Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas.
Departamento Ingeniería Eléctrica.

Capítulo 2.

Procesamiento Digital de Señales.

2.0 Repaso: Transformada de Fourier
2.1 Muesteo de la Señal.
2.2 Análisis de Señales Discretas.
2.3 Filtros Digitales
2.4 DTF y FFT
2.5 Ventana de Hamming
2.6 LCP
2.7 Cuantización Vectorial.

Profesor: Néstor Becerra

Repaso: Transformada de Fourier

- **Significado matemático:**

Representación de una señal dad en términos de una suma infinita de exponenciales complejas ponderadas cada una por F(w)df

- **Significado en aplicaciones: Completar**

Repaso: Transformada de Fourier

$$f_T(t) = \sum_{n=-\infty}^{\infty} F_n e^{jnw_o t}$$

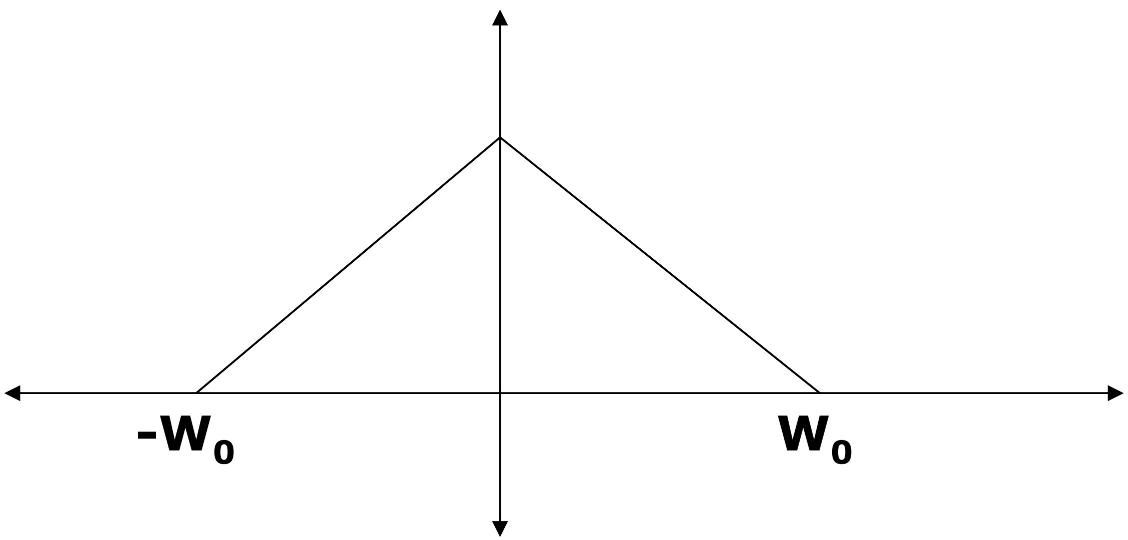
$$F_n = \frac{1}{T} \int f_T(t) e^{-jnw_o t} dt$$

$$w_o = \frac{2\pi}{T}$$

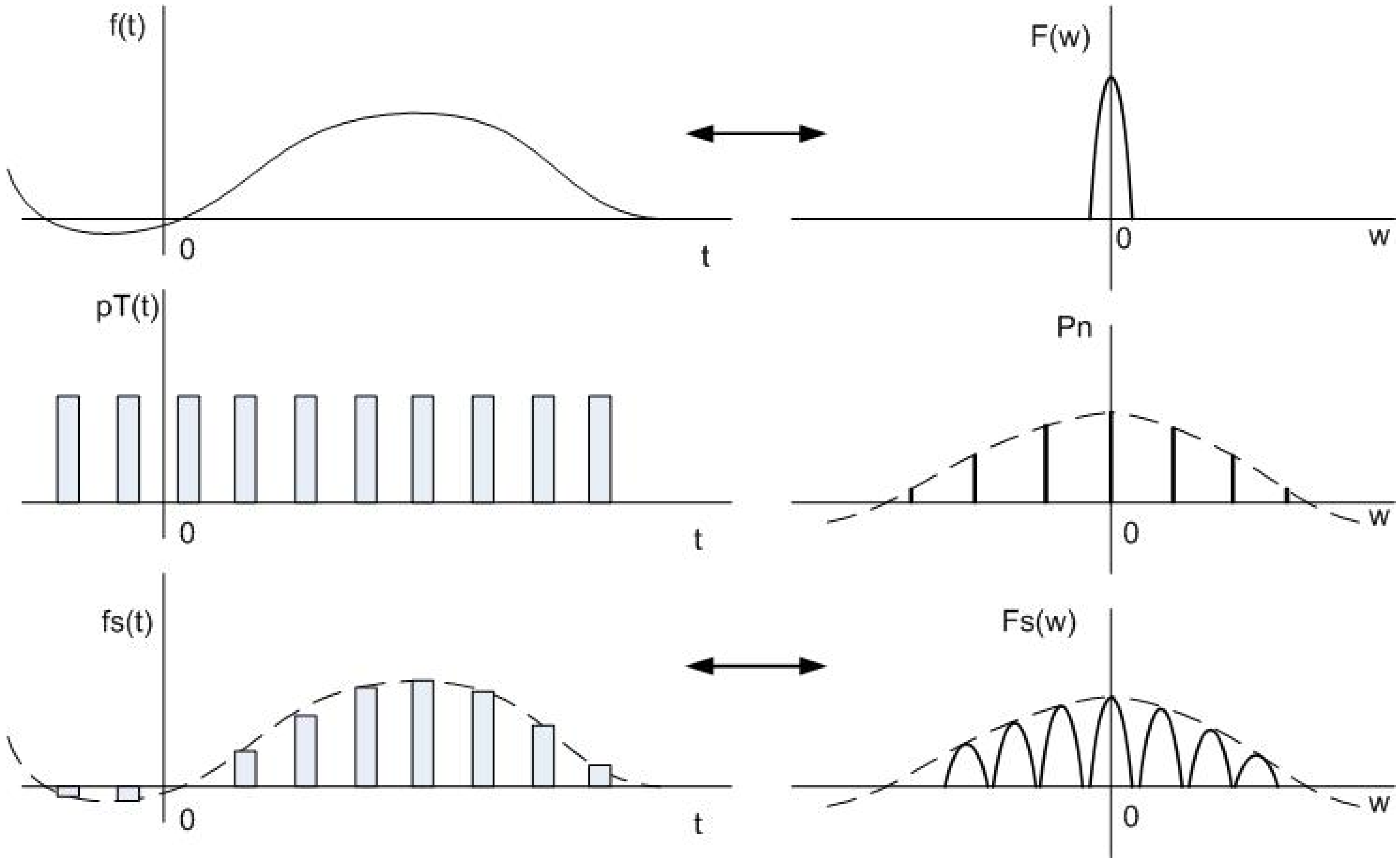
2.1. Muestreo de la señal.

- Consideremos una señal s(t), donde t es el tiempo.

