

IMPLEMENTACION DE UN PREDICTOR DIGITAL

Eugenio Tacconi, Ricardo Mantz, Juan Carasi Ings.
Laboratorio de Electrónica Industrial, Control e Instrumentación (LEICI), Universidad Nacional de La Plata.

RESUMEN

Se describe la implementación de un filtro predictor digital en tiempo real realizado con un sistema de microcomputo basado en un microprocesador MC6802.

Previamente se realiza una introducción sobre predicción lineal y los diversos métodos de estimación de los coeficientes de predicción.

INTRODUCCION

Un filtro predictor es aquel que excitado con la señal que se desea predecir, provee a su salida la evolución futura de dicha señal.

La exactitud de la predicción depende de la información accesible. Así, si la única información disponible es la observación de la señal a predecir es posible realizar una primera predicción en base al valor de la señal y al de su velocidad de cambio.

Una apreciable mejora puede ser obtenida si se dispone también de algún conocimiento acerca del proceso que origina el fenómeno a predecir. Si se considera que la señal a predecir es la salida de un sistema $G(S)$ (fig. (1)) cuando se lo excita con una señal $z(t)$, un conocimiento de los estados de dicho sistema permite lograr una substancial mejora en la predicción.

En este caso el error de predicción se debe exclusivamente a la incertidumbre existente debido al desconocimiento de la excitación $w(t)$ entre el instante presente y aquel en el cual la señal se quiere predecir.

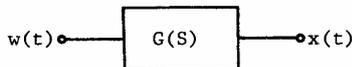


Figura 1

Es de acotar que de existir un tiempo de retardo en el sistema, la predicción puede realizarse sin error, al menos desde el punto de vista teórico.

Originalmente los filtros predictores se emplearon para analizar la evolución futura de series económicas y para predecir la trayectoria de blancos en movimiento. Actualmente su aplicación se extiende prácticamente a todas las ciencias, existiendo espectativas altamente positivas para su divulgación debido a las posibilidades de implementación en sistemas de microcomputo.

1.- BREVE DESCRIPCION DEL FUNDAMENTO DE LA PREDICCIÓN LINEAL

Debido a que todo proceso físico posee un espectro de frecuencias de banda limitada, es posible realizar una predicción de su e-

volución considerando la vinculación que existe entre el valor que tomará la señal en el futuro y los valores que ha tomado anteriormente. Naturalmente el error de predicción dependerá en forma directa del tiempo de predicción deseado.

El filtro de predicción lineal que se estudia en el presente trabajo estima la salida futura del proceso a partir de una combinación lineal de las salidas presente y pasadas.

Con el objeto de procesar digitalmente la señal analógica $x(t)$ que se desea predecir, esta debe ser muestreada a una frecuencia superior a la de Nyquist con la finalidad de no perder su contenido armónico. Se obtiene así una señal muestreada $x(nT)$ que toma valor solamente en los instantes de muestreo. Denominaremos x_n al valor que toma la señal en el instante nT .

Conocida la evolución del proceso hasta el instante nT , se desea estimar su salida futura x_{n+1} mediante la expresión (1).

$$\hat{x}_{n+1} = a_1 x_n + a_2 x_{n-1} + \dots + a_p x_{n-p+1} \quad (1)$$

Donde a_i , para $1 \leq i \leq p$, son los coeficientes del filtro de predicción.

La expresión (1) puede indicarse en forma vectorial como:

$$\hat{\vec{x}}_{n+1} = \vec{a}' \vec{X}_{n+1} \quad (2)$$

con:

$$\vec{a}' = \{a_1, \dots, a_p\}$$

$$\vec{X}_{n+1} = \{x_n, \dots, x_{n-p+1}\}$$

La notación $(.)'$ denota al vector traspuesto.

El desarrollo de un filtro de predicción lineal puede implementarse en base al diagrama en bloques de la Fig. 2.

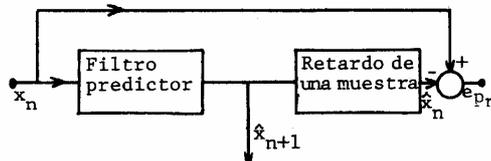


Figura 2

Los coeficientes óptimos del filtro de predicción, según el criterio de cuadrados mínimos, serán aquellos que minimicen el error medio cuadrático definido por:

$$P = E(e_{Pn}^2) = E \{ (x_n - \hat{x}_n)^2 \} \quad (3)$$

Donde la operación $E(\cdot)$ denota esperanza matemática.

Teniendo en cuenta que cualquier sistema puede ser modelizado con un modelo autorregresivo puro de infinitos coeficientes, y aproximado por uno de un número finito de ellos, se supondrá que la señal x_n a predecir proviene de un proceso autorregresivo:

$$x_n = z_n + b_1 x_{n-1} + b_2 x_{n-2} + \dots + b_\ell x_{n-\ell} \quad (4)$$

Siendo z_n un ruido blanco de distribución gaussiana.

Puede demostrarse que si el número de coeficientes p del predictor es igual o mayor que ℓ la esperanza del error medio cuadrático en función de los valores de los coeficientes a_i del filtro predictor, es un hiperparaboloide que presenta un mínimo absoluto en el punto dado por las coordenadas

$$\begin{aligned} a_i &= b_i & \text{para } 1 \leq i \leq \ell \\ a_i &= 0 & \text{para } \ell < i \leq p \end{aligned}$$

Es de interés entonces que el filtro predictor tenga como coeficientes los b_i que son óptimos en el sentido de minimizar la esperanza del error medio cuadrático.

El cálculo de estos coeficientes puede ser realizado en forma directa o iterativa.

1.1 Resolución directa de los coeficientes óptimos del filtro.

Reemplazando (2) en (3) se obtiene:

$$P = E \{ (x_n - \vec{a}' \vec{X}_n)^2 \} \quad (5)$$

Desarrollando el cuadrado y teniendo en cuenta que el vector \vec{a} es constante, se obtiene la expresión (6)

$$P = E\{x_n^2\} + \vec{a}' E\{\vec{X}_n \vec{X}_n'\} \vec{a} - 2\vec{a}' E\{x_n \vec{X}_n\} \quad (6)$$

Se define a la función de autocorrelación del proceso x como:

$$R(m) = E\{x_n x_{n-m}\} \quad (7)$$

Reemplazando en (6) resulta:

$$P = R(0) + \vec{a}' R \vec{a} - 2\vec{a}' \vec{C} \quad (8)$$

Donde:

$$\vec{C}' = \{R(1), \dots, R(p)\}$$

$$R = \begin{bmatrix} R(0) & \dots & R(p-1) \\ \vdots & \ddots & \vdots \\ R(p-1) & \dots & R(0) \end{bmatrix}$$

Derivando la expresión (8) respecto a \vec{a} e igualando a cero se logran los coeficientes óptimos \vec{a}^{OP} deseados.

$$\frac{\partial (e_{Pn}^2)}{\partial \vec{a}} = 0 = 2R\vec{a}^{OP} - 2\vec{C} \quad (9)$$

$$\vec{a}^{OP} = R^{-1}\vec{C} \quad (10)$$

Las Ecs. (10) se conocen con el nombre de ecuaciones normales o ecuaciones de Yule-Walker. Su resolución requiere la estimación previa de la función de autocorrelación $R(m)$ y la inversión de una matriz cuadrada de orden p .

1.2 Resolución iterativa de los coeficientes óptimos del filtro.

Los métodos iterativos de obtención del mínimo de una función, involucran generalmente menor número de operaciones matemáticas y proveen además una estimación de los coeficientes a_i adaptable a las fluctuaciones de los coeficientes verdaderos b_i .

De los diversos métodos iterativos, solamente será analizado el algoritmo de Widrow, por ser el de menor complejidad, facilitando su implementación con un microprocesador.

1.2.1 Algoritmo del gradiente

El algoritmo del gradiente en el cual se basan los trabajos de Widrow, es un método iterativo que consiste en descender sobre el hiperparaboloide del error medio cuadrático siguiendo la dirección del gradiente dada por la Ec. (11)

$$\nabla P = \frac{\partial P}{\partial a_1} \vec{a}_1 + \dots + \frac{\partial P}{\partial a_p} \vec{a}_p \quad (11)$$

Donde los \vec{a}_i indican a los versores en las direcciones de a_i .

Para visualizar el procedimiento, para el caso de un sistema AR de 2 coeficientes, se grafica la superficie determinada por la ecuación del error medio cuadrático P en función de los parámetros a_1 y a_2 (fig. 3) y las curvas de error medio cuadrático constante (fig. 4).

Se inicializa el predictor utilizando todo conocimiento previo sobre los valores probables de los coeficientes a calcular, luego la estimación inicial es desplazada descendiendo sobre la superficie del error medio cuadrático en la dirección del gradiente (perpendicular a las curvas de nivel), hasta alcanzar el mínimo (línea de trazos en la Fig. 4).

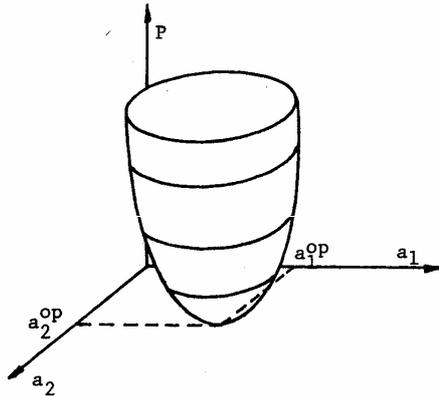


Fig. 3. - Superficie del error medio cuadrático en función de los coeficientes a_1 y a_2 .

El algoritmo más simple para llevar a cabo tal operación, consiste en descender a pasos en forma iterativa mediante la aplicación de las ecuaciones:

$$a_i^{n+1} = a_i^n - \mu \frac{\partial P}{\partial a_i} \quad \text{para } 1 \leq i \leq p \quad (12)$$

Donde el índice superior indica el orden de iteración.

