

IMPLEMENTACION PROPIA DE LA FUNCION PBURG PARA LA OBTENCION DEL ESPECTRO DE POTENCIAS

cREACION E IMPLEMENTACION DE UNA FUNCION EN MATLAB QUE REALICE EL METODO DE BURG PARA LA OBTENCION DE COEFICIENTESDE REGRESION Y ESTIMACION DE ESPECTRO DE POTENCIAS

PhD. rOBIN aLVAREZ

Fierro Martin – Erick Moreira

# Planteamiento del Problema

Implementar nuestra propia función **mi\_pburg** para determinar los coeficientes de la regresión y con estos determinar el espectro de potencias. Debería obtener el mismo resultado que con la función de Matlab **pburg**.

# Resolucion Teorica

Metodos Parametricos

Los métodos paramétricos permiten una estimación espectral de alta resolución, y se han mostrado como una alternativa interesante a la FT en el análisis de bioseñales, como, por ejemplo, en el caso de series temporales RR [1].

COEFICIENTES DE MODElo

La estimación de los coeficientes (parámetros) del modelo (autorregresivo) se realiza a partir de las muestras de la señal. Como regla general, si tenemos N muestras de la señal, suele utilizarse la mitad para estimar los coeficientes y el resto para validar el modelo.

Se han desarrollado diversos algoritmos para obtener los coeficientes del modelo. Los más utilizados son: a) Forward-backward; b) Least-squares; c) Yule-Walker; **d) Burg**; e) Geometric lattice. Los dos últimos utilizan la estructura AR lattice (representación como filtro de celosía de la función de transferencia del modelo AR) para obtener los coeficientes.

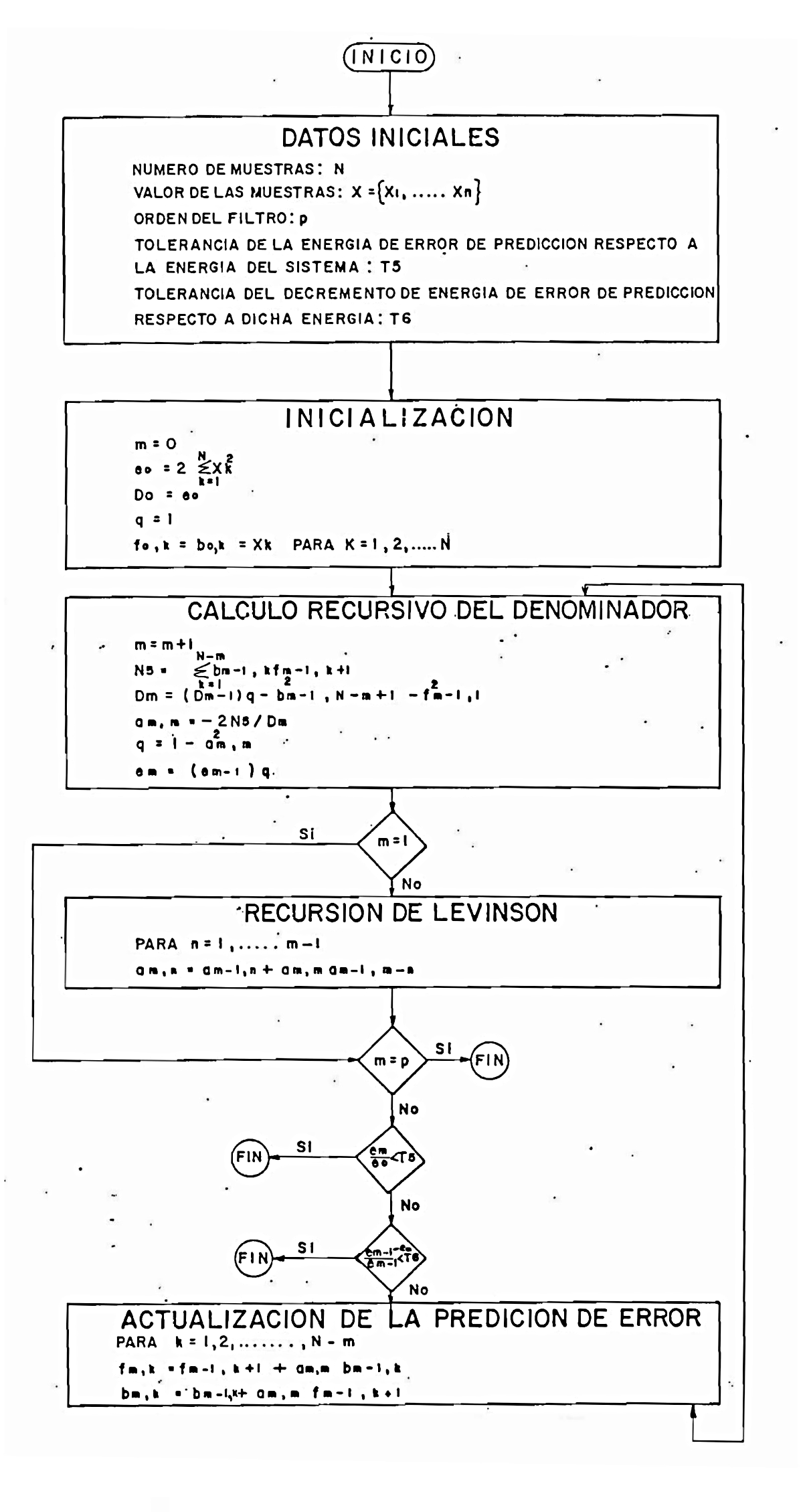
Metodo de burg

Algoritmo

Fue planteado en 1968. Es un algoritmo eficiente que ofrece importantes resultados al analizar sinusoides que contienen ruido aditivo.

Burg planteo que los parámetros AR deben ser tales que minimicen la suma de energías de error de predicción en adelanto y retardo.

A continuación, se muestra un diagrama de flujo que se empleara para la creación de la función de Matlab:

FIG. 1 Diagrama de Flujo de el algoritmo de Burg [2].

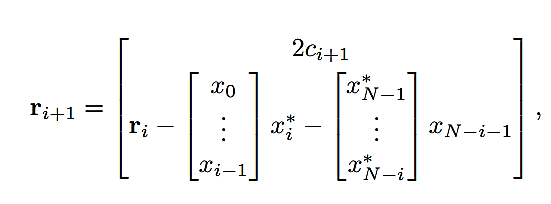
Algoritmo Matemático Pseudocódigo:

1. Inicialización:

