PROCESAMIENTO DIGITALES DE SEÑALES

Larry Escobar

PROCESAMIENTO DIGITAL DE SEÑALES (PDS)

El PDS es un área de la ingeniería que agrupa un conjunto de operaciones que se aplican sobre señales discretas, estas operaciones se describen como transformaciones matemáticas.

Algunos objetivos:

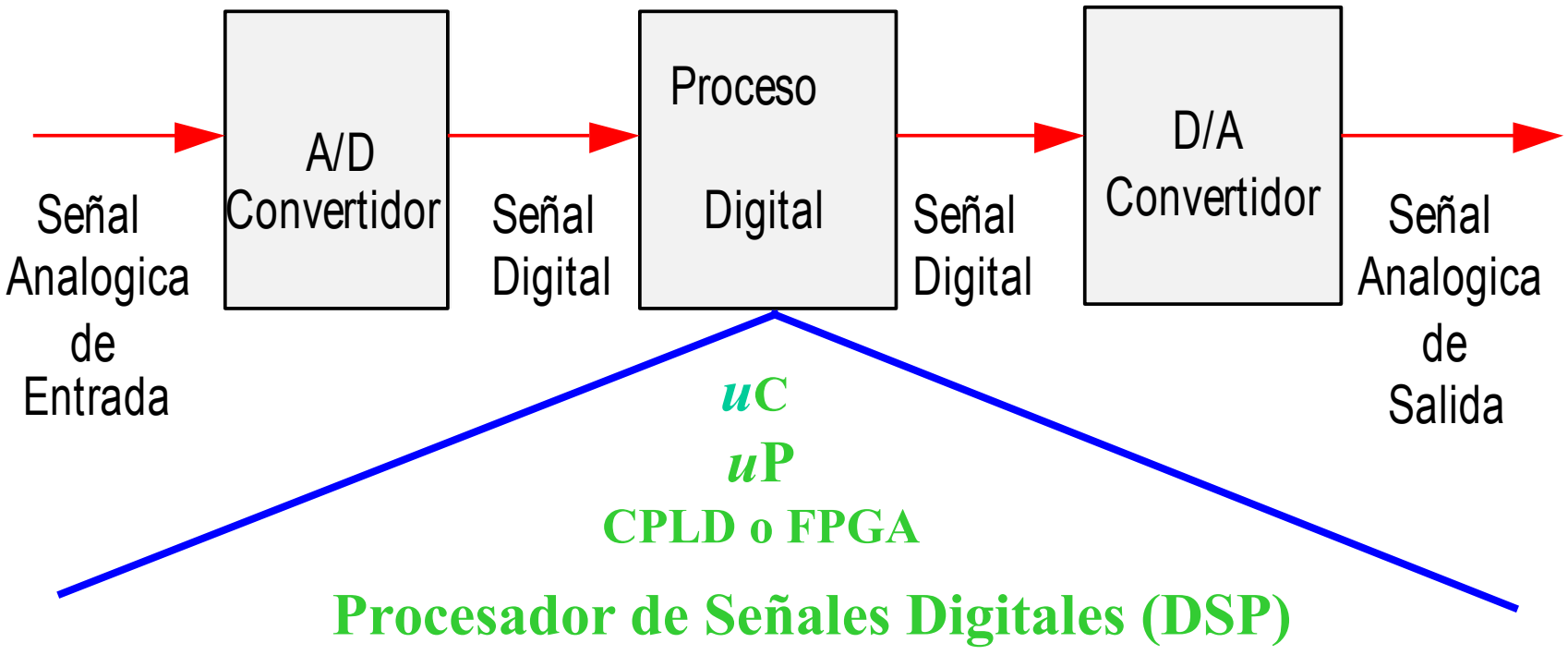
- Proveer una mejor aproximación del análisis o estimación del contenido de la información.

- Analizar, representar, transformar, manipular señales y el contenido de la información.

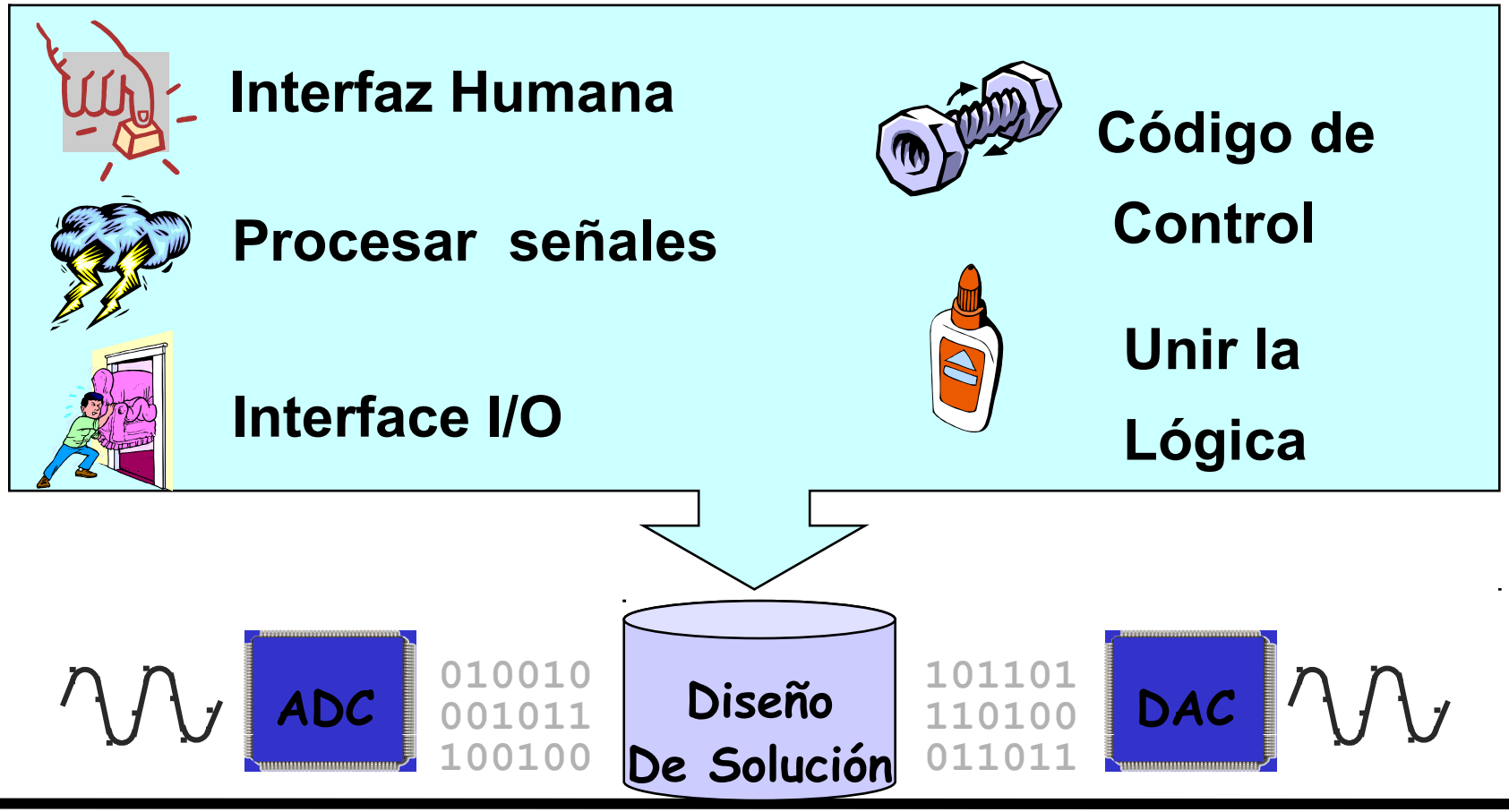
FUNDAMENTOS DEL PDS

- Señales y sistemas
- Matemáticas Discretas
- Probabilidad y Estadística
- Variable compleja, Transformada Z (TZ) y TZI
- Análisis de espectral:
  - Transformada de Fourier en el Tiempo Discreto (DTFT)
  - Transformada Discreta de Fourier (DFT)
  - Transformada Rápida de Fourier (FFT)
  - Métodos paramétricos y no paramétricos
  - Transformada Coseno
- Filtros digitales
- Estimación de parámetros

SISTEMA BASICO DE PDS



DESARROLLO DE UNA SOLUCION



¿Porqué procesar digitalmente?

- Existen procesos que son muy difíciles o casi imposibles de realizar analógicamente
- Ejemplos:
  - Filtros FIR con fase lineal
  - Filtrado Adaptable
- El procesamiento analógico es realizado con: resistores, capacitores, inductores, etc.

La tolerancia inherente de estas componentes, temperatura, cambios de voltaje y vibraciones mecánicas pueden afectar el desempeño de los circuitos analógicos

APLICACIONES DEL PDS

PROCESAMIENTO DE SEÑALES
Convolución
Encriptado, Energía
Codificación, Decodificación
Compresión, Expansión
Procesamiento homomórfico
Ley mu, Ley A, Conversión,
Correlación
Análisis de transitorios

FILTRADO DIGITAL
Respuesta finita al impulso (FIR)
Respuesta infinita al impulso (IIR)
Lattice-Ladder
Windowing
Filtrado adaptable
Eliminación de ruido
Generación de señales

TELECOMUNICACIONES
Modulación
Modems, Telefonía celular
Mutiplexión de canales
Igualación de canal
Cancelación de eco
Video conferencia
Espectro esparcido
Líneas de repetición

ANALISIS ESPECTRAL
Transformada rápida de Fourier (FFT)
Transformada discreta de Fourier (DF
Transformada coseno
Modelo moving average (MA)
Modelo autorregresivo (AR)
Modelo ARMA

PROCESAMIENTO NUMERICO
Operaciones matriciales
Funciones trascendentales
Funciones no lineales
generación de números aleatorios
Aproximaciones numéricas

VOZ
Filtrado
Reconocimiento
Texto a voz
Síntesis de voz
Correo de voz
Comandos de voz

APLICACIONES DEL PDS

CONTROL
Drives de discos
Máquinas
Impresora laser
Motores
Robots
Servo mecanismos
Control numérico
Monitoreo en línea
Seguridad en accesos

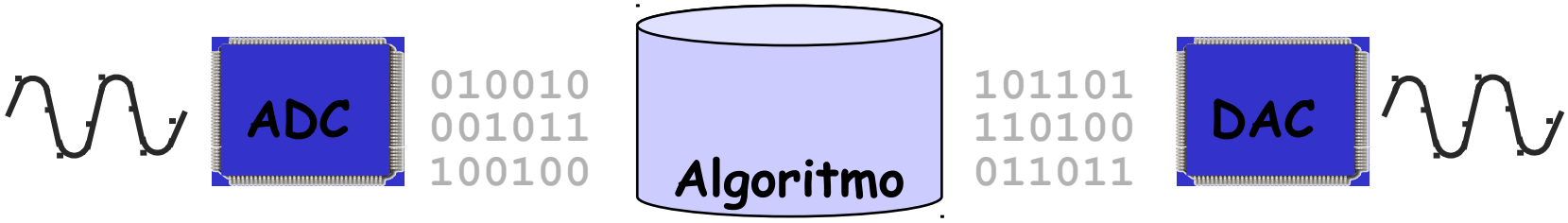
IMÁGENES
Filtrado
Rotación en 3-D
Animación
Realce
Compresión
Reconocimiento de patrones
Compresión y transmisión
Visión de robots
Estaciones de trabajo

OTRAS
Radio y TV digital
Juguetes
Síntesis de música
Potencia
Máquinas
Detección por radar
Posicionamiento global
Análisis de vibraciones
Detección de minerales
Windowing
Transformada de Hilbert

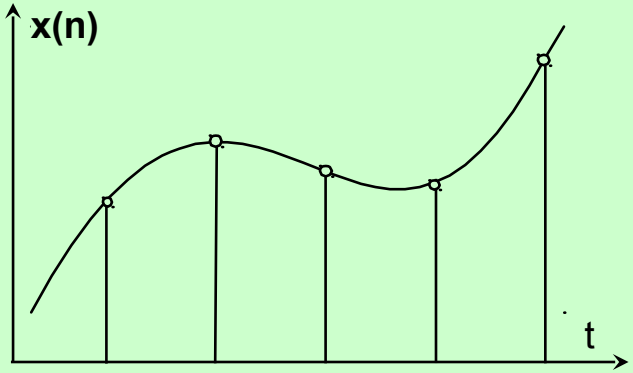
MILITARES
Procesamiento de imágenes
Comandos por voz
Guía de misiles
Navegación
Radar, Sonar
Comunicaciones seguras
Seguimiento de objetos

MEDICINA
Diagnóstico de equipo
Monitoreo fetal
Monitoreo de pacientes
Equipo de ultrasonido
Prótesis
Equipos para el oído

Algoritmo Básico de PDS



Muestreo de una señal analógica

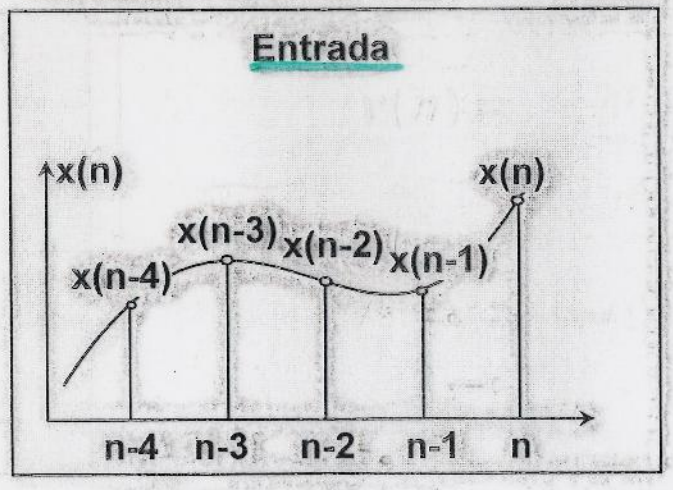


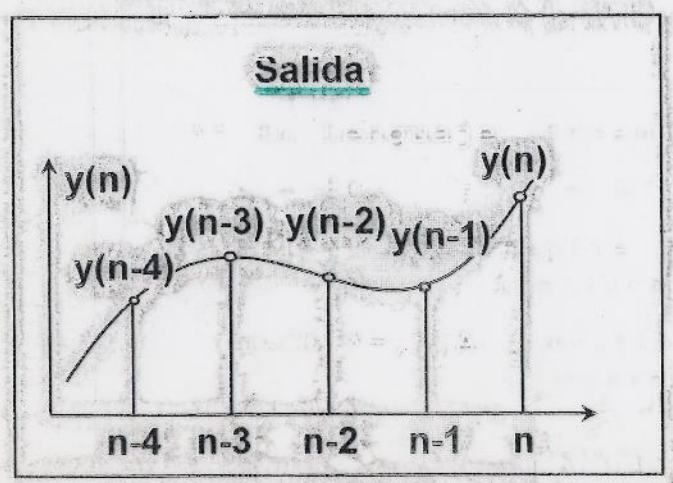
La mayoría de los algoritmos de PDS utilizan la operación convolución:

$$y(n) = \sum_{i=0}^{N-1} x(i)h(n-i)$$

```
for (i = 0; i < N-1; i++){
  sum += h[i] * x[i] }
```

Algoritmo Básico de PDS: La convolución





$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i)$$

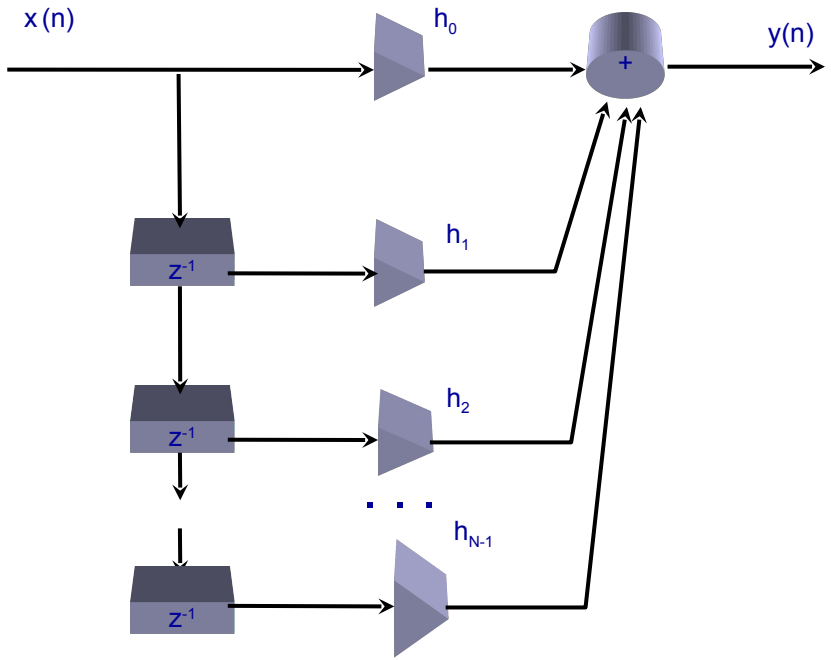
```
/* En lenguaje C    */

    y=0

for (i = 1; i <   N-1; i++){
  y += h[i] * x[i] }
```

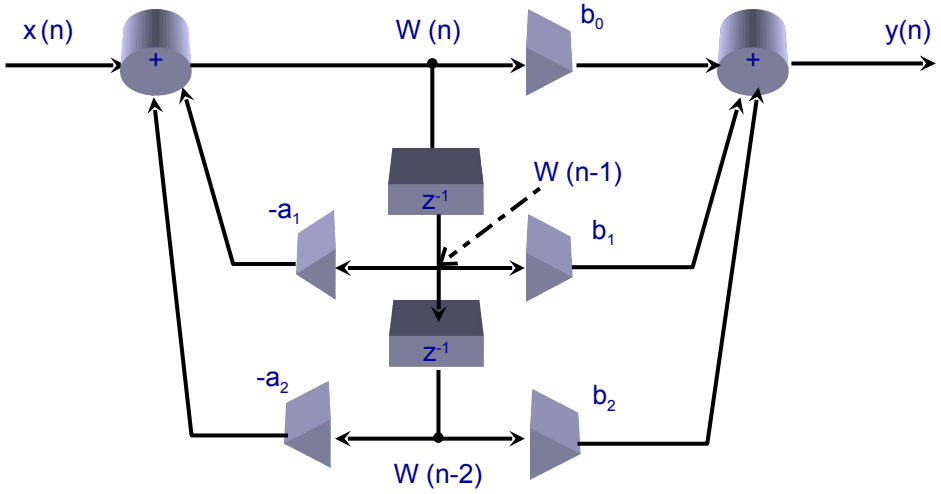
```
** En Lenguaje  Ensamblador
  A = #0      ;  A = 0
  RPT #N_1    ; Repite N veces la siguiente
              ; instrucción
  MACD *-,h,A ; multiplica, acumula y
              ; mueve datos
```

APLICACIONES TIPICAS DE PDS, filtro FIR



$$y(n) = \sum_{k=0}^{N-1} h(k)\, x(n-k)$$

APLICACIONES TIPICAS DE PDS, filtro IIR



$$y(n)=\sum_{i=0}^{q-1} b\,i\,x(n-i)-\sum_{i=1}^{p-1} a\,i\,y(n-i)$$

ENFOQUES DEL PROCESAMIENTO DIGITAL DE SEÑALES

- **Conceptual,** fundamentos del PDS y generación de algoritmos.
- **Algorítmico,** utilización de los algoritmos existentes y la proposición de soluciones a problemas reales (aplicaciones).
- **La implantación de los algoritmos**
  - Validación y verificación de los algoritmos. (Simulación de los algoritmos a través de lenguajes de alto nivel y paquetes).
  - Evaluación de los algoritmos en arquitecturas de DSPs. (Solución algorítmica-software-hardware al problema). Trasladar la aplicación a un prototipo final

Procesamiento Digital de Señales

METODOLOGIA DE IMPLEMENTACION

Permite sistematizar el diseño y desarrollo de actividades del PDS para realizarlas de una manera óptima, proponer etapas y tareas cuyo desarrollo y evaluación conduzcan al objetivo planteado.

1) Definición del problema a resolver

2) Conceptualización del problema: solución matemática (se proponen soluciones)

3) Proposición de una solución basada en técnicas de PDS: solución algorítmica.
La implementación de los algoritmos matemáticos implica hacer una valoración de sus estructuras.

4) Simulación en lenguaje de alto nivel de la propuesta algorítmica: solución software

5) Simulación en el lenguaje de un dispositivo dedicado: solución micro-software

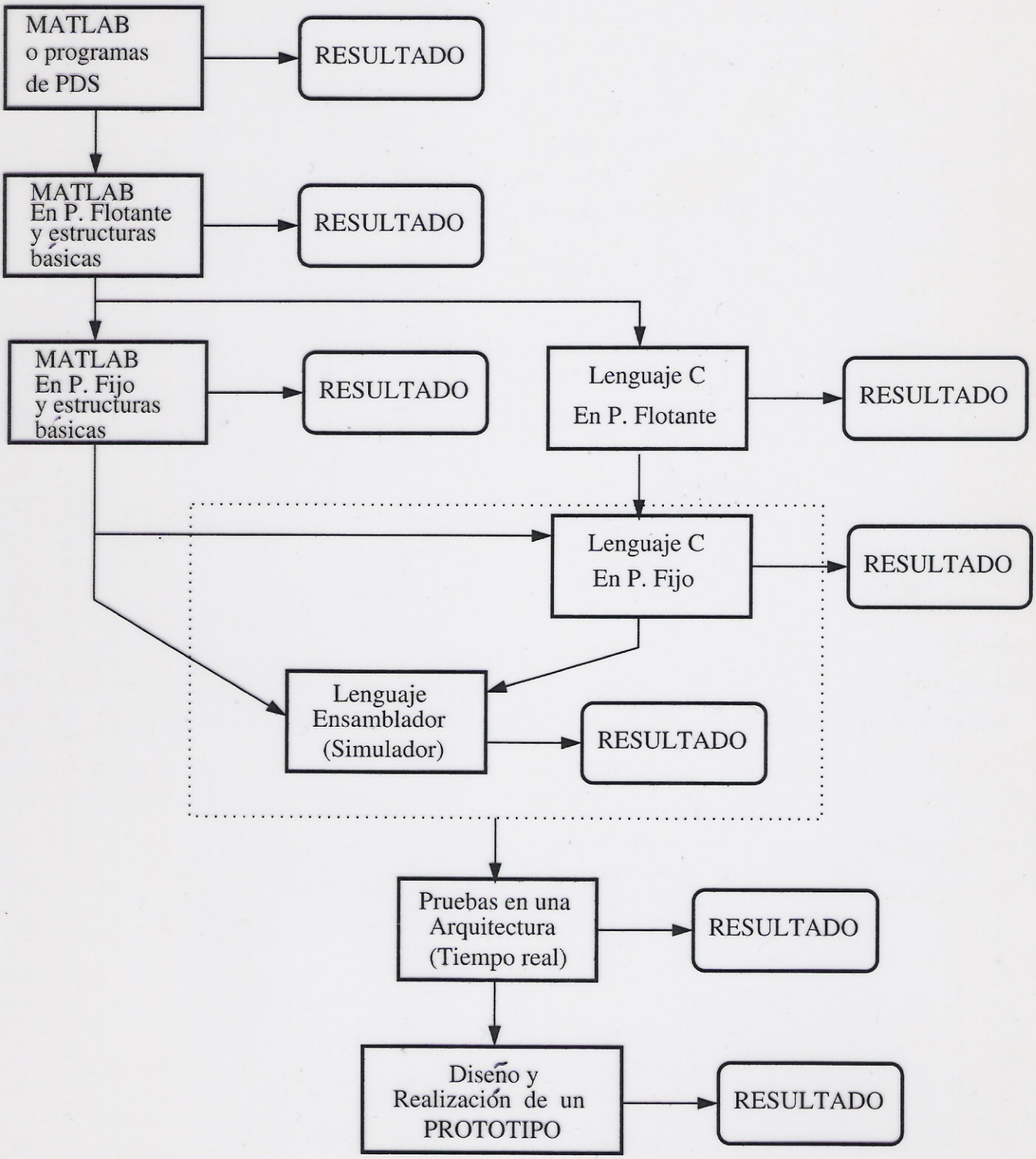
6) Diseño y realización de una arquitectura de cálculo: solución software / hardware
La solución hardware debe coincidir con la solución software

7) Evaluación, pruebas y validación de un prototipo final: solución algorítmica-software-hardware

L. Escobar

PROCESO DE SIMULACION

DE APLICACIONES EN PDS



Algorítmos Básicos en el PDS

Convolución

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i) \tag{1}$$

Ecuación en diferencias

$$y(n) = \sum_{i=0}^{q} b_i x(n-i) - \sum_{j=1}^{p} a_j y(n-j) \tag{2}$$

Sistema FIR

$$y(n) = h_0 x(n) + h_1 x(1) + ...h_{N-1} x(n-N+1) = \sum_{i=0}^{N-1} h(i)x(n-i) \tag{3}$$

Su función de transferencia es la Transformada zeta (TZ) de h(n):

$$H(z) = h_0 + h_1 z^{-1} + ...h_{N-1} z^{-N+1} = \sum_{i=0}^{N-1} h(i) z^{-i} \tag{4}$$

Respuesta al impulso:

$$h(n) = \begin{cases} h_i & 0 \leq i \leq N-1 \\ 0 & \text{otro i} \end{cases} \tag{5}$$

Sistema tipo IIR

$$y(n) = \sum_{i=0}^{q} b(i)x(n-i) - \sum_{i=1}^{p} a(i)y(n-i) \tag{6}$$

Función de transferecia $H(z)$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{B(z)}{A(z)} = \frac{\sum_{i=0}^{q} b(i) z^{-i}}{\sum_{i=1}^{p} b(i) z^{-i}} = \frac{b_0 + b_1 z^{-1} + ... + b_q z^{-q}}{1 + a_1 z^{-1} + ... + a_p z^{-p}}, \qquad a_0 = 1 \tag{7}$$

Operaciones matriciales

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i) = \mathbf{h}^T \mathbf{x} = \mathbf{x}^T \mathbf{h} \tag{8}$$

Correlación

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l) \qquad l = 0, \pm 1, \pm 2, ... \tag{9}$$

La transformada discreta de Fourier (DFT)

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-2\pi kn/N} \qquad k = 0, 1, 2, ..., N-1 \tag{10}$$

Mariposa para la FFT

$$A = a + W_N^r b \qquad \text{a y b son número complejos} \tag{11}$$

$$B = b - W_N^r b \qquad W_N = e^{-2\pi/N} \tag{12}$$

Decimación para realizar la FFT

Índice	Patrón de bits	Patrón decimado
0	0 0 0	0 0 0
1	0 0 1	1 0 0
2	0 1 0	0 1 0
3	0 1 1	1 1 0
4	1 0 0	0 0 1
5	1 0 1	1 0 1
6	1 1 0	0 1 1
7	1 1 1	1 1 1

Tabla 5: Decimación para una señal de ocho puntos

Modelo AR

Ecuaciones de *Yule-Walker*

$$\begin{bmatrix} r_{yy}(0) & r_{yy}(-1) & r_{yy}(-2) & \cdots & r_{yy}(p-1) \\ r_{yy}(1) & r_{yy}(0) & r_{yy}(-1) & \cdots & r_{yy}(p-2) \\ r_{yy}(2) & r_{yy}(1) & r_{yy}(0) & \cdots & r_{yy}(p-3) \\ \vdots & \vdots & \vdots & & \vdots \\ r_{yy}(p-1) & r_{yy}(p-2) & r_{yy}(p-3) & \cdots & r_{yy}(0) \end{bmatrix} \begin{bmatrix} a_p(1) \\ a_p(2) \\ a_p(3) \\ \vdots \\ a_p(p) \end{bmatrix} = - \begin{bmatrix} r_{yy}(1) \\ r_{yy}(2) \\ r_{yy}(3) \\ \vdots \\ r_{yy}(p) \end{bmatrix} \tag{13}$$

Se tiene el sistema $\mathbf{R_p a_p} = \mathbf{r_p}$ y si la matriz R es no singular, la solución es $\mathbf{a_p} = \mathbf{R_p^{-1} r_p}$, esto es similar a la ecuación normal o de Wiener-Hopf.

Método de Levinson-Durbin

El método de Levinson-Durbin es uno de los algoritmos recursivos más importantes en el procesamiento digital de señales. Dados los valores de la autocorrelación $r_{yy}(0), r_{yy}(1), r_{yy}(0), \cdots, r_{yy}(N)$

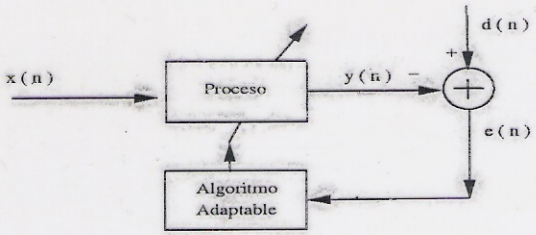
Pasos	Cálculos
1	m=1 $P_1 = r_{yy}(0)$ $k_1 = \alpha_1(1) = -\frac{r_{yy}(1)}{r_{yy}(0)}$
2	m=2,3,...,N $P_m = P_{m-1}[1 - k_{m-1}^2]$ $k_m = \alpha_m(m) = -\frac{r_{yy}(m)+\sum_{n=1}^{m-1} \alpha_{m-1}(n) r_{yy}(m-n)}{P_m}$ $\alpha_m(n) = \alpha_{m-1}(n) + [\alpha_{m-1}(m-n)]k_m$, n=1,2,3,...,m-1
3	$\alpha_N(n) = \alpha_N(n)$, n=2,3,...,N

Tabla 6: Algoritmo de Levinson-Durbin

En la práctica para estimar la autocorrelación se puede utilizar un estimador asintóticamente no sesgado

$$\hat{r}_{yy}(m) = \frac{1}{L_y} \sum_{n=1}^{L_y - |m|} y(n) y(n-m) \qquad |m| << L_y,\ m = 0, 1, 2, \cdots, N \tag{14}$$

FILTRADO ADAPTABLE



Proceso adaptable

Algoritmo LMS

$$\hat{x}(n) = w_p^T(n)x_p(n) \tag{15}$$

$$e(n) = d(n) - \hat{x}(n) \tag{16}$$

$$w_p(n) = w_p(n-1) + \mu e(n)x_p(n) \tag{17}$$

Algoritmo RLS

$$\epsilon_p(n) = d(n) - w_p^T(n)x_p(n) \tag{18}$$

$$K_p(n) = \frac{x_p(n)^T R_p^{-1}(n-1)}{\lambda + x_p(n)^T R_p(n-1)x_p(n)} \tag{19}$$

$$w_p(n) = w_p(n-1) + \epsilon_p(n)K_p(n) \tag{20}$$

$$R_p^{-1}(n) = \frac{1}{\lambda}\left[R_p^{-1}(n-1) - \frac{R_p^{-1}(n-1)x_p(n)x_p^T(n)R_p^{-1}(n-1)}{\lambda + x_p^T(n)R_p^{-1}(n-1)x_p(n)}\right] \tag{21}$$

Algoritmo Rápido de Kalman o FRLS

$$e_p^f(n) = y(n) + A_p^T(n-1)Y_p(n) \tag{22}$$

$$A_p(n) = A_p(n-1) + K_p(n-1)e_p^{fT}(n) \tag{23}$$

$$\epsilon_p^f(n) = y(n) + A_p^T(n)Y_p(n-1) \tag{24}$$

$$\alpha_p^f(n) = \lambda\alpha_p^f(n-1) + \epsilon_p^f(n)e_p^{fT}(n) \tag{25}$$

$$K_{p+1}(n) = \begin{bmatrix} 0 \\ K_p(n-1) \end{bmatrix} - \begin{bmatrix} I \\ A_p(n) \end{bmatrix} \frac{\epsilon_p^f(n)}{\alpha_p^f(n)} = \begin{vmatrix} M_p(n) \\ \mu(n) \end{vmatrix} \tag{26}$$

$$e_p^b(n) = y(n-p) + B_p^T(n-1)Y_p(n) \tag{27}$$

$$K_p(n) = \frac{M_p(n) - B_p(n-1)\mu(n)}{1 + e_p^{bT}\mu(n)} \tag{28}$$

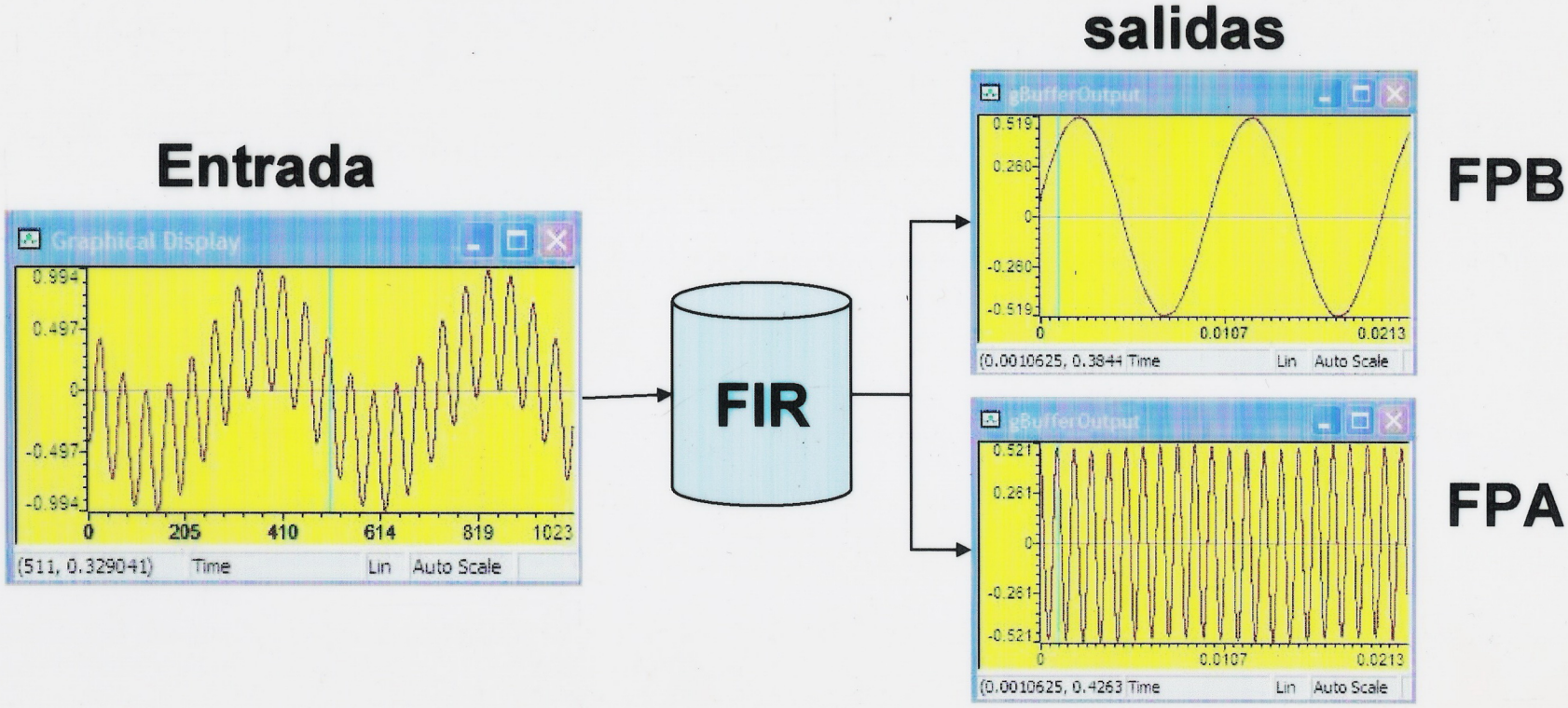
$$B_p(n) = B_p(n-1) + K_p(n-1)Y_p(n) \tag{29}$$

$$e_p(n) = d(n) + H_p^T(n-1)Y_p(n) \tag{30}$$

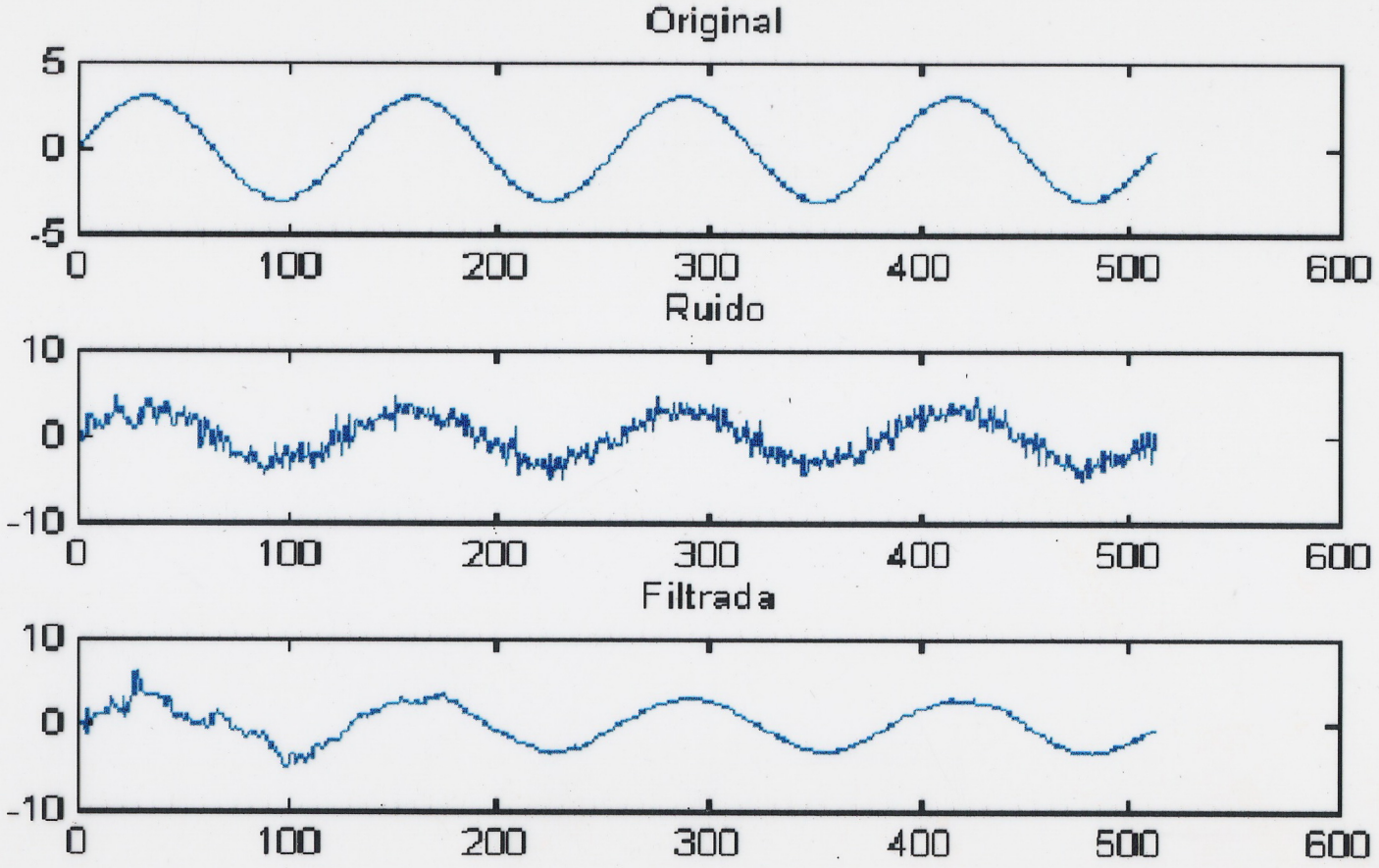
$$H_p(n) = H_p(n-1) + K_p(n)e_p^T(n) \tag{31}$$

Aplicación

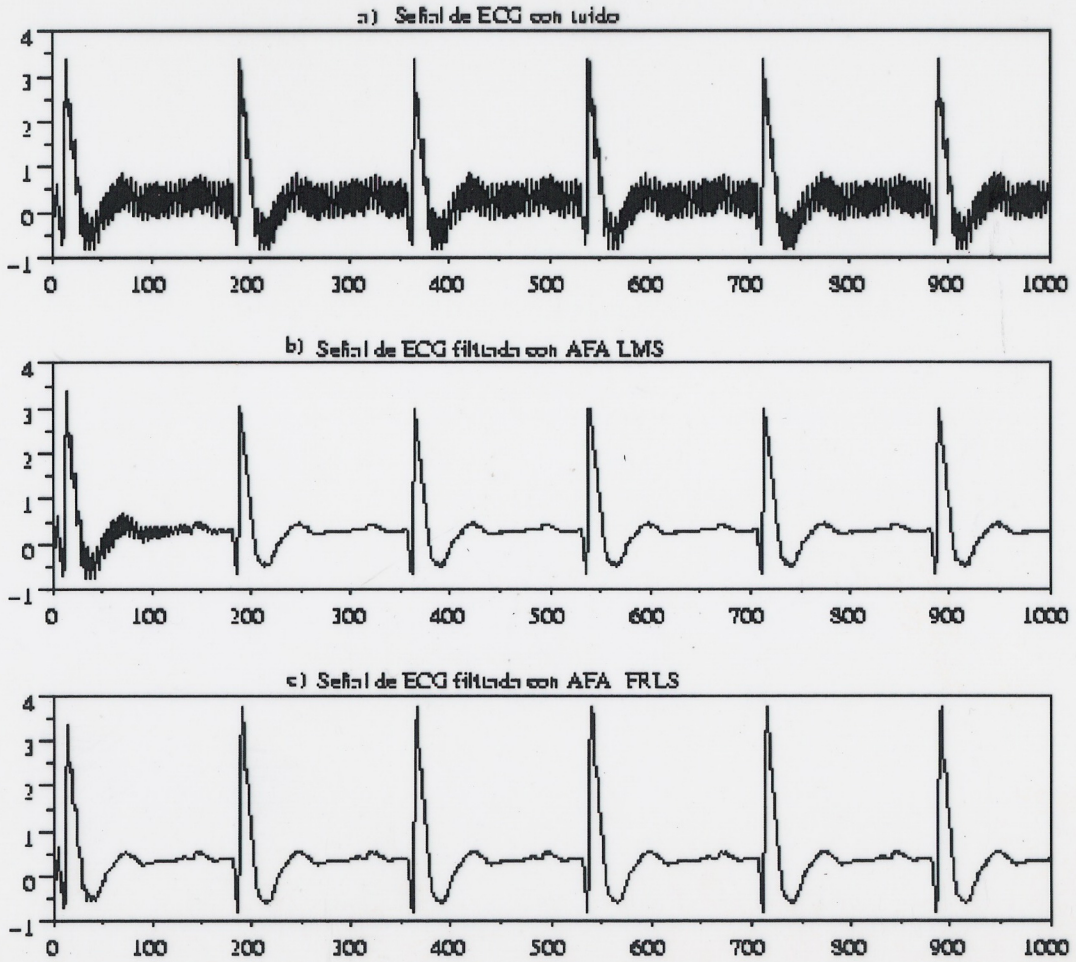
Filtrado de una señal con senoides



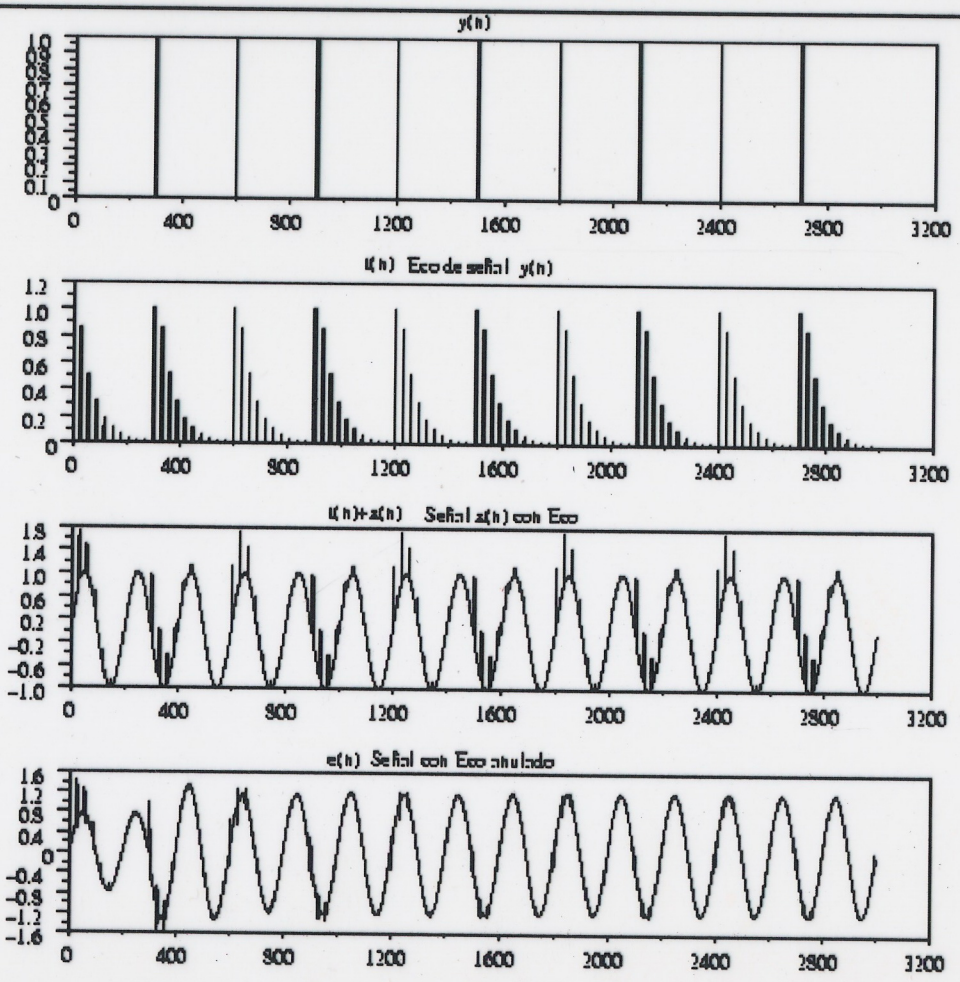
AFA: En la cancelaciónde ruido



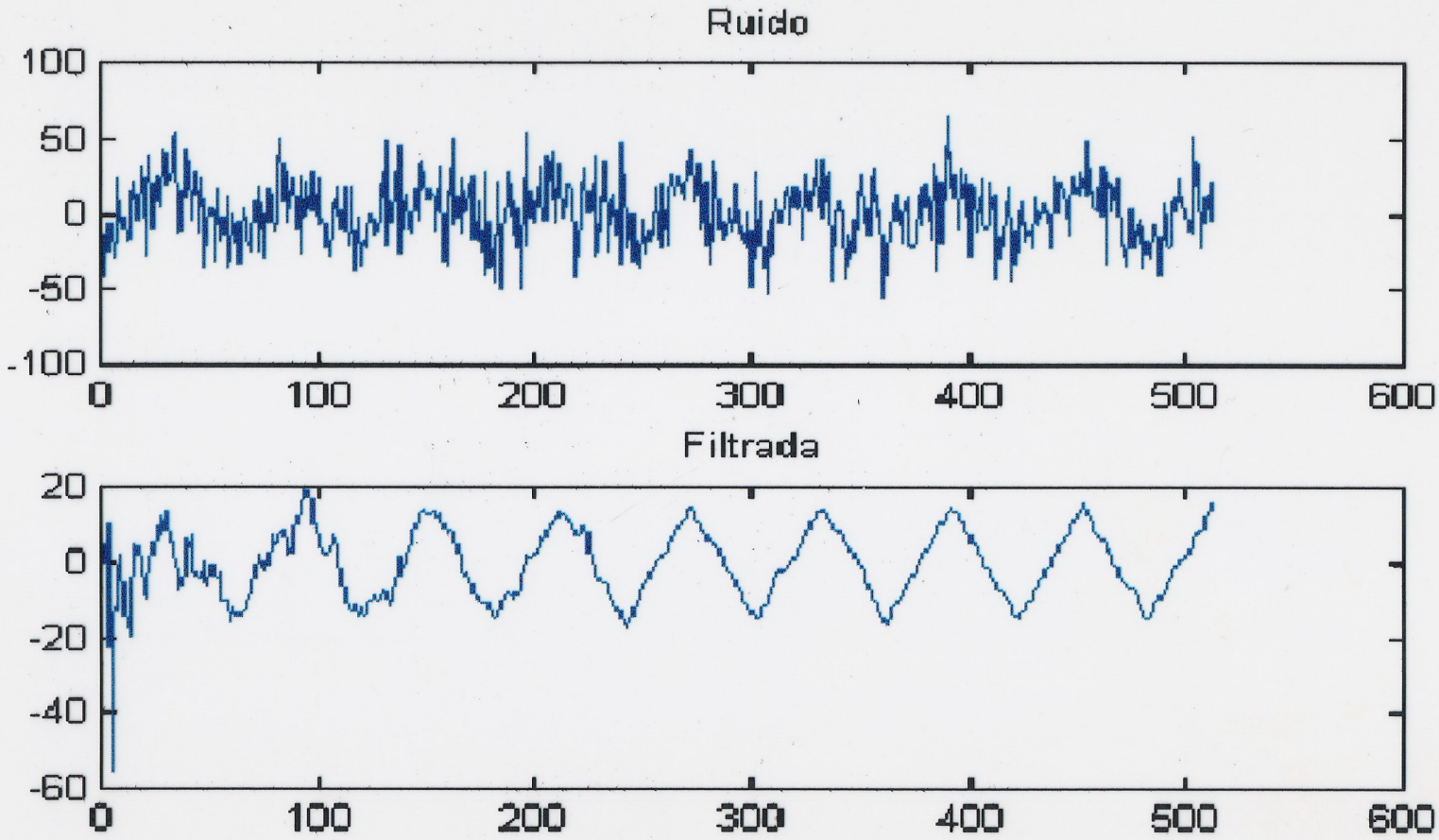
AFA: En la cancelaciónde ruido



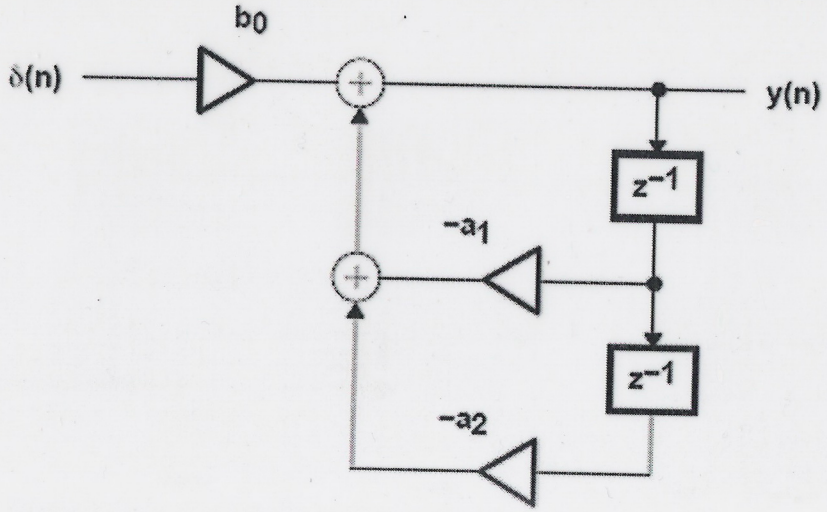
AFA: En la cancelación de Eco , Resultados



AFA: En la cancelaciónde ruido



Oscilador



$$y(n) = -a_1 y(n-1) - a_2 y(n-2) + b_0 \delta(n)$$

$$y(-1) = y(-2) = 0,$$

$$a_1 = -2\cos(\omega_0), \quad a_2 = 1,$$

$$b_0 = A\sin(\omega_0), \quad w_0 = \frac{2\pi f_0}{f_s}.$$

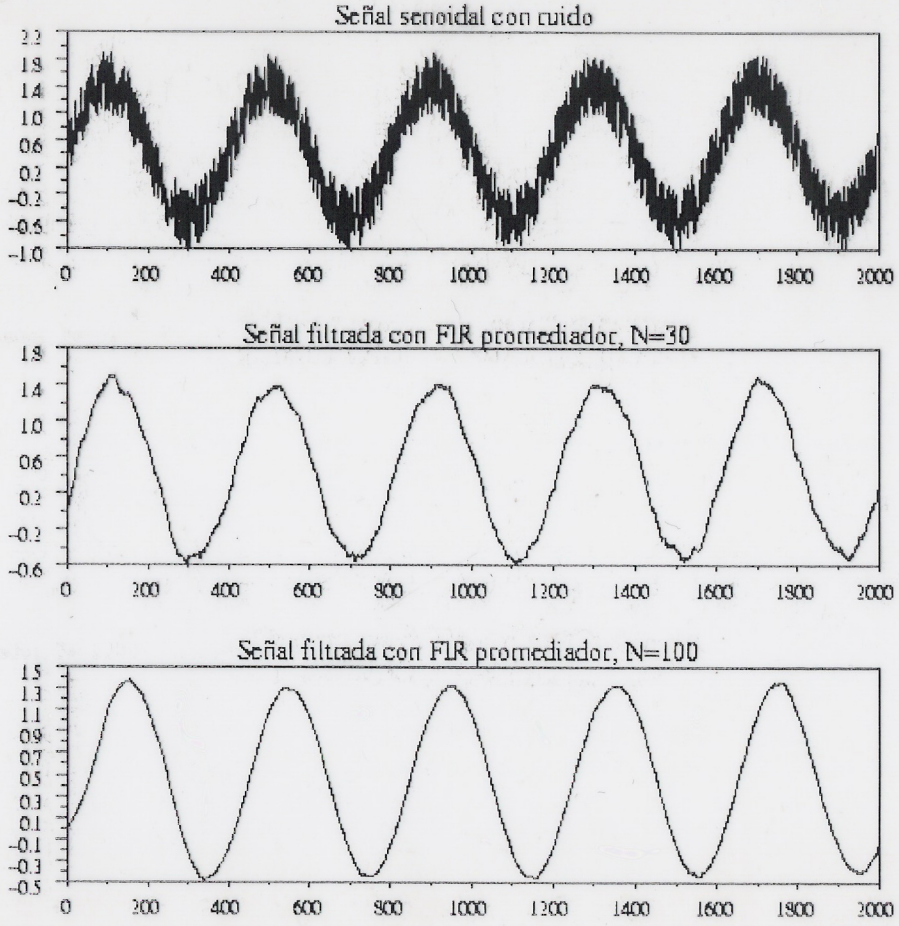
La Suma de Productos

es el elemento clave del PDS

Algorithm	Equation
Finite Impulse Response Filter	$y(n)=\sum_{k=0}^{M} a_k\, x(n-k)$
Infinite Impulse Response Filter	$y(n)=\sum_{k=0}^{M} a_k\, x(n-k)+\sum_{k=1}^{N} b_k\, y(n-k)$
Convolution	$y(n)=\sum_{k=0}^{N} x(k)h(n-k)$
Discrete Fourier Transform	$X(k)=\sum_{n=0}^{N-1} x(n)\exp[-j(2\pi/N)nk]$
Discrete Cosine Transform	$F(u)=\sum_{x=0}^{N-1} c(u).f(x).\cos\left[\frac{\pi}{2N}u(2x+1)\right]$

L. Escobar

Filtrado con FIR promediador



L. Escobar