El coeficiente μ es una constante que determina la dimensión del paso; un valor de μ demasiado pequeño requerirá un número elevado de iteraciones para alcanzar el mínimo y un valor excesivo puede ocasionar inestabilidades en el algoritmo.

1.2.2 Algoritmo de Widrow

El algoritmo del gradiente requiere la determinación del gradiente del error medio cuadrático en cada iteración.

Widrow analizó las condiciones de estabilidad, constante de tiempo de convergencia, y error de estimación del algoritmo anterior reemplazando el valor del gradiente por una estimación del mismo empleando una sola muestra. Es decir, aproximando la esperanza matemática del error cuadrático por el error cuadrático en dicha iteración (Expresión (13)).

$$a_i^{n+1} = a_i^n - \mu \frac{\partial (e_p^2)}{\partial a_i} \quad \text{para } 1 \leq i \leq p \quad (13)$$

Siendo el error de predicción de la forma:

$$e_{p_n} = \sum_{i=1}^p (b_i - a_i) x_{n-i} + z_n \quad (14)$$

Elevando al cuadrado esta expresión, calculando las derivadas parciales y reemplazando en la Ec. (13), se obtiene:

$$a_i^{n+1} = a_i^n + 2\mu \cdot e_{p_n} \cdot x_{n-i} \quad (15)$$

con:

$$e_{p_n} = x_n - \sum_{i=1}^p a_i^n x_{n-i} \quad (16)$$

Las Ecs. (15) y (16) permiten estimar los coeficientes del predictor en forma iterativa a partir de las salidas del sistema físico.

A modo de ejemplo puede verse en la fotografía 1 el ajuste automático de dos de los coeficientes de un filtro predictor implementado, se ha empleado un osciloscopio con memoria como registrador gráfico. Se observa cómo cada coeficiente partiendo de su valor inicial converge al valor que minimiza la esperanza del error medio cuadrático.

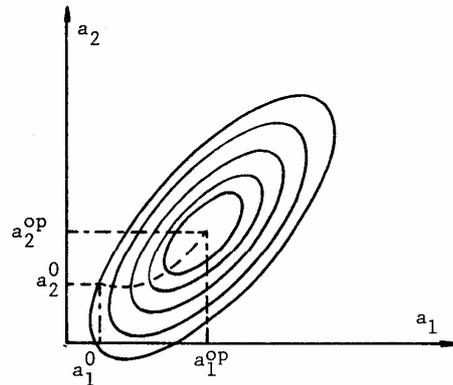


Fig. 4. - Curvas de error medio cuadrático constante.

Resumiendo, las ecuaciones de predicción son las siguientes:

$$\hat{x}_{n+1} = \sum_{i=1}^p a_i^n x_{n+1-i}$$

$$a_i^n = a_i^{n-1} + 2\mu e_{p_n} x_{n-i} \quad (17)$$

$$e_{p_n} = x_n - \hat{x}_n$$

1.3 Predicción de varias muestras en adelante.

El filtro de predicción es anticipativo en r muestras si al recibir la muestra x_n provee a su salida una estimación \hat{x}_{n+r} de la muestra x_{n+r} .

Para predecir r muestras futuras deben incorporarse, en el diagrama en bloques de la Fig. 2, r unidades de retardo y las Ecs. del filtro de predicción se transforman en:

$$\hat{x}_{n+r} = \sum_{i=1}^p a_i^n x_{n-i+1} \quad (18)$$

$$a_i^n = a_i^{n-1} + 2\mu e_{p_n} x_{n-r+1-i}$$

para $1 \leq i \leq p$

En general, el error de predicción depende inversamente del número de coeficientes del filtro y aumenta en proporción directa al número de muestras r a predecir.

Esto puede verificarse considerando un proceso aleatorio gaussiano y calculando la probabilidad condicional de las muestras futuras.

2.- IMPLEMENTACION DEL PREDICTOR DIGITAL

2.1 Descripción del sistema

La implementación del predictor digital se realizó empleando un sistema de microcomputo basado en un microprocesador MC6802.

Para el muestreo de la señal a predecir se empleó un convertidor análogo-digital de 8 bits BURR-BROWN, MP21,⁵ compatible con la familia de microprocesadores 6800. Debido a que esta unidad es tratada como una posición de memoria la conversión es realizada con simples instrucciones de lectura.

Para el procesamiento de la señal, consistente en la resolución de las ecuaciones de predicción, se utilizó una Unidad Aritmética de Procesamiento (APU) de Advanced Micro Devices, Am 9511A,⁶ que puede realizar, entre otras operaciones, las aritméticas básicas (suma, resta, multiplicación y división) tanto en punto fijo (en 16 y 32 bits) como en punto flotante (32 bits).

Una de las características más importantes del circuito integrado es que posee en su interior capacidad de almacenamiento de datos, formando una pila del tipo LIFO de 8 palabras de 16 bits o 4 de 32 bits.

Los datos almacenados en esta pila pueden ser manipulados empleando comandos especiales. Comandos que permiten, por ejemplo, intercambiar la ubicación de datos en la pila, duplicar datos, realizar conversiones de punto fijo a flotante y viceversa, etc.

Las transferencias de datos se hacen de a 8 bits a través de un bus bidireccional. Así la carga y descarga de datos como el envío de comandos se realizan mediante instrucciones de lectura y escritura del microprocesador.

Una vez procesada la señal en forma digital es convertida a analógica mediante un convertidor digital-analógico BURR BROWN MP11,⁷ electrónica y funcionalmente compatible con el microprocesador MC6802. Este periférico es también tratado como una posición de memoria.

2.2 Descripción del programa

El filtro predictor implementado consta de tres coeficientes adaptables (a_1 , a_2 y a_3) y permite predecir hasta cinco muestras adelante, siendo la cantidad de muestras a predecir seleccionable mediante la modificación del contenido de una posición de memoria. En la fig. (5) se ha graficado el diagrama de flujo del programa del predictor.

Las operaciones fundamentales se realizan en punto flotante, para tal fin se desarrolló una subrutina que permite convertir valores de punto fijo a punto flotante. Por otra parte, debido a la cantidad de datos que se manipulan en la resolución de las ecuaciones y a la limitada capacidad de la pila de la Unidad de Procesamiento Aritmético, es necesario almacenar datos en 32 bits temporariamente en memoria. Para ello se utilizan subrutinas de carga y descarga de la APU.

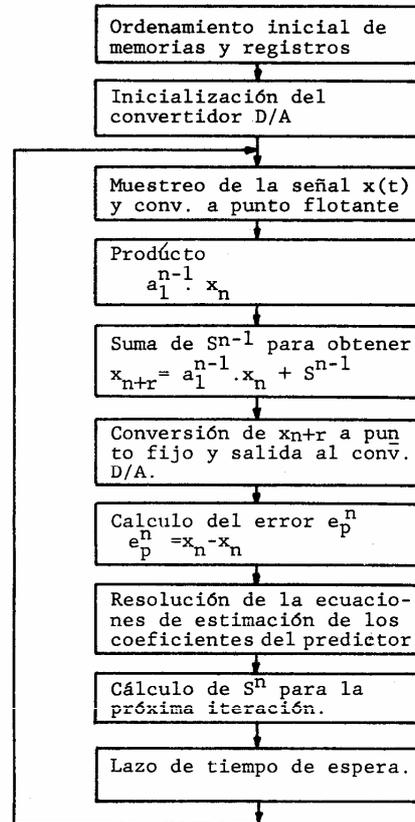


Fig.5.-Diagrama de flujo del programa

Existe un tiempo, desde el instante en que se dispone de la información necesaria para realizar la predicción hasta que esta se encuentra disponible a la salida del convertidor D/A, cuyo efecto es reducir el tiempo efectivo de predicción. El programa se ha diagramado con la finalidad de minimizar este tiempo de retardo.

Si se reescribe la ecuación (18) para el caso de tres coeficientes de la siguiente manera:

$$\hat{x}_{n+r} = x_n a_1^{n-1} + S^{n-1} \quad (19)$$

donde

$$S^{n-1} = a_2^{n-1} x_{n-1} + a_3^{n-1} x_{n-2}$$

se observa que el término S^{n-1} depende de

las muestras pasadas de la señal a predecir, (x_{n-1} y x_{n-2}) las cuales previamente almacenadas en memoria pueden emplearse para calcular S^{n-1} antes de adquirir la muestra actual x_n .

La secuencia de operaciones realizadas para obtener \hat{x}_{n+r} son las siguientes:

- se muestrea la señal x_n
- se multiplica x_n por el coeficiente a_1^{n-1}
- y - se adiciona al producto el término S^{n-1} previamente calculado.

Posteriormente se calcula el error de predicción e_n^p , se estiman los coeficientes a_1^n , a_2 y a_3 y por último se halla el término S^n , que almacenado en la pila de la APU corresponderá a S^{n-1} en la próxima iteración.

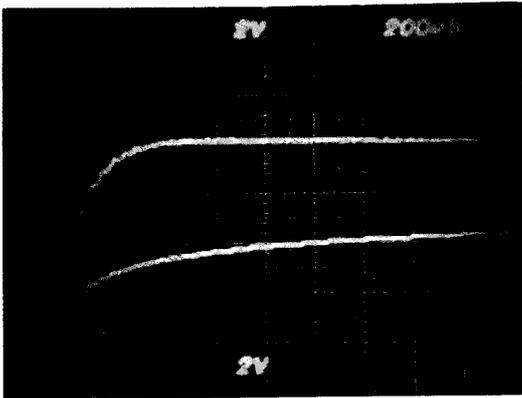
El período de muestreo mínimo puede ser calculado teniendo en cuenta los ciclos de reloj de las instrucciones involucradas en el programa. La unidad de procesamiento aritmético puede operar en forma asincrónica y hasta con una frecuencia de reloj de 3 MHz.

