

# IDENTIFICACION PARAMETRICA DISCRETA

**Ing. Byron Zárate Mora**  
**ELECTRONICA Y CONTROL**  
**PROCELEC CIA. LTDA.**

**Ing. Patricio Burbano Romero**  
**MSc. SYSTEMS AND CONTROL**  
**ESCUELA POLITECNICA NACIONAL**

## ABSTRACT

The most important discret parametric identification methods are reviewed, specially both of them best linear unbiased estimator (BLUE) and maximum of Likelihood. A program using C language using Borland C and ProtoGen C+ under Windows enviroment is developed to identify parametrics models with simulation and at real time with electric circuits prototypes and scale prototypes.

## RESUMEN

Los métodos más importantes para identificación paramétrica discreta son revisados, especialmente el método del mejor estimador lineal no desviado y el método de Máximo de Likelihood. El programa se desarrolló usando Borland C, y ProtoGen C+, en el ambiente Windows para identificar modelos en simulación y en tiempo real a través de la utilización de circuitos eléctricos y prototipos a pequeña escala.

## 1. INTRODUCCION

En la actualidad la tendencia del control es hacia un control computarizado en donde las acciones de identificación y control son realizadas en línea, es decir el computador forma parte del lazo de identificación y control. Las técnicas de identificación se utilizan hoy en día debido a la complejidad de los sistemas y a la presencia significativa de ruido por lo que la aplicación de técnicas clásicas de modelación resultan limitadas.

Existen dos métodos de obtener el modelo matemático o analítico de un sistema: estructuralistas y globalistas.

El método estructuralista parte del conocimiento de la estructura física y los componentes del sistema, se conoce como modelación. Se parte del conocimiento de las leyes físicas, químicas, y de otras ciencias así como de principios de ingeniería; y, se los aplica a los componentes que conforman un sistema a través de las leyes de Kirchoff, de Newton, de conservación de masa y energía (balance de materia y energía) para obtener un modelo dinámico del mismo.

Los métodos globalistas no requieren del conocimiento de la estructura ni de los componentes de los sistemas. Considera al sistema como un bloque (caja negra) al cual se le aplica una entrada y se obtiene una salida o respuesta. Entonces el modelo se obtiene a partir de mediciones de entrada y salida (que reflejan la dinámica del sistema) y se procesan dichas mediciones mediante un paquete computacional, considerando ciertas condiciones preestablecidas. A este proceso se conoce con el nombre de identificación.

El método de identificación paramétrica que se desarrolla en este trabajo se puede realizar fuera de línea (simulación) ó en línea (tiempo real) para obtener modelos para control discreto.

El problema de identificación paramétrica discreta esta basado en obtener los parámetros del modelo conociendo los valores de entrada y salida del sistema (conociendo los estados). Se utiliza el modelo ARMA (Auto Regressive Moving Average) que es una representación canónica en la cual se tiene el mínimo número de parámetros necesarios para describir totalmente a un sistema.

El modelo ARMA se puede expresar como:

$$y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_1 u(k-r) + \dots + b_m u(k-m) \quad (1.1)$$

donde  $r$  es el retardo entre la salida y la entrada.

La ecuación (1.1) en forma simplificada está dada por

$$y(k) = x(k)\Theta(k) \quad (1.2)$$

en donde:

$$x(k) = [-y(k-1) \quad \dots \quad -y(k-n) \quad u(k-r) \quad \dots \quad u(k-m)] \quad (1.3)$$

$$\Theta(k) = \begin{bmatrix} a_1 \\ \vdots \\ a_n \\ b_r \\ \vdots \\ b_m \end{bmatrix} = [a_1 \quad \dots \quad a_n \quad b_r \quad \dots \quad b_m]^T \quad (1.4)$$

Correspondencia a :

Zárate Byron Ing.  
PROCELEC CIA LTDA.  
Teléfono : 480-375 / 370 - 161  
email : procelec@uio.telconet.net

El vector fila  $x(k)$  contiene la información de los valores anteriores de salida  $y(k)$ , y de entrada  $u(k)$  que constituyen los estados del sistema, por lo tanto el problema radica en base a estas mediciones identificar o estimar los parámetros del vector  $\Theta(k)$ .

Existen algunos algoritmos de identificación paramétrica discreta entre los que se puede señalar, los métodos determinísticos que son: mínimos cuadrados ordinarios, mínimos cuadrados recursivos, etc. Además existen métodos más elaborados, para en presencia de perturbaciones tanto en la entrada como en la salida, llegar a identificar los verdaderos valores de los parámetros; estos métodos son estadísticos entre los que se tiene que el más importante es el máximo de Likelihood.

## 2. METODOS DE IDENTIFICACION PARAMETRICA DISCRETA

Los métodos de identificación paramétrica se basan en el método de mínimos cuadrados, método que radica en hallar los parámetros de un sistema en base a la minimización del error.

### 2.1. Mínimos cuadrados ordinarios (MCO).

El método de Mínimos Cuadrados Ordinarios (MCO) es un algoritmo para procesamiento de datos en lote; esto es, no es recursivo, sin embargo es el algoritmo de partida para el estudio de la identificación de sistemas mediante un modelo ARMA a ecuación de diferencias.

Al analizar la ecuación (1.2) que describe a un sistema, se observa que para tener una adecuada identificación los parámetros estimados  $\hat{\Theta}(k)$ , deben tender a los verdaderos valores de los parámetros  $\Theta^o(k)$ , es decir se busca que

$$\left[ \hat{\Theta}(k) - \Theta^o(k) \right] \rightarrow 0$$

Sin embargo al no conocer el valor de  $\Theta^o(k)$  no es posible conocer el error en los parámetros. La ecuación (1.2) se cumple para los parámetros verdaderos  $\Theta^o(k)$ , pero para los valores estimados  $\hat{\Theta}(k)$  tiene un error que depende tanto de la medición  $y(k)$  como de los parámetros  $\Theta(k)$ , por lo cual, en este caso se tiene un error de la ecuación (1.2) y puede escribirse como:

$$y(k) = x(k)\hat{\Theta}(k) + e(k, \Theta) \quad (1.5)$$

La ecuación (1.5) es la base para el desarrollo del algoritmo de identificación basado en el modelo ARMA. La identificación de un sistema mediante el método de MCO, para  $k$  mediciones está dada por la expresión:

$$\hat{\Theta}(k) = \left( X^T(k)X(k) \right)^{-1} X^T(k)Y(k) \quad (1.6)$$

La ecuación (1.6) permite hallar el valor de los parámetros de la ecuación (1.2) al utilizar el método de los Mínimos Cuadrados Ordinarios.

### 2.2 Mínimos cuadrados recursivos (MCR).

En general el MCO tiene algunos inconvenientes para ser utilizado, entre los cuales se puede señalar:

- Para encontrar los parámetros estimados es necesario invertir la matriz  $(X^T X)$ .
- Los parámetros se calculan de un lote de datos, lo que hace que no se pueda ver variaciones de los parámetros al ir agregando mediciones.
- La cantidad de memoria utilizada depende del número de mediciones.
- No es implementable en forma recursiva en línea.
- En general el método de MCO es un algoritmo básico de identificación de parámetros, por lo tanto no permite mayores refinamientos.

Por lo tanto se trata de desarrollar un algoritmo recursivo como una variante del método de MCO; señalar algunas características matemáticas y la posibilidad de implementarlo computacionalmente.

### ALGORITMO RECURSIVO

Cuando se desarrolla el método de mínimos cuadrados recursivos, se parte del algoritmo inicial de mínimos cuadrados ordinarios al instante  $k$ , y lo que se hace es añadir una medición. En la ecuación (1.6) del vector de parámetros al instante  $k$  está dada por:

$$\hat{\Theta}(k) = \left[ X^T(k)X(k) \right]^{-1} X^T(k)Y(k)$$

se añade información al instante  $(k+1)$  en la matriz de información y en el vector de la salida, por lo que se expresa la matriz de información y el vector de salida como:

$$X(k+1) = \begin{bmatrix} x(n) \\ x(n+1) \\ \vdots \\ x(k) \\ \text{-----} \\ x(k+1) \end{bmatrix} = \begin{bmatrix} X(k) \\ \text{-----} \\ x(k+1) \end{bmatrix} \quad (1.7)$$

$$Y(k+1) = \begin{bmatrix} y(n) \\ y(n+1) \\ \vdots \\ y(k) \\ \text{-----} \\ y(k+1) \end{bmatrix} = \begin{bmatrix} Y(k) \\ \text{-----} \\ y(k+1) \end{bmatrix} \quad (1.8)$$

Reemplazando las ecuaciones (1.7) y (1.8) en la ecuación (1.6) de parámetros estimados se tiene:

$$\hat{\Theta}(k+1) = [X^T(k+1)X(k+1)]^{-1} X^T(k+1)Y(k+1) \quad (1.9)$$

al desarrollar la matriz de información y el vector de salida en función de sus valores anteriores se puede llegar a tener que el vector de parámetros estimados está dado por las ecuaciones :

$$\varepsilon(k+1) = y(k+1) - x(k+1)\hat{\Theta}(k) \quad (1.10)$$

$$L(k+1) = \frac{P(k)x^T(k+1)}{1 + x(k+1)P(k)x^T(k+1)} \quad (1.11)$$

$$P(k+1) = \left( I - \frac{P(k)x^T(k+1)x(k+1)}{1 + x(k+1)P(k)x^T(k+1)} \right) P(k) \quad (1.12)$$

donde :

- $\varepsilon(k+1)$  es el error de predicción
- $L(k+1)$  es una matriz de ganancia
- $P(k+1)$  es la matriz de covarianza

reemplazando las ecuaciones (1.10) y (1.11) en la ecuación (1.9) se tiene:

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + L(k+1)\varepsilon(k+1) \quad (1.13)$$

La ecuación (1.13) muestra la naturaleza recursiva del método, ya que los nuevos valores de  $\hat{\Theta}(k+1)$  se hallan

en función del anterior  $\hat{\Theta}(k)$  más una corrección en función del error.

El método de mínimos cuadrados recursivos tiene algunas ventajas con respecto al método de mínimos cuadrados ordinarios:

- Se evita la inversión de la matriz  $(X^T X)$ , pues se reduce a la inversión de un escalar.

- Es un algoritmo adecuado para implementarlo en línea debido a que utiliza la información anterior y lo que se hace es corregir el valor de los parámetros estimados al ir incrementando información.

- La cantidad de memoria que se necesita para hallar los parámetros no depende del número de mediciones, y se mantiene fija a lo largo de todo el proceso de identificación.

El dar igual importancia a todas las mediciones que se adquieren para estimar los parámetros en un principio puede ser aceptable, sin embargo a medida que se van adquiriendo más valores de entrada - salida se debe tener en cuenta algún criterio para dar mayor peso a las últimas mediciones con respecto a las primeras, por cuanto las últimas reflejan de mejor forma la dinámica de la planta, sobre todo si existen cambios lentos de la misma. Se debe entonces implementar un mecanismo para ponderar las mediciones.

La forma más sencilla y adecuada de ponderar las mediciones es mediante un factor escalar  $\gamma$ , llamado factor de olvido.

Minimizando el error de la ecuación (1.5) considerando el factor de olvido se llega a una forma similar que en el caso de mínimos cuadrados recursivos para hallar el valor de los parámetros estimados.

El hecho de incluir el factor de olvido en los algoritmos de identificación paramétrica discreta evita la posibilidad de inestabilidad numérica, la cual se presenta por la tendencia de que los valores de  $\hat{\Theta}(k)$  tiendan a un valor constante debido a que la matriz de covarianza  $P(k)$  tiende a cero, además el factor de olvido al evitar que la matriz de covarianza  $P(k)$  tienda a cero permite rastrear o actualizar cualquier cambio o variación lenta de los parámetros, si la variación es fuerte, entonces habrá que esperar que el algoritmo trabaje con las nuevas mediciones que reflejen la nueva dinámica de la planta.

### 2.3 Mínimos cuadrados estocásticos (MCS)

Al analizar el algoritmo de identificación paramétrica discreta mediante el método de los mínimos cuadrados

recursivos, no se comentó nada acerca de la posibilidad de que los datos tomados fueran contaminados por ruido, y de como esto afectaría al algoritmo.

A continuación se considera el algoritmo de mínimos cuadrados estocásticos se tiene que:

$$y(k) = a^T \Theta^o(k) + v(k) \quad (1.14)$$

en términos de matriciales se tiene:

$$Y(k) = A(k) \Theta^o(k) + V(k) \quad (1.15)$$

donde:  $A(k)$  es la matriz de información.

