

GOODWIN, Graham C. KWAI-SANG-SIN.- Adaptive Filtering, Prediction and Control. Prentice-Hall Inc., N.J. 1.984

FRANKLIN, Gene F. POWELL, David J.- Digital Control of Dynamic Systems. Addison-Wesley Pub.Co.Inc., Filipinas, 1.980

BENDAT, Julius S. PIERSON, Allan G.- Random Data: Analysis and Measurements Procedures. Wiley Interscience, N.Y. 1971

PAPULIS, Athanasios.- Probability, Random Variables, and Stochastic Processes. McGraw-Hill, N.Y., 1.984

BROGAN, William L.- Modern Control Theory. Quantum Pub.Inc..N.Y., 1.974

Maximum Likelihood and Prediction Error Methods, K.J. Aström, AUTOMATICA Vol.16 p.551-574

Least Squares Parameter Estimation, V.Streic, AUTOMATICA Vol.16 p.535-549

Correlation Methods, K.R. Goodfrey, AUTOMATICA Vol.16 p.527-533

Theory and Applications of Self-Tuning Regulators, K.J.Aström, U.Borisson, L.Ljung, B.Wittenmark, AUTOMATICA Vol.13 p.457-461

CORDERO A., Carlos P.- Tesis: "Mínimos Cuadrados Generalizado". E.P.N., 1.986

KERNIGHAN, Brian W. RITCHIE, Dennis M.- El Lenguaje de Programación C. Prentice-Hall, México, 1.986

WAITE, M. PRATA, S. MARTIN, D.- Programación en C: Introducción y conceptos avanzados. Anaya, Madrid, 1.985

The C Language: The State of C, B.Kernighan and D.Ritchie; A Better C, Bjarn Stroustrup; It's an attitude, Jonathan Linowes; BYTE, Agosto 1988, Vol.13 No.8, pgs.202-227

The Great C Compiler War, CHAOS MANOR MAIL, BYTE, Octubre 1987, Vol.12 No.11, p.264

ROJAS, V. MACATO, J.- Técnica de Fluigramas, Cicetronic, Quito, 1.980

SELBY, Samuel M.- Standard Mathematical Tables. C.R.C.Press Inc., U.S.A., 1.974

7. BIOGRAFIA DEL AUTOR



Edison Ricardo Cruz Mora, nacido en Quito, el 15 de Julio de 1.960. Título de Ingeniero en Electrónica y Telecomunicaciones, E.P.N. 1.989. Se ha desempeñado como Jefe del departamento técnico de EPSON en 1.987 y en Asistencia Técnica en PHILIPS ECUADOR en 1.988. Actualmente trabaja en SITE, oficina de prestación de servicios.