A partir de los ciclos empleados por cada operación de esta unidad y del tiempo de conversión del convertidor A/D (40 μ seg.) el período de muestreo mínimo posible de obtener es del orden de 3mseg y el tiempo de retardo, considerando el tiempo de establecimiento del convertidor D/A (25 μ s), es de aproximadamente 0,5 mseg. El período de muestreo puede ser incrementado modificando el tiempo de espera incorporado en el lazo.

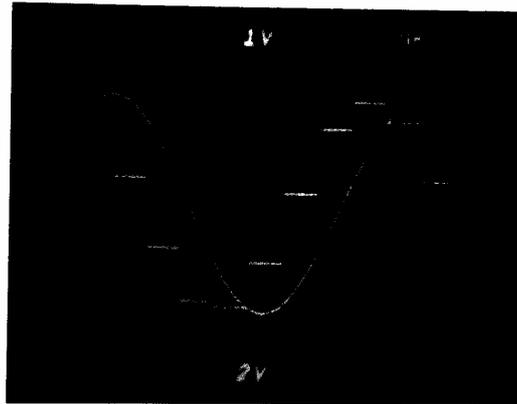
En el apéndice 1 se adjunta el listado del programa empleado.

Resultados experimentales

A continuación se muestran una serie de fotografías que permiten visualizar los resultados alcanzados con el filtro predictor implementado. Las fotografías 2 corresponden al caso en que se excita el filtro predictor con una señal determinística (señal sinusoidal), puede observarse en ella tanto la señal a predecir como su predicción en dos muestras en adelante.



Fotografía 1.-Ajuste automático de dos de los coeficientes del filtro predictor.



Fotografía 2.-Señal determinística y su predicción en dos muestras en adelante.

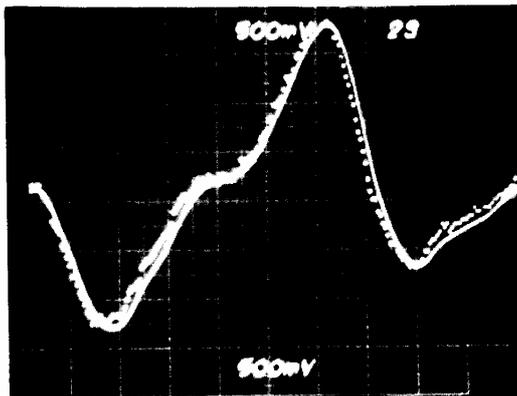
En las fotografías 3 y 4 puede verse el comportamiento del filtro predictor cuando es excitado con una señal aleatoria obtenida a partir de un generador de ruido de baja frecuencia. En la fotografía 3 la predicción es de dos muestras en adelante mientras que en la 4 es de tres.

Por último, en la fotografía 5 el filtro es excitado con una señal sinusoidal amortiguada común de encontrar en sistemas realimentados de control, campo en el cual la incorporación de filtros predictores puede traer apreciables mejoras.

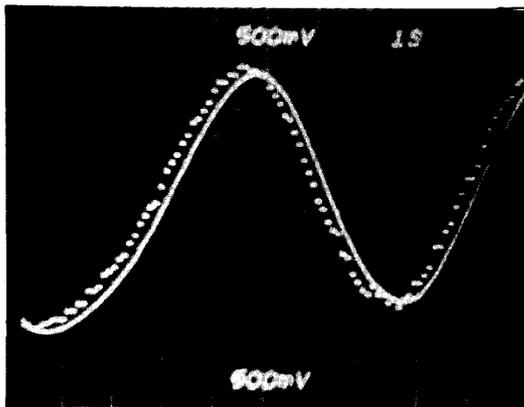
CONCLUSIONES

La simplicidad de las ecuaciones de predicción basadas en el algoritmo de Widrow permite una sencilla implementación de filtros de predicción empleando sistemas de microcomputo.

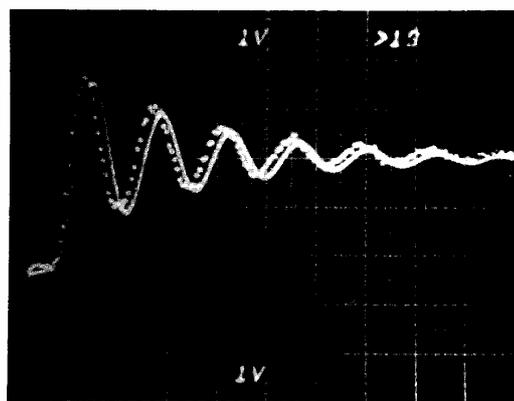
Los resultados obtenidos concuerdan con los esperados a partir del estudio teórico. De la observación de las fotografías se desprende que la salida del filtro permite predecir con buena aproximación la evolución futura de un proceso físico.



Fotografía 3.-Excitación y salida del filtro predictor para el caso correspondiente a una señal aleatoria.



Fotografía 4.-Señal ruido de baja frec. y su predicción en tres muestras.



Fotografía 5.-Señal oscilatoria amortiguada y su predicción en tres muestras.

APENDICE 1.

Programa del filtro predictor

M6801A IS THE PROPERTY OF MOTOROLA SPD, INC.
COPYRIGHT 1978 BY MOTOROLA INC

MOTOROLA M6801 CROSS ASSEMBLER, RELEASE 1.0

```

00001          NAM    PREDIC
00002          *-----
00003          *
00004          *PRGRAMA DEL PREDICTOR DIGITAL.
00005          *-----
00006          *
00007          *EL NUMERO DE MUESTRAS A PREDECIR (R) DEBE SER CARGADO EN LA
00008          *POSICION DE MEMORIA $0009.
00009          *LAS MUESTRAS ACTUAL Y PASADAS DEL PROCESO A PREDECIR (X) SE
00010          *ALMACENAN DESDE LA POSICION DE MEMORIA $0000 A LA $0007.
00011          *LAS PREDICIONES ACTUAL Y ANTERIORES DEL PROCESO (XP) SE
00012          *ALMACENAN DESDE LA POSICION DE MEMORIA $0020 A LA $0033.
00013          *EL VALOR DE MU SE CARGA DESDE LA POSICION $0010 A LA $0013.
00014          *LOS COEFICIENTES A1,A2 Y A3 SE ALMACENAN TEMPORARIAMENTE
00015          *DESDE LA POSICION $0014 A LA $001F.
00016          *EL DATO PARA VARIAR EL TIEMPO DE ESPERA DEBE SER CARGADO EN
00017          *LAS POSICIONES $000C Y $000D.
00018          *-----
00019 0040          ORG    $0040
00020 0040 7F 0008  CLR    RH          REORDENAMIENTO INIC. DE MEMORIAS Y REGISTROS
00021 0043 7F 000A  CLR    PH
00022 0046 B6 30    LDA    A    $*30
00023 0048 5F      CLR    B
00024 0049 5C      SALT02 INC    B
00025 004A D1 09   CMP    B    RH+1
00026 004C 27 05   BEQ    SALTO1
00027 004E 80 04   SUB    A    $*04
00028 0050 20 F7   BRA    SALT02
00029 0053 97 0B   SALT01 STA    A    PH+1
00030 0055 CE FFFF LDX    $*FFFF  INICIALIZACION DE CONVERTIDORES D/A
00031 0058 FF 3FF4 STX    DAC
00032 005B FF 3FF6 STX    DACI
00033 005F B6 2FF0 INICIO LDA    A    ADC  MUESTREO DEL PROCESO X(T)
00034 0062 01      NOP
00035 0063 B6 2FF0 LDA    A    ADC
00036 0066 97 00   STA    A    XN
00037 0068 BD 0166 JSR    SUBCON  CARGA DE X(N) EN LA APU Y CONV. A FLOTANTE
00038 006B CE 0014 LDX    $*0014 CARGA DE A1(N-1) EN LA APU
00039 006E BD 0180 JSR    SUBCAR
00040 0071 B6 12    LDA    A    $*12  PRODUCTO XN*A1(N-1)
00041 0073 B7 4008 STA    A    APUCOM
00042 0076 B6 10    LDA    A    $*10  SUMA X(N)*A1(N-1)+S(N-1)=XP(N+R)
00043 007B B7 4008 STA    A    APUCOM
00044 007B B6 17    LDA    A    $*17  DUPLICACION DE XP(N+R)
00045 007D E7 4008 STA    A    APUCOM
00046 0080 B6 1F    LDA    A    $*1F  CONVERSION A PUNTO FIJO
00047 0082 B7 4008 STA    A    APUCOM

```

00048	0085	B6	4000	LDA	A	APUDAT	SALIDA DE XP(N+R) AL D/A
00049	0088	B6	4000	LDA	A	APUDAT	
00050	008B	8B	80	ADD	A	##80	
00051	008D	43		COM	A		
00052	008E	B7	3FF4	STA	A	DAC	
00053	0091	DE	0A	LDX		PH	CARGA DE XP(N) EN LA APU
00054	0093	BD	0180	JSR		SUBCAR	
00055	0096	B6	19	LDA	A	##19	ROTACION DE LOS DOS PRIMEROS DATOS DE LA APU
00056	0098	B7	4008	STA	A	AFUCOM	
00057	009D	A6	04	SALTO3	LDA	A	04,X
00058	009F	A7	00	STA	A	00,X	REACOMODACION DE LOS XP(N+I) PARA LA PROXIMA ITERACION
00059	00A1	08		INX			
00060	00A2	BC	0034	CPX		##0034	
00061	00A5	26	F6	BNE		SALTO3	
00062	00A7	CE	0030	LDX		##0030	ALMACENAMIENTO DE XP(N+R) EN MEMORIA
00063	00AA	BD	0195	JSR		SUBAL	
00064	00AD	96	00	LDA	A	XN	CARGA DE XN EN LA APU Y CONV. A FLOTANTE
00065	00AF	BD	0166	JSR		SUBCON	
00066	00B2	B6	11	LDA	A	##11	RESTA XP(N)-X(N)=-EP(N)
00067	00B4	B7	4008	STA	A	AFUCOM	
00068	00B7	CE	0010	LDX		##0010	CARGA DE -MU EN LA APU
00069	00BA	B6	12	LDA	A	##12	PRODUCTO -MU*(-EP(N))
00070	00BC	B7	4008	STA	A	AFUCOM	
00071	00BF	B6	17	LDA	A	##17	DUPLICACION DE MU*EP(N)
00072	00C1	B7	4008	STA	A	AFUCOM	
00073	00C4	DE	08	LDX		RH	CARGA DE X(N-R) EN LA APU Y CONV. A FLOTANTE
00074	00C6	A6	00	LDA	A	00,X	
00075	00C8	BD	0166	JSR		SUBCON	
00076	00CB	B6	12	LDA	A	##12	PRODUCTO MU*EP(N)*X(N-R)
00077	00CD	B7	4008	STA	A	AFUCOM	
00078	00D0	CE	0014	LDX		##0014	CARGA DE A1(N-1) EN LA APU
00079	00D3	BD	0180	JSR		SUBCAR	
00080	00D6	B6	10	LDA	A	##10	SUMA A1(N-1)+MU*EP(N)*X(N-R)=A1(N)
00081	00D8	B7	4008	STA	A	AFUCOM	
00082	00DB	BD	0195	JSR		SUBAL	ALMACENAMIENTO DE A1(N) EN MEMORIA
00083	00DE	B6	17	LDA	A	##17	DUPLICACION DE MU*EP(N)
00084	00E0	B7	4008	STA	A	AFUCOM	
00085	00E3	DE	08	LDX		RH	CARGA DE X(N-R-1) EN LA APU Y CONV. A FLOTANTE
00086	00E5	A6	01	LDA	A	01,X	
00087	00E7	BD	0166	JSR		SUBCON	
00088	00EA	B6	12	LDA	A	##12	PRODUCTO MU*EP(N)*X(N-R-1)
00089	00EC	B7	4008	STA	A	AFUCOM	
00090	00EF	CE	0018	LDX		##0018	CARGA DE A2(N-1) EN LA APU
00091	00F2	BD	0180	JSR		SUBCAR	
00092	00F5	B6	10	LDA	A	##10	SUMA A2(N-1)+MU*EP(N)*X(N-R-1)=A2(N)
00093	00F7	B7	4008	STA	A	AFUCOM	
00094	00FA	B6	17	LDA	A	##17	DUPLICACION DE A2(N)
00095	00FC	B7	4008	STA	A	AFUCOM	
00096	00FF	BD	0195	JSR		SUBAL	ALMACENAMIENTO DE A2(N) EN MEMORIA
00097	0102	B6	19	LDA	A	##19	ROTACION DE LOS DOS PRIMEROS DATOS DE LA APU
00098	0104	B7	4008	STA	A	AFUCOM	
00099	0107	DE	08	LDX		RH	CARGA DE X(N-R-2) EN LA APU Y CONV. A FLOTANTE
00100	0109	A6	02	LDA	A	02,X	
00101	010B	BD	0166	JSR		SUBCON	
00102	010E	B6	12	LDA	A	##12	PRODUCTO MU*EP(N)*X(N-R-2)
00103	0110	B7	4008	STA	A	AFUCOM	
00104	0113	CE	001C	LDX		##001C	CARGA DE A3(N-1) EN LA APU
00105	0116	BD	0180	JSR		SUBCAR	
00106	0119	B6	10	LDA	A	##10	SUMA A3(N-1)+MU*EP(N)*X(N-R-2)=A3(N)
00107	011B	B7	4008	STA	A	AFUCOM	
00108	011E	B6	17	LDA	A	##17	DUPLICACION DE A3(N)
00109	0120	B7	4008	STA	A	AFUCOM	
00110	0123	BD	0195	JSR		SUBAL	ALMACENAMIENTO DE A3(N) EN MEMORIA
00111	0126	DE	08	LDX		RH	REORDENAMIENTO DE LOS X(N-I) PARA LA PROXIMA ITERACION
00112	012D	A6	01	SALTO4	LDA	A	01,X
00113	012F	A7	02	STA	A	02,X	
00114	0131	09		DEX			
00115	0132	BC	0000	CPX		##0000	
00116	0135	26	F6	BNE		SALTO4	
00117	0137	96	01	LDA	A	XN+1	
00118	0139	97	02	STA	A	XN+2	
00119	013B	BD	0166	JSR		SUBCON	CARGA DE X(N-1) EN LA APU Y CONV. A FLOTANTE
00120	013E	B6	12	LDA	A	##12	PRODUCTO A3(N)*X(N-1)
00121	0140	B7	4008	STA	A	AFUCOM	
00122	0143	B6	19	LDA	A	##19	ROTACION DE LOS DOS PRIMEROS DATOS DE LA PILA
00123	0145	B7	4008	STA	A	AFUCOM	
00124	0148	96	00	LDA	A	XN	REORDENAMIENTO DE DATOS
00125	014A	97	01	STA	A	XN+1	
00126	014C	BD	0166	JSR		SUBCON	CARGA DE X(N) EN LA APU Y CONV. A FLOTANTE

```

00127 014F 86 12          LDA A  ##12      PRODUCTO A2(N)*X(N)
00128 0151 B7 4008        STA A  APUCOM
00129 0154 86 10          LDA A  ##10      SUMA A2(N)*X(N)+A3(N)*X(N-1)=S(N)
00130 0156 B7 4008        STA A  APUCOM
00131 0159 DE 0C          LDX  ESPERA      LAZO DE TIEMPO DE ESPERA
00132 0160 09             SALTOS DEX
00133 0161 26 FD          BNE  SALTOS
00134 0163 7E 005F        JMP  INICIO
00135
00136 *-----
00137 *SUBROUTINA DE CARGA EN LA APU Y CONVERSION A PUNTO FLOTANTE
00138 0166 5F             SUBCON CLR B      CARGA DEL DATO EN LA APU EN PUNTO FIJO
00139 0167 11             CBA
00140 0168 2B 08          BMI  SUB1
00141 016A B7 4000        STA A  APUDAT
00142 016D F7 4000        STA B  APUDAT
00143 0170 20 08          BRA  SUB2
00144 0172 B7 4000        SUB1 STA A  APUDAT
00145 0175 86 FF          LDA A  ##FF
00146 0177 B7 4000        STA A  APUDAT
00147 017A 86 1D          SUB2 LDA A  ##1D      CONVERSION DE PUNTO FIJO A FLOTANTE
00148 017C B7 4008        STA A  APUCOM
00149 017F 39             RTS
00150
00151 *-----
00152 *SUBROUTINA DE CARGA DE DATOS EN LA APU
00153 0180 A6 03           SUBCAR LDA A  03,X      CARGA DEL DATO EN LA APU EN PUNTO FLOTANTE
00154 0182 B7 4000        STA A  APUDAT
00155 0185 A6 02           LDA A  02,X
00156 0187 B7 4000        STA A  APUDAT
00157 018A A6 01           LDA A  01,X
00158 018C B7 4000        STA A  APUDAT
00159 018F A6 00           LDA A  00,X
00160 0191 B7 4000        STA A  APUDAT
00161 0194 39             RTS
00162
00163 *-----
00164 *SUBROUTINA DE ALMACENAMIENTO DE DATOS EN MEMORIA
00165 0195 B6 4000        SUBAL LDA A  APUDAT
00166 0198 A7 00           STA A  00,X
00167 019A B6 4000        LDA A  APUDAT
00168 019D A7 01           STA A  01,X
00169 019F B6 4000        LDA A  APUDAT
00170 01A2 A7 02           STA A  02,X
00171 01A4 B6 4000        LDA A  APUDAT
00172 01A7 A7 03           STA A  03,X
00173 01A9 39             RTS
00174
00175 2FF0 ADC EQU #2FF0
00176 3FF4 DAC EQU #3FF4
00177 3FF6 DACI EQU DAC+2
00178 4000 APUDAT EQU #4000
00179 4008 APUCOM EQU APUDAT+B
00180 0000 ORG #0000
00181 0000 0008 XN RMB 8
00182 0008 0002 RH RMB 2
00183 000A 0002 PH RMB 2
00184 000C 0002 ESPERA RMB 2
00185 000E 0028 RMB 40
00186 END