$v(k)$  como una variable aleatoria de tipo ruido blanco con media cero y covarianza conocida, por lo que se tiene:

$$\begin{aligned} E[v(k)v(j)] &= \sigma^2 \quad (k=j) \\ &= 0 \quad (k \neq j) \\ E[V(k)V^T(k)] &= \sigma^2 I \end{aligned} \quad (1.16)$$

De manera similar que para el caso de los mínimos cuadrados recursivos, en los mínimos cuadrados estocásticos el problema es encontrar  $\Theta(k)$ , tal que minimice la ecuación de error, desarrollando la solución se tiene:

$$\hat{\Theta}(k) = (A^T(k)A(k))^{-1} A^T(k)Y(k) \quad (1.17)$$

Se debe indicar que aunque se utilice los verdaderos valores de  $\Theta^o$  no se debe esperar obtener un error igual a cero debido a los efectos del ruido por lo que se tiene que ampliar el concepto de lo que constituye una buena estimación.

Un buen estimador debe tener las siguientes características:

- Consistencia.
- No desviación.
- Mejor estimador lineal (BLUE).

Las características anteriores se cumplen para algoritmo de MCS por lo que se concluye que el método de mínimos cuadrados recursivos es valido o sea, consistente, no desviado y el mejor estimador lineal aun en presencia de ruido blanco al cual se le puede aproximar el ruido presente en procesos industriales.

### 2.3.1 Mínimos cuadrados generalizados (MCG)

Este método basa la identificación de parámetros en el modelo ARMAX, definido como:

$$(1+A(z))y(k) = z^{-1}B(z)u(k) + \xi(k) \quad (1.18)$$

Todo proceso estocástico (ruido coloreado o correlacionado)  $\xi(k)$  puede obtenerse haciendo pasar ruido blanco  $v(k)$  por el filtro  $\frac{F(z)}{G(z)}$ . Para el caso de

mínimos cuadrados generalizados se tiene que el filtro está definido como  $F(z) = 1$  y  $G(z) = 1+C(z)$ .

Por lo que  $\xi(k) = \frac{v(k)}{1+C(z)}$  y ( $c_0 = 1$ ), y  $v(k) = (1+C(z))\xi(k)$ , entonces se llega a:

$$(1+A(z))y_e(k) = z^{-1}B(z)u_e(k) + v(k) \quad (1.19)$$

donde:

$$\begin{aligned} y_e(k) &= (1+C(z))y(k) \\ u_e(k) &= (1+C(z))u(k) \end{aligned}$$

Por lo tanto la ecuación (1.19) es un problema de Mínimos Cuadrados Recursivos pero en las variables  $y_e(k)$ ,  $u_e(k)$  en donde se puede obtener los parámetros de  $A$  y  $B$ . Sin embargo se mide  $y$ ,  $u$  con lo que es necesario identificar  $C(z)$  para obtener  $y_e(k)$ ,  $u_e(k)$ , que es otro caso de identificación de MCR, es decir se filtra la información actual a través de  $C(z)$ . A continuación se examina una variante de este método que es más sencillo de implementar computacionalmente, llamado mínimos cuadrados extendidos.

### 2.3.2 Mínimos cuadrados extendidos (MCE)

Una variante del método MCG se tiene si el ruido se define como:

$$\xi(k) = (1+C(z))v(k) \quad (1.20)$$

por lo que si se reemplaza (1.20) en el modelo general de identificación se tiene:

$$\begin{aligned} (1+A(z))y(k) &= z^{-1}B(z)u(k) + (1+C(z))v(k) \\ y(k) &= \frac{z^{-1}B(z)}{1+A(z)}u(k) + \frac{1+C(z)}{1+A(z)}v(k) \end{aligned} \quad (1.21)$$

entonces lo que se hace es extender el vector de parámetros a los coeficientes  $C(z)$  y extender el vector de información al ruido blanco  $v(k)$ , y se tiene:

$$y(k) = x(k)\Theta(k-1)$$

donde se define el nuevo vector de información como:

$$x(k) = \begin{pmatrix} -y(k-1) & \dots & -y(k-n) & u(k-r) & \dots & u(k-m) \\ & & & v(k-1) & \dots & v(k-p) \end{pmatrix} \quad (1.22)$$

$$\Theta(k) = (a_1 \dots a_n \quad b_r \dots b_m \quad c_1 \dots c_p) \quad (1.23)$$

sin embargo como no se conoce el ruido blanco  $v(k)$  en el proceso se aproxima por el error de predicción que se lo puede calcular como:

$$\varepsilon(k) = y(k) - x(k)\Theta(k-1) \quad (1.24)$$

Al desarrollar la ecuación (1.24) se tiene que:

$$\varepsilon(k) = y(k) + A(z)y(k) - z^{-r}B(z)u(k) - C(z)\varepsilon(k) \quad (1.25)$$

de la ecuación (1.25) el error de predicción puede ser expresado como:

$$\varepsilon(k) = \frac{1}{1+C(z)} \left( (1+A(z))y(k) - z^{-r}B(z)u(k) \right) \quad (1.26)$$

en la ecuación (1.26) se observa que la mediciones de entrada y salida son filtradas a través del filtro  $1+C(z)$ . Entonces a medida que  $A(z) \rightarrow A^o(z)$ ,  $B(z) \rightarrow B^o(z)$ ,  $C(z) \rightarrow C^o(z)$ , se tiene que  $\varepsilon(k) \rightarrow v(k)$ .

### 2.4 Máximo de Likelihood (Máximo de Verosimilitud MLH)

El Máximo de Likelihood es un método estadístico (probabilístico) pues se manejan variables aleatorias en forma de proceso estocástico.

Para analizar procesos estadísticos es necesario tener modelos estocásticos, tal como el modelo ARMAX para la correcta identificación de los parámetros del modelo.

Cuando se tiene procesos estocásticos una buena alternativa es la utilización de un método estadístico, para encontrar parámetros y de esta manera garantizar resultados satisfactorios a costo de aumentar la complejidad y el tiempo de computación en los algoritmos.

### FUNCIÓN DE LIKELIHOOD

Si se considera un proceso estocástico gaussiano  $X$ , que tiene una distribución de Gauss, en donde la función densidad de probabilidad está dada por la ecuación:

$$f_x(\xi) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2}\frac{(\xi-\mu)^2}{\sigma^2}} \quad (1.27)$$

en donde  $\sigma^2$  (varianza),  $\mu$  (valor medio), describen completamente a  $f_x$  para un proceso escalar.

Para la identificación de procesos por el método de verosimilitud se toma la función de densidad de probabilidad porque es un proceso estocástico, pero se hace que esta función de densidad de probabilidad dependa de los parámetros. Entonces:

$$L(y, \Theta) = f(y, \Theta) \quad (1.28)$$

donde  $L$  es la función densidad de probabilidad o función de Likelihood.

Se utiliza el cambio de notación para expresar el hecho de que lo que se mide es  $y$  a la salida;  $\Theta$  son los parámetros desconocidos del modelo y  $L$  la función densidad de probabilidad se hace depender de  $\Theta$  para que en el algoritmo de identificación se maximice la función densidad de probabilidad y lo mejor que se puede hacer es obtener los parámetros con el máximo de probabilidad para el proceso estadístico.

Para construir la función de Likelihood se utilizan los errores de las mediciones, al utilizar los errores de las mediciones se pone la función de Likelihood en función de  $\Theta$ . Sea el modelo que se está identificando:

$$Y = X\Theta^o + \xi \quad (1.29)$$

en donde el error está dado por el ruido y la exactitud de los parámetros  $\Theta$ . Para los verdaderos valores de  $\Theta^o$  se tiene que el ruido  $\xi$  se puede escribir como:

$$\xi = Y - X\Theta^o \quad (1.30)$$

por lo que la función de probabilidad se puede escribir como  $f(y, \Theta^o)$  que significa densidad de probabilidad del proceso estocástico dado  $\Theta^o$  así:

$$f(y, \Theta^o) = \frac{1}{(2\pi\sigma)^{m/2}} e^{-\frac{1}{2\sigma^2} \left\{ (y - X\Theta^o)^T (y - X\Theta^o) \right\}} \quad (1.31)$$

donde:  $Y$  es el vector de observaciones.  
 $X$  es la matriz de información.  
 $\Theta^o$  vector de parámetros.  
 $m$  es el número de mediciones.

Se genera la función de Likelihood sustituyendo el valor de  $\Theta^0$  por el de  $\Theta$  y entonces se tiene la función L como:

$$L(y|\Theta) = \frac{1}{(2\pi\sigma)^{m/2}} e^{-\frac{1}{2\sigma^2}\{(y-x\Theta)^T(y-x\Theta)\}} \quad (1.32)$$

Si se maximiza esta función de probabilidad en función de los parámetros  $\Theta$  que se quiere identificar se llega a el resultado similar al encontrado al desarrollar el método de mínimos cuadrados determinísticos, y se tiene:

$$\frac{1}{2\sigma^2} (X^T X \Theta - X^T Y) = 0 \quad (1.33)$$

Con esto se concluye que el Método de Likelihood o de Probabilidad Máxima es en esencia Mínimos Cuadrados pero desde un punto de vista probabilístico, en el que se utiliza la función densidad de probabilidad.

### ALGORITMO DE PROBABILIDAD MÁXIMA

Se tiene el modelo ARMAX, que se utiliza para desarrollar el algoritmo de Probabilidad Máxima.

$$y(k) + \sum_{i=1}^n a_i y(k-i) = \sum_{i=1}^m b_i u(k-i) + v(k) + \sum_{i=1}^p c_i v(k-i) \quad (1.34)$$

lo que se trata es de encontrar un algoritmo que permita encontrar los parámetros del modelo en función de los valores anteriores de  $y(k)$  y  $u(k)$ .

Al instante  $k$ ,  $y(k-i)$ ,  $u(k-i)$  son conocidas pues corresponden a los valores (mediciones) anteriores de  $u$ , y entonces una buena aproximación de  $v(k-i)$  puede calcularse a partir de  $u$ , y.

Desarrollando la ecuación (1.34) para obtener el error de predicción se llega a:

$$e(k) = y(k) - \sum_{i=1}^n a_i y(k-i) - \sum_{i=1}^m b_i u(k-i) + \sum_{i=1}^p (c_i - a_i) y(k-i) \quad (1.35)$$

Esta ecuación no es lineal para los parámetros  $\Theta$ , por cuanto existen productos de los parámetros en el último término, por consiguiente no se puede realizar el algoritmo de minimización en forma analítica, y en consecuencia se debe plantear un método numérico alternativo que resuelva la ecuación (1.35) y se llega a determinar que las ecuaciones de identificación están dadas por:

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + P(k+1) x(k) \varepsilon(k+1) \quad (1.36)$$

$$P(k+1) = \left( I - \frac{P(k)x^T(k+1)x(k+1)}{1+x(k+1)P(k)x^T(k+1)} \right) P(k) \quad (1.37)$$

$$L(k+1) = \frac{P(k)x^T(k+1)}{1+x(k+1)P(k)x^T(k+1)} \quad (1.38)$$

$$\varepsilon(k+1) = y(k+1) - X(k+1)L(k+1)\hat{\Theta}(k) \quad (1.39)$$

$$x(k+1) = XL[0][i] = -\sum_{i=1}^p c_i XL[0][k-i] + X(k+1)L(k+1) \quad (1.40)$$

$$\eta(k+1) = y(k+1) - x(k+1)\hat{\Theta}(k+1) \quad (1.41)$$

donde:

$\eta(k+1)$  residual

$x(k+1)$  vector de datos filtrados

XL matriz de datos filtrados a través de los coeficientes  $c_i$  actualizados.

X(k+1)L(k+1) vector de información con el residual.

### 3. PROGRAMA DE IDENTIFICACION PARAMETRICA

El software que se desarrolló en este trabajo está enfocado a trabajar a nivel de simulación así como también en tiempo real.

Entre los compiladores más importantes para el desarrollo de programas en C, se tiene el *BORLAND C++*, sin embargo para el desarrollo de aplicaciones en ambiente Windows se tiene software adicional de programación que convierte algunas tareas comunes de programación en un desarrollo gráfico del mismo, para estas tareas se utiliza el programa *ProtoGen+*.

Por lo tanto para desarrollar el programa de identificación se utilizó los 2 paquetes anteriores que son:

- *ProtoGen+*; ver 4.2 1994-1995.

- *BORLAND C++ FOR WINDOWS*; ver 3.1 1993.

Para el correcto funcionamiento del software desarrollado es necesario tener los requerimientos mínimos para que la plataforma Windows versión 3.1 funcione adecuadamente, los mismo que son suficientes para que el software de identificación paramétrica discreta trabaje correctamente. Estos requerimientos son:

- Computador 386 o superior.

- Sistema Operativo MS-DOS ver 5.0 ó posterior y la plataforma Windows 3.1.

- Memoria mínima 4 MB.

Para un trabajo a nivel de tiempo real es necesario además que se posea la tarjeta de adquisición de datos, que en este caso es la DAS-128.

El lenguaje C es adecuado para crear aplicaciones que funcionen bajo el ambiente Windows, en este lenguaje se puede tener básicamente 2 filosofías de desarrollo de software, la primera (que se utiliza en este trabajo) es crear aplicaciones utilizando el sistema de mensajes y la segunda utilizando una programación orientada a objetos.

El programa "IPD.EXE" se desarrolló utilizando el sistema de mensajes. Un mensaje desde el punto de vista de una aplicación se puede considerar como una notificación de que se ha producido algún suceso de interés que puede o no requerir de una acción específica. Los mensajes en Windows pueden provenir de 4 fuentes que son: el mismo Windows, el propio programa, otra aplicación o el usuario.

En general cada programa tiene un bucle de procesamiento de mensajes que es el encargado de procesar los distintos mensajes que le lleguen al mismo a través de Windows y realizar alguna acción en caso de que el mensaje sea de interés. A través del sistema de mensajes se logra tener un sistema multitarea que es el que se maneja por medio de Windows. Además de esta característica, es necesario indicar que una de las mayores ventajas que se tiene al desarrollar software para Windows es que el programador se despreocupa del manejo de los periféricos ya que este manejo lo hace la misma plataforma.

El programa *ProtoGen+* es un conjunto de herramientas que ayudan a desarrollar de una manera gráfica: menús, cuadros de diálogos, ventanas especiales, íconos, etc, para luego dicha información guardada en archivos de recurso (archivos gráficos) sea transformada en archivos de código C, el cual puede ser editado y modificado en el compilador *BORLAND C++*. Además de realizar la interface gráfica el programa *ProtoGen+* tiene algunas tareas básicas como la captura de información desde los cuadros de diálogo de una manera transparente al programador. El programa *ProtoGen+* da la capacidad de a la vez que se va desarrollando la interface gráfica de la aplicación ir paulatinamente transformando esta información a lenguaje C e ir escribiendo las funciones o rutinas propias del programa que se está desarrollando.

Luego de que la interface gráfica ha sido desarrollada y transformada en archivos en C, estos archivos están en capacidad de manejar los distintos mensajes asignados al programa y a cada cuadro de diálogo, para realizar una tarea específica que haya sido desarrollada.

### 3.1 Programa Principal

El módulo denominado programa principal es el responsable de manejar la comunicación entre el ambiente Windows y el software desarrollado, el código de este programa se encuentra bajo el nombre de *ipd.c*.

En general este módulo provee la capacidad de manejar los distintos mensajes que le envía el sistema o el usuario al programa "IPD.EXE", y de esta manera encaminarlos a los restantes módulos. Estos mensajes en el programa son entre otros los siguientes: abrir/cerrar un cuadro de diálogo, abrir/cerrar archivos, etcétera.

El programa "IPD.EXE" tiene un menú principal que abarca las siguientes opciones:

*Archivos, Planta, Entrada, Modelos, TiempoReal, Gráficos, Acerca.*

Las cuales a su vez tienen sub-menus para ingresar o presentar resultados de acuerdo al tipo de cuadro de diálogo que se trate.

En la figura 1 se muestra la pantalla inicial del programa de Identificación Paramétrica Discreta.

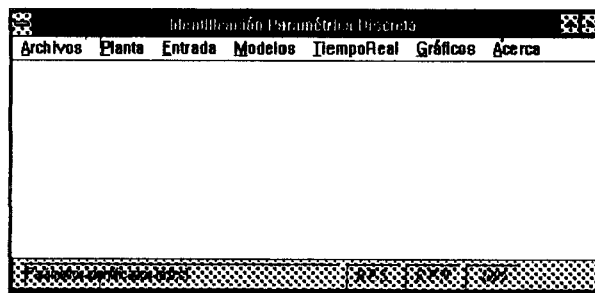


Fig. 1 Menú Principal del programa IPD.EXE

## 4. RESULTADOS

### 4.1 Resultados de simulación

#### SISTEMA DE PRIMER ORDEN (RUIDO BLANCO A LA SALIDA)

Sea el sistema de primer orden que se simula y que se va a identificar está dado por la ecuación de diferencias:

$$y(k) = 0.4y(k-1) + 1.2u(k-1)$$

en donde los parámetros verdaderos a identificar son:

$$\Theta(k) = [0.4 \quad 1.2]^T$$

Este sistema tiene las siguientes características:  
 Entrada: escalón (1) + ruido randómico al 100%.  
 Salida: ruido randómico 100%.

Identificando el sistema mediante el método de mínimos cuadrados recursivo se llega a tener los siguientes parámetros:

$$\Theta(k) = [0.4 \quad 1.2]^T$$

A continuación se presenta los gráficos más relevantes en la identificación paramétrica del sistema.

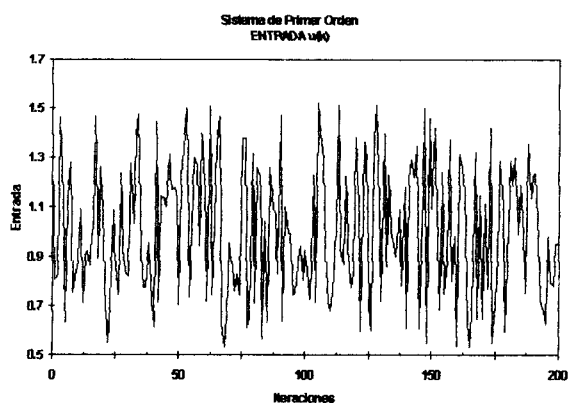


Fig. 2 Entrada al sistema

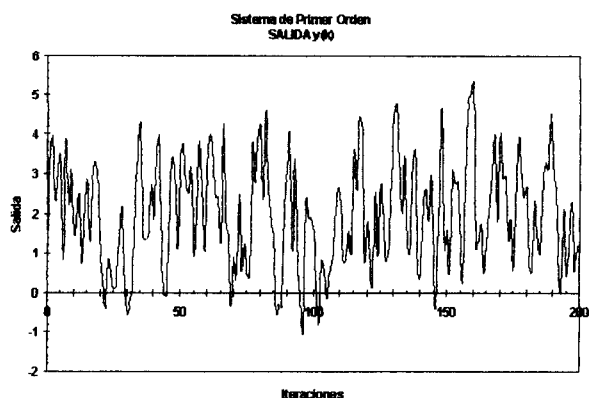


Fig. 3 Salida del sistema

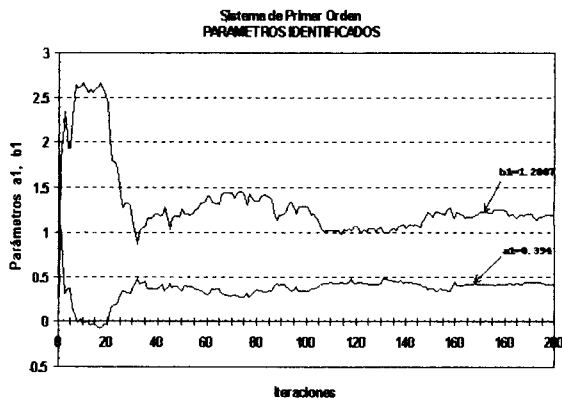


Fig. 4 Parámetros identificados

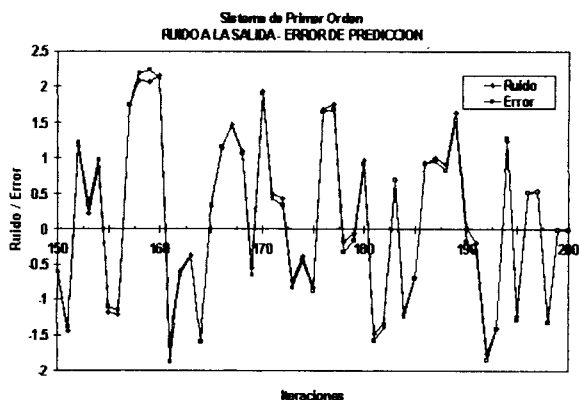


Fig. 5 Ruido a la salida y Error de predicción

### SISTEMA DE SEGUNDO ORDEN (RUIDO CORRELACIONADO A LA SALIDA)

El sistema de segundo orden que se simula y que se va a identificar está dado por la ecuación de diferencias:

$$y(k) = 1.2y(k-1) - 0.35y(k-2) + 0.2u(k-3) + 0.14u(k-2) + v(k) + 0.9v(k-1) + 0.14v(k-2)$$

en donde los parámetros a identificar son:

$$\Theta(k) = [1.2 \quad -0.35 \quad 0.2 \quad 0.14 \quad 0.9 \quad 0.14]^T$$

Este sistema tiene las siguientes características:

Entrada: Ruido randómico

Ruido a la salida: Ruido correlacionado estadístico 50%.

Al tener ruido correlacionado a la salida se utiliza el método de mínimos cuadrados extendido y el de máximo de Likelihood para identificar parámetros. A continuación se presenta los parámetros identificados al aplicar el método de mínimos cuadrados extendidos.

$$\Theta(k) = [1.37 \quad -0.534 \quad 0.23 \quad 0.092 \quad 0.56 \quad 0.16]^T$$

A continuación se presenta los gráficos más importantes.

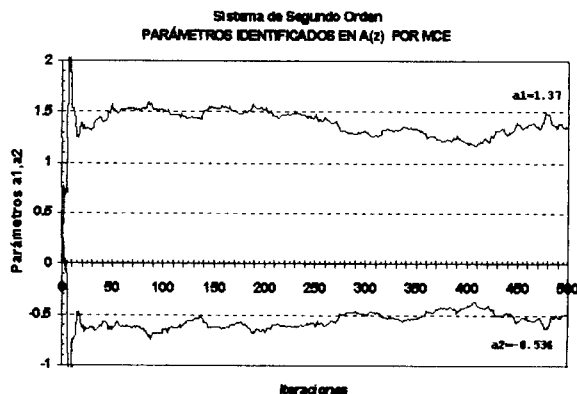


Fig. 6 Parámetros Identificados del polinomio A(z)



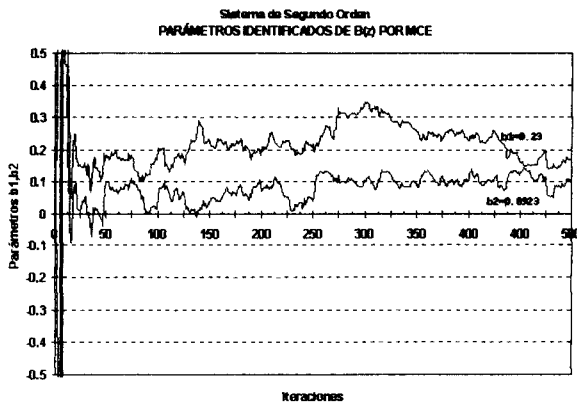


Fig. 7 Parámetros Identificados del polinomio B(z)

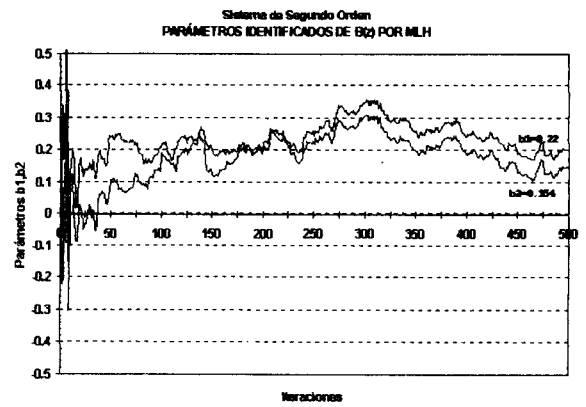


Fig. 10 Parámetros Identificados del polinomio B(z)

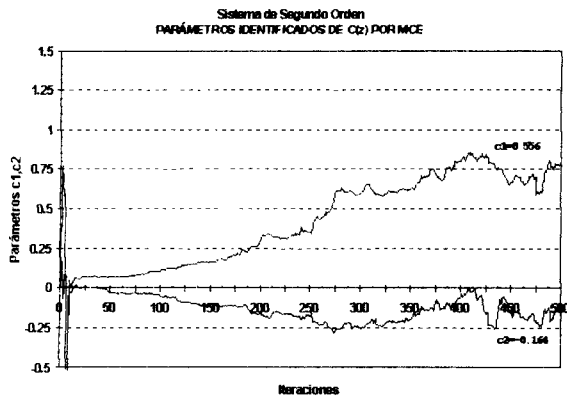


Fig. 8 Parámetros Identificados del polinomio C(z)

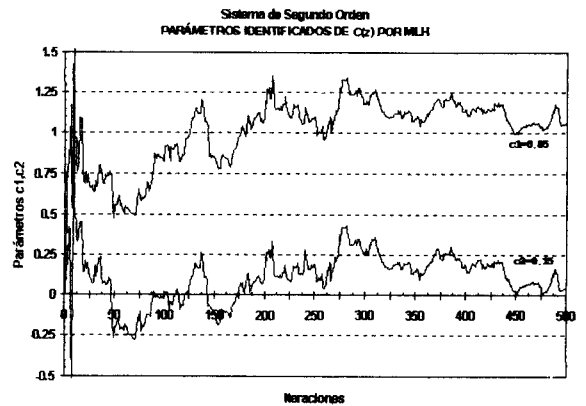


Fig. 11 Parámetros Identificados del polinomio C(z)

A continuación se presentan los resultados de identificar el sistema mediante el método de máximo de Likelihood.

$$\Theta(k) = [1.2 \quad -0.369 \quad 0.22 \quad 0.15 \quad 0.85 \quad 0.15]^T$$

A continuación se presenta los gráficos más importantes.

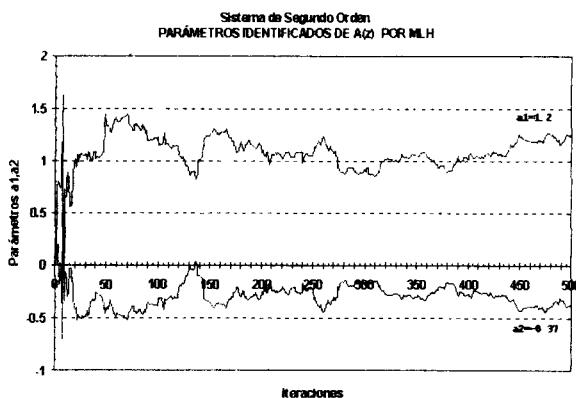


Fig. 9 Parámetros Identificados del polinomio A(z)

#### 4.2 Sistemas en tiempo real

El circuito RC de segundo orden que se utilizó en esta prueba es el mostrado en la figura 12. Al ser ésta una prueba de análisis del funcionamiento del programa de identificación, se discretizó el sistema para un valor de período conocido para de esta manera tener la posibilidad de comparar los valores identificados, con el valor de los parámetros reales del sistema.

Para esta prueba se utiliza el método de mínimos cuadrados recursivo.

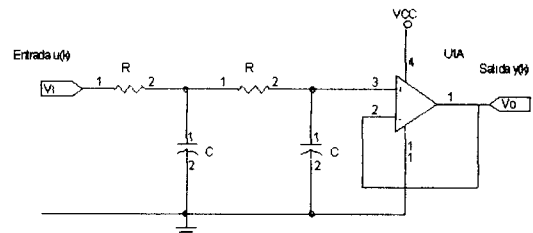


Fig. 12 Circuito RC de Segundo Orden

La función de transferencia del sistema es :

$$G(s) = \frac{V_o(s)}{V_i(s)} = \frac{1}{R^2 C^2 s^2 + 3RCs + 1}$$

Para valores de :

$$C = 10 \mu\text{f}$$

$$R = 120 \text{ k}\Omega$$

$$\text{se tiene que } G(s) = \frac{1}{1.44s^2 + 3.6s + 1}$$

Discretizando la función de transferencia al periodo de muestreo  $T=550$  msec, se tiene:

$$G(z) = \frac{0.0423 + 0.066z}{z^2 - 1.14z + 0.257}$$

por lo que la ecuación de diferencias es:

$$y(k) = 1.14y(k-1) - 0.257y(k-2) + 0.066u(k-1) + 0.0423u(k-2)$$

y los parámetros reales del circuito son:

$$\Theta(k) = [1.14 \quad -0.257 \quad 0.066 \quad 0.0423]^T$$

El sistema presenta las siguientes características para su identificación en tiempo real.

Entrada: Ruido PRBS de amplitud 5.  
Ruido a la salida: Desconocido.

Identificando el sistema mediante el método de mínimos cuadrados recursivos, se tiene los siguientes parámetros:

$$\Theta(k) = [1.155 \quad -0.237 \quad 0.057 \quad 0.0375]^T$$

Número de Iteraciones = 599

A continuación se presenta los gráficos más importantes en el proceso de identificación, es necesario indicar que la información esta referida a las 200 últimas iteraciones.

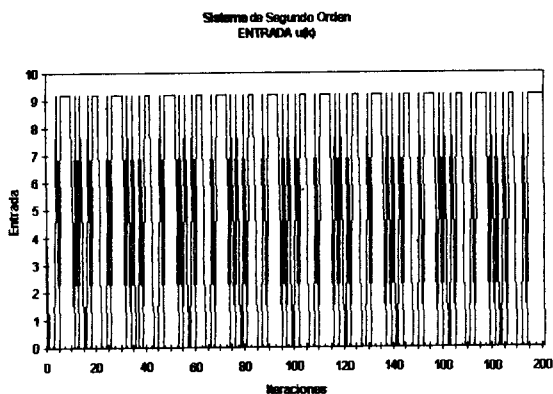


Fig. 13 Entrada al Circuito

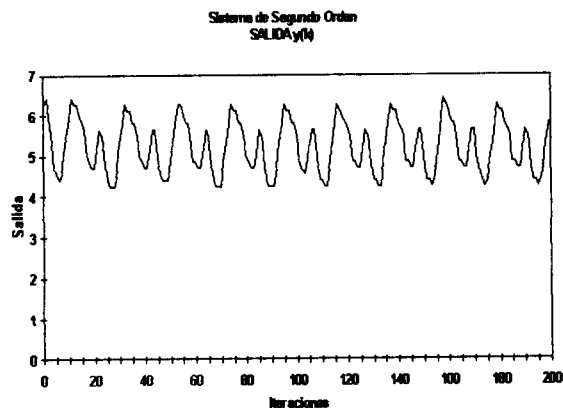


Fig. 14 Salida del Circuito

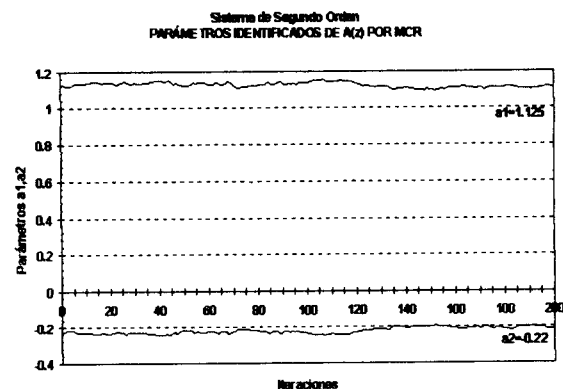


Fig. 15 Parámetros Identificados del polinomio A(z)

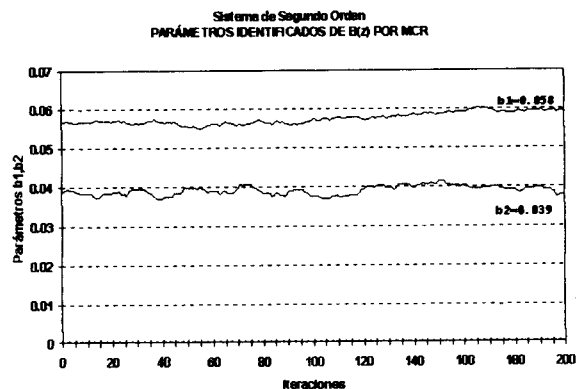


Fig. 16 Parámetros Identificados del polinomio B(z)

### 5. CONCLUSIONES

Al identificar sistemas que tengan ruido blanco a la salida el algoritmo de mínimos cuadrados recursivos (MCR) identifica los parámetros adecuadamente, es decir no existe una desviación en el valor de los parámetros estimados. Esto pone de manifiesto el hecho de que el método de MCR con ruido blanco a la salida, da un algoritmo de identificación no desviado. Sin embargo a medida que el orden del sistema se incrementa es necesario incrementar el número de iteraciones, así como el ruido que se tenga a la salida no debe ser exageradamente alto para tener una adecuada identificación.

Al analizar las curvas de ruido a la salida junto con la curva de error de predicción se puede notar que cuando los parámetros tienden a estabilizarse el error de predicción sigue en forma fiel al ruido a la salida.

Al identificar sistemas con ruido correlacionado a la salida el método de mínimos cuadrados recursivos presenta una desviación en el valor de los parámetros identificados. En este caso se hace necesaria la utilización de algoritmos más robustos de identificación como son: el método de mínimos cuadrados extendidos (MCE) y método de máximo de Likelihood (MLH).

Cuando se tiene sistemas con presencia de ruido correlacionado, es necesario incrementar el número de iteraciones para tener una correcta identificación de los parámetros.

La señal de entrada para identificar a un sistema es muy importante en el proceso de identificación y es necesario indicar que dependiendo del orden del sistema se debe escoger el tipo de entrada adecuada para excitar al sistema. Si el sistema es de primer orden se puede utilizar una señal de entrada escalón sumada un porcentaje de ruido, mientras que si el sistema a identificar es de orden superior, entonces se debe utilizar una señal aleatoria la cual es de excitación persistente o de cualquier orden de excitación, y es la recomendada para identificar sistemas, esto debido a que a través de ésta se puede tener respuesta del sistema en una amplia gama de frecuencias ya que caso contrario cuando la señal de excitación no cumple los requerimientos de excitación persistente se puede llegar a inestabilidad numérica en los algoritmos o que los parámetros identificados sean desviados.

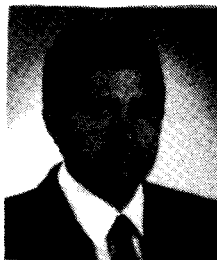
Al identificar el sistema de segundo orden planteado se utilizó el método de mínimos cuadrados recursivos, método que encontró los valores de los parámetros muy cercanos a los verdaderos valores, con lo cual se puede ver que para casos de estimación en tiempo real el MCR funciona adecuadamente.

Es necesario notar que en tiempo real intervienen algunas condiciones que no aparecen en simulación debido a que los sistemas reales presentan variaciones, perturbaciones debida a la dinámica de la planta así como también interacciones del sistema con otros sistemas cercanos como la tarjeta de adquisición de datos. Además que el valor de resistencias así como el valor de los capacitores utilizados tienen tolerancia lo que produce en última instancia que los valores llamados reales de los parámetros sean una aproximación a los valores y que en realidad el programa identifica estos, tomando en cuenta todas estas situaciones.

## 6. BIBLIOGRAFIA

- 1.- Zárate M. B, Identificación Paramétrica Discreta, Tesis de Grado; Escuela Politécnica Nacional. 1996, Quito-Ecuador.
- 2.- Åström K.J., "Introduction to Stochastic Control Theory", Vol 70, Academic Press, New York.
- 3.- Åström K.J., "MAXIMUM LIKELIHOOD AND PREDICTION ERROR METHODS", Automática Vol 16, pp 551-574, 1980.
- 4.- Åström K.J. & Wittenmark B., "ADAPTIVE CONTROL", Lund Institute of Technology; Addison-Wesley Publishing Company, 1989.

## 7. REFERENCIAS



**ZARATE BYRON.** Nació en Quito-Ecuador el 4 de Abril de 1971. Obtuvo el título de Bachiller en Humanidades Modernas en el Colegio La Salle de Quito en 1989. Sus estudios superiores los realizó en la Escuela Politécnica Nacional, obteniendo el título de Ingeniero en Electrónica y Control en abril de 1996.

Actualmente se desempeña como Ingeniero de Proyectos en PROCELEC CIA LTDA.



**BURBANO PATRICIO.** Nació en Quito el 9 de Octubre de 1951. Curso sus estudios superiores en la Escuela Politécnica Nacional obteniendo el título de ingeniero en Electrónica y Telecomunicaciones en 1974. Realizó su post-grado en Control y Sistemas en el UMIST en la Universidad de Manchester en 1983.

Actualmente es Profesor Principal de la Facultad de Ingeniería Eléctrica, EPN, y su campo de investigación esta orientado al control digital, multivariable, adaptivo.