$$s(t) \leftrightarrow S(W)$$

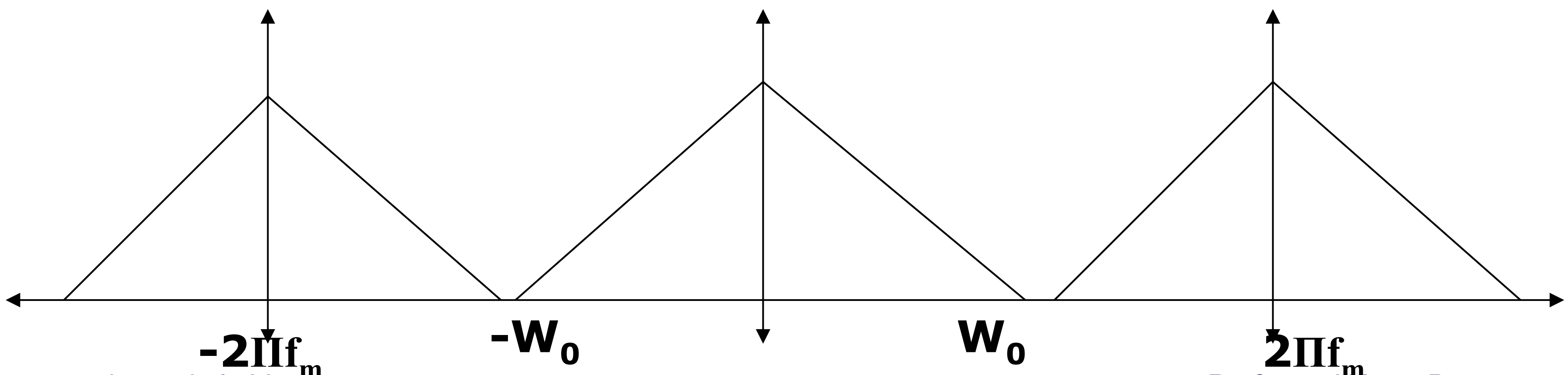


2.1 Muestreo de la señal



2.1. Muestreo de la señal

- Supongamos que la señal s(t) es muestreada a intervalos regulares T (fn=1/T, fn es la frecuencia de muestreo). El espectro de la señal muestreada es representado por:



2.1 Muestreo de la señal

- Para que no haya sobre posición,

$$2\pi f_m > 2W_0 = 2 * 2\pi f_0$$

$$f_m > 2 f_0$$

f_0 Es la máxima frecuencia de la señal a muestrear.

2.2 Análisis de Señales Discretas

- Una señal muestreada podría ser representada por S(nT), donde n es un número entero y T es el período de muestreo.
- En realidad las señales muestreadas se pueden representar como secuencias de números enteros (digitalizadas) .
- Indexadas por números enteros se representan:

$$s(0), s(1), s(2), s(3), \ldots\ldots$$

2.2 Análisis de Señales Discretas

- Todos los tipos de procesamiento en electrónica analógica se pueden implementar en el dominio discreto.

  Ejemplos: Amplificadores, filtros, multiplicadores, etc.

- Otras funciones más complejas se pueden realizar mediante operaciones aritméticas en programas de computadores.

2.3 Filtros Digitales

- Dos tipos: FIR,IIR

FIR (Finite Impulse Response):

•La salida es función únicamente de la entrada .

•El polinomio que describe este filtro:

$$Y(n) = a_0 x(n) + a_1 x(n-1) + a_2 x(n-2) + \dots\dots\dots + a_k x(n-k)$$

•Un filtro FIR también se puede definir por:

$$A = [a_0 \quad a_1 \quad a_2 \qquad\qquad a_k] \quad \text{y} \quad k$$

¿Porqué Respuesta Impulsiva Finita?

2.3 Filtros Digitales

IIR (Infinite Impulse Response):

•La salida en n depende de la entrada y de la salida hasta n-1.

•El polinomio que describe este filtro:

$$Y(n) = a_0 x(n) + a_1 x(n-1) + a_2 x(n-2) + \ldots\ldots\ldots + a_k x(n-k)$$
$$+ b_1 y(n-1) + b_2 y(n-2) + \ldots\ldots\ldots + b_L y(n-L)$$

•Un filtro IIR también se puede definir por:

$A = [a_0 \quad a_1 \quad a_2 \qquad\qquad a_k]$ y k

¿Porqué Respuesta Impulsiva Infinita?

¿Estabilidad?

Anexo: Transformada Z

Se emplea para analizar sistemas discretos.
(De forma similar como opera Laplace para sistemas continuos).

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$

Mediante la propiedades de la transformada Z se puede concluir:

$$x(n) \leftrightarrow X(z)$$

$$x(n-k) \leftrightarrow X(z) z^{-k}$$

Anexo: Transformada Z

Ejemplo:

Considere el filtro FIR

$$Z\{y(n)\} = Z\{a_0 x(n) + a_1 x(n-1) + a_2 x(n-2) + \ldots\ldots + a_k x(n-k)\}$$

$$Z\{y(n)\} = Z\{a_0 x(n)\} + Z\{a_1 x(n-1)\} + Z\{a_2 x(n-2)\} + \ldots\ldots + Z\{a_k x(n-k)\}$$

$$Z\{y(n)\} = a_0 Z\{x(n)\} + a_1 Z\{x(n-1)\} + a_2 Z\{x(n-2)\} + \ldots\ldots + a_k Z\{x(n-k)\}$$

$$Y(z) = a_0 X(z) + a_1 X(z) z^{-1} + a_2 X(z) z^{-2} + \ldots\ldots + a_k X(z) z^{-k}$$

$$\frac{Y(z)}{X(z)} = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots\ldots + a_k z^{-k}$$

Anexo: Respuesta en Frecuencia

- La respuesta en frecuencia de un sistema discreto se puede estimar reemplazando Z por:

$$z = l^{+jw} = \cos(w) + jsen(w)$$

Ejemplo: Sea $H(z) = 1 + az^{-1}$

$$H(e^{+jw}) = 1 + ae^{+jw} = 1 + a\left[\cos(w) - jsen(w)\right]$$

$$H(e^{+jw}) = 1 + a \cdot \cos(w) - j \cdot a \cdot sen(w)$$

$$\left|H(e^{+jw})\right| = \sqrt{1 + 2a\cos(w) + a^2}$$

$$\angle H(e^{+jw}) = \arctan\left(\frac{-asen(w)}{1 + a\cos(w)}\right)$$

2.4 DFT y FFT

2.4.1 DFT (Discrete Fourier Transform)

Una secuencia de N muestras de valor complejo en el dominio de la frecuencia dado por:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} \qquad 0 \le k \le N-1$$

Donde:

$$W_N = e^{-j\frac{2\pi}{N}}$$

2.4 DFT y FFT

2.4.1 DFT (Discrete Fourier Transform)

La trasformada inversa se define como:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) W_N^{-kn} \qquad 0 \le n \le N-1$$

Donde:

$$W_N = e^{-j\frac{2\pi}{N}}$$

- X(K) es una función periódica con periodo N.
- La propiedades de la DFT son similares a las propiedades de la transformada de Fourier análoga.

2.4 DFT y FFT

2.4.2 FFT(Fast Fourier Transform)

- La DTF requiere N^2 multiplicadores.
- Empleo de simetría.
- Algoritmo mas eficiente de cálculo.
- Calcula N componentes de frecuencia a partir de N muestras de tiempo para $N=2^r$, con r cualquier entero postivo.

2.5 Ventana de Hamming

- Para reducir el efecto de discontinuidad al mínimo debemos emplear tipos de ventana que tiendan a reducir a 0 los valores de las muestras en los extremos.

- Aunque existe un buen número de tipos de ventana, la más común en el análisis de la voz es la que se conoce como ventana de Hamming.

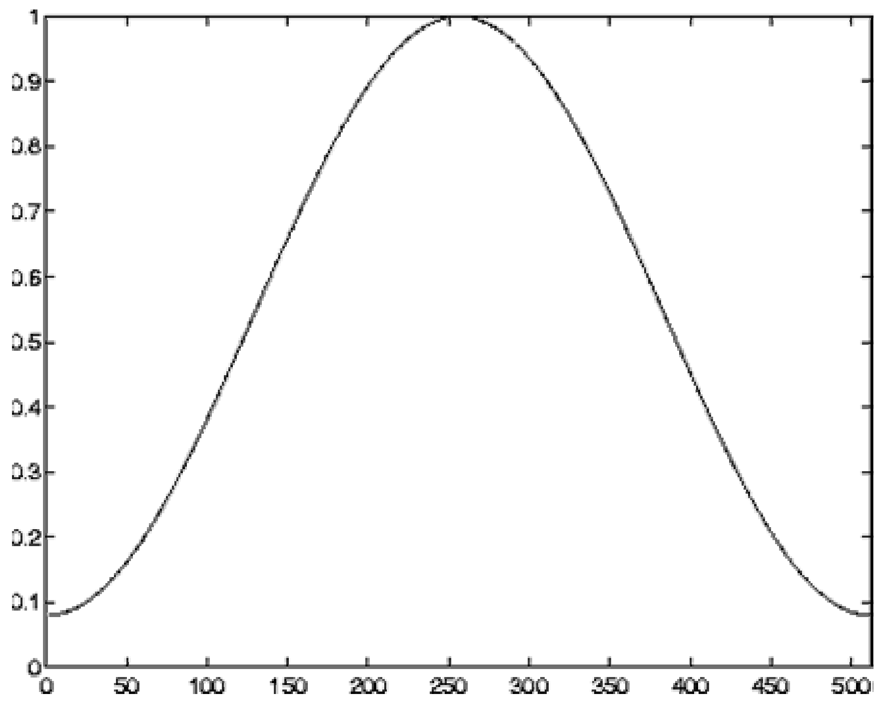
2.5 Ventana de Hamming

$$JH(n) = \begin{cases} 0.54 - 0.46 * \cos\left[\dfrac{2\pi n}{N-1}\right] & 0 \le n < N \\ 0 & n < 0 \cdots N \le n \end{cases}$$

Se recomienda que la ventana sea de 10 a 30 ms, pues en este periodo la señal de voz puede considerarse estacionaria o quasi-estacionaria.

2.5 Ventana de Hamming

Forma genérica de la ventana de Hamming



2.6 LPC (Linear Predictive Coding)

- En 1927 se propuso que una serie temporal de muestras correlacionadas podría ser generada a partir de una serie de muestras estadísticamente independientes (ruido blanco) procesadas a través de un filtro lineal.
- Existen métodos distintos de modelamiento que están basados en procesos estocásticos y/o procesos AR(autoregresivos).

2.6 LPC (Linear Predictive Coding)

Modelo Auto Regresivo.

- Es un proceso estocástico genérico.
- La idea principal se basa en que la voz puede modelarse a través de una combinación lineal de p muestras anteriores más una señal de excitación o ruido blanco.

$$x[n] = \sum_{k=1}^{p} a_k x[n-k] + e[n]$$

Donde	$x(n)$	Coeficientes de la señal
	a_i	Habitualmente son LPC
	$e[n]$	Ruido blanco

2.6 LPC (Linear Predictive Coding)

Aplicando transformada Z, a la formula anterior, también conocida como error de predicción.

$$x[n] = \sum_{k=1}^{p} a_k x[n-k] + e[n]$$

$$X(z) = H_I(z) E(z)$$

$$H_I = \frac{1}{\sum_{k=0}^{p} a_k z^{-k}}$$

Considerando:

$$a_o = 1$$

$$a_k = -\alpha_k$$

$$1 \le k \le p$$

$H_I(z)$ corresponde a un filtro IIR

2.6 LPC (Linear Predictive Coding)

Modelo Auto Regresivo.

El teorema de Wold dice que *"cualquier proceso estocástico discreto en el tiempo y estacionario puede descomponerse como la suma de un proceso autoregresivo genérico y un proceso predictivo, ambos no correlacionados".*

$$y(n) = x(n) + s(n)$$

$y(n)$ Proceso estocástico discreto

$x(n)$ Proceso no correlacionado.

$s(n)$ Proceso predictivo.

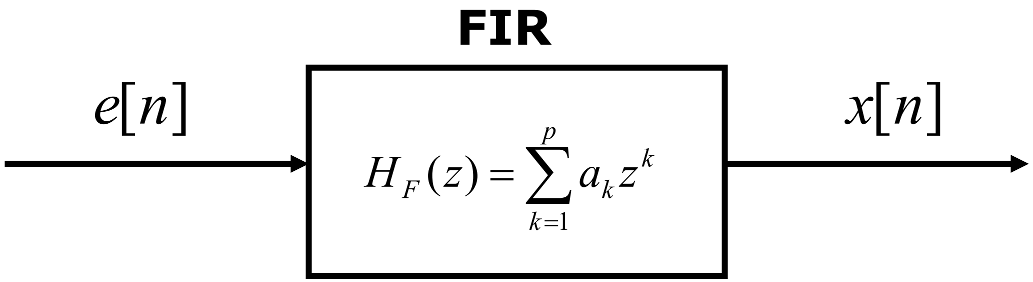
2.6 LPC (Linear Predictive Coding)

Modelo Auto Regresivo.

Un proceso regular genérico esta representado:

$$x[n]=\sum_{k=0}^{p} b_k e[n-k]$$

$$X(z)=H_F(z)E(z)$$



H_f corresponde a un filtro FIR de fase mínima y puede ser reemplazado por un filtro IIR como el H_I esta fórmula también se conoce como el predictor de orden p.

2.6 LPC (Linear Predictive Coding)

Modelo Auto Regresivo.

Un análisis de LPC puede ser vista como un modelamiento AR de una señal de voz en intervalos de tiempo de la señal son considerados estacionarios. La señal multiplicada por una ventana de tiempo centrada en 1 esta dada por:

$$x(n,1) = v(n)w(i-1)$$

Simplificando la notación:

$$x(n) = v(n)w(n)$$

2.6 LPC (Linear Predictive Coding)

Para determinar los coeficientes del LCP se trabaja a partir de la ecuaciones de Yule-Walker.(Basadas error cuadrático medio)

$$E = \sum_{n=-\infty}^{\infty} e^2(n) = \sum_{n=-\infty}^{\infty} \left[x(n) - \sum_{k=1}^{p} \alpha_k x(n-k) \right]^2$$

$$\frac{\partial E}{\partial \alpha_k} = 0 \qquad k = 1,2,3,\ldots\ldots, p$$

2.6 LPC (Linear Predictive Coding)

Luego se tiene p ecuaciones lineales:

$$\sum_{n=-\infty}^{\infty} x(n-i)x(n) = \sum_{k=1}^{p} \alpha_k \sum_{i=-\infty}^{\infty} x(n-i)x(n-k)$$

$$i = 1,2,3,\ldots\ldots, p$$

Definimos:

$$R(i) = \sum_{n=i}^{N-1} x(n)x(n-i)$$

$$i = 1,2,3,\ldots\ldots, p$$

Reemplazando:

$$\sum_{k=1}^{p} \alpha_k R(i-k) = R(i) \qquad \textbf{(**)}$$

$$i = 1,2,3,\ldots\ldots, p$$

2.6 LPC (Linear Predictive Coding)

De las ecuaciones anteriores tenemos que, la energía residual mínima o el error de predicción mínima E_p para un modelo de p polos:

$$E_p = R(0) - \sum_{k=1}^{p} \alpha_k R(k)$$

La ecuación (**) en forma matricial:

$$\begin{bmatrix} R(0) & R(1) & R(2) & \dots & R(p-1) \\ R(1) & R(0) & R(1) & \dots & R(p-2) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \vdots & \dots & \vdots \\ R(p-1) & R(p-2) & R(p-3) & \dots & R(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ \vdots \\ R(p) \end{bmatrix}$$

2.6 LPC (Linear Predictive Coding)

El sistema escrito en forma matricial se define:

$$R \; * \; A \; = \; r$$

Donde R matriz de p X p. Invirtiendo la matriz R se puede obtener el vector A de coeficientes del LPC.

Sin embargo existe un método más eficiente, que consiste en aprovechar las simetrías de la matriz.

2.6 LPC (Linear Predictive Coding)

Método de Levinson-Durbin

$$k_m = \frac{R(m) - \sum_{k=1}^{m-1} \alpha_k^{m-1} R(m-k)}{E_{m-1}}$$

$$\alpha_m^m = k_m$$

$$\alpha_k^m = \alpha_k^{m-1} - k_m \alpha_{m-k}^{m-1}$$

$$1 \le k \le m-1$$

2.6 LPC (Linear Predictive Coding)

Método de Levinson-Durbin

$$E_m = \left(1 - k_m^2\right) \cdot E_{m-1}$$

Donde inicialmente: $E_0 = R(0)$

$$\alpha_0 = 0$$

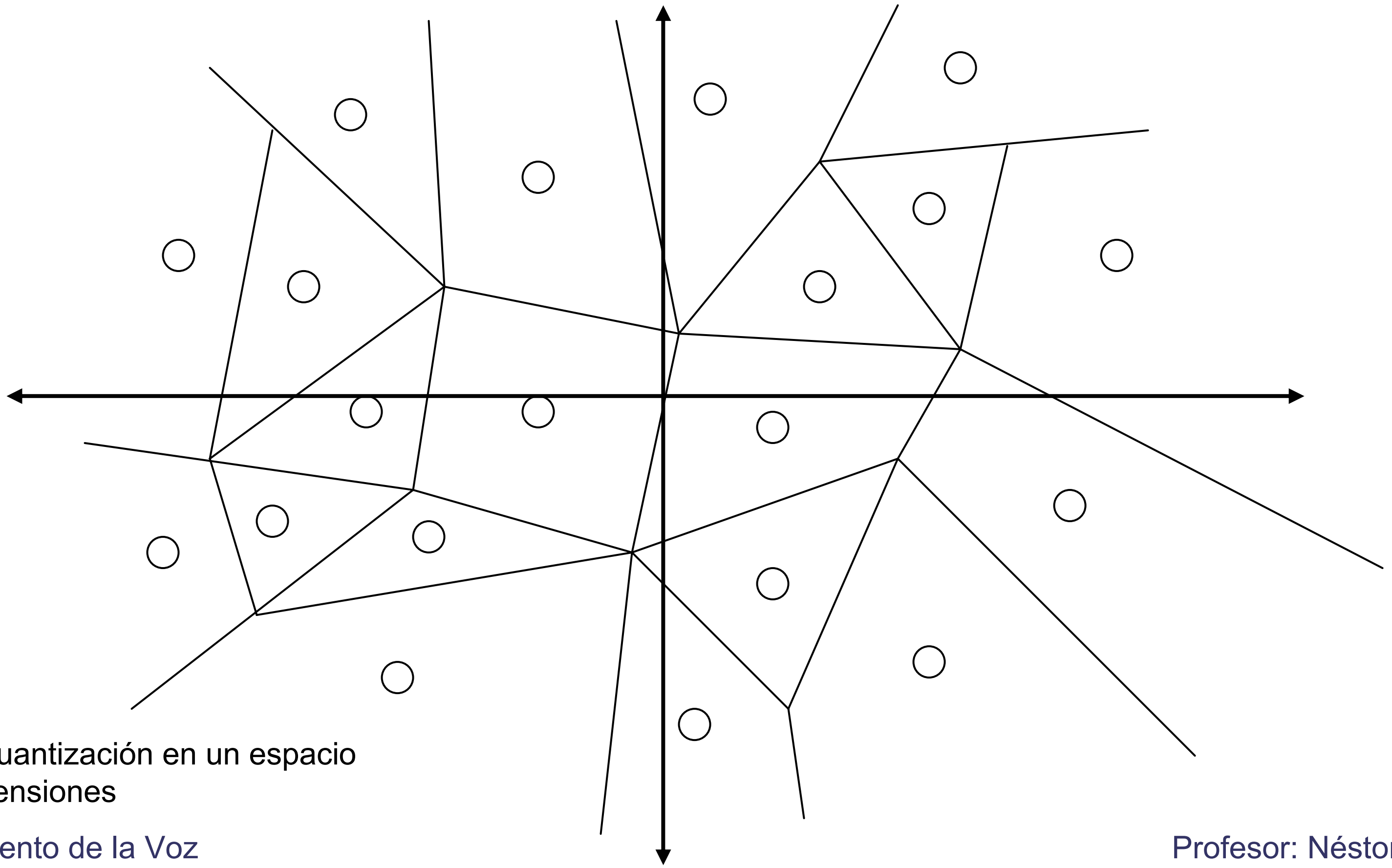
Donde α_k^m indicando el coeficiente LCP en la m-ésima iteración.

2.7 Cuantización Vectorial

- La **"cuantización vectorial"** es un proceso de aproximación por el cual una señal cuya amplitud presenta valores continuos es representada por una nueva señal cuya amplitud se puede asumir con valores discretos.
- La cuantización es una herramienta muy útil para codificación pues el número de bits a representar es una variable que puede ser reducida cuanto se quiera cumpliendo una exigencia de distorsión máxima exigida.

2.7 Cuantización Vectorial

$$q(T) = z_1 \mid d(X, z_1) \leq d(X, z_j)$$



Ejemplo de cuantización en un espacio de varias dimensiones

2.7 Cuantización Vectorial

- Un conjunto de todos los z_i $1 \leq i \leq L$ es denominado el codebook (Libro de código).
- Un método comúnmente utilizado en la elaboración de un codebook es el algoritmo **"k-means".** Este método emplea una minimización de la distorción global, definida como:

$$D = \sum_{i=1}^{L} D_i$$

Donde D_i es la distorsión dentro del la celda i.

2.7 Cuantización Vectorial

Donde $d(T,z_i)$ es la distancia entre un padrao de teste localizado en la celda C_{i}.

$$D_i = \frac{1}{N_i} \sum_{T \in C_i} d(T, z_i)$$

$$z_i = \frac{1}{N_i} \sum_{T \in C_i} T$$

2.7 Cuantización Vectorial

- **Algoritmo de K-means.**

Paso1. Inicialización, escoger utilizando un método adecuado un codebook.

Paso2. Clasificación, clasificar cada vector T (cuadro)…completar….

Paso3. Actualización de Codebook recalcular cada code-word a través de la media aritmética de dos elementos no interiores a la celda correspondiente.

2.7 Cuantización Vectorial

- **Algoritmo de K-means.**

Paso 4. Test de Convergencia completar.