```

SYMBOL TABLE

```

SALTO2 0049 SALTO1 0053 INICIO 005F SALTO3 009D SALTO4 012D SALTO5 0160
SUBCAR 0180 SUBAL 0195 ADC 2FF0 DAC 3FF4 DACI 3FF6 APUDAT 4000
PH 000A ESPERA 000C
SUBCON 0166 SUB1 0172 SUB2 017A
APUCOM 4008 XN 0000 RH 0008

```

REFERENCIAS

- 1 Hamming, Richard W.: Introduction to applied numerical analysis, Ed. McGraw-Hill, 1971.
- 2 Widrow Bernard et al.: Stationary and nonstationary learning characteristics of the LMS adaptive filter, Proceedings of the IEEE, agosto 1976.
- 3 Widrow Bernard et al.: Adaptive noise cancellation; Principles and Applications, Proceedings of the IEEE, diciembre 1975.

- 4 Box, G.E.P. y Jenkins, G.M.: Time series analysis forecasting and control, Holden Day 1976.
- 5 BURR-BROWN, Analog input microperipheral, PDS-375A, December 1977.
- 6 Advanced Micro Devices, Am 9511A, Arithmetic Processor, 1979.
- 7 BURR-BROWN, Microprocessor analog output components, PDS-363A, December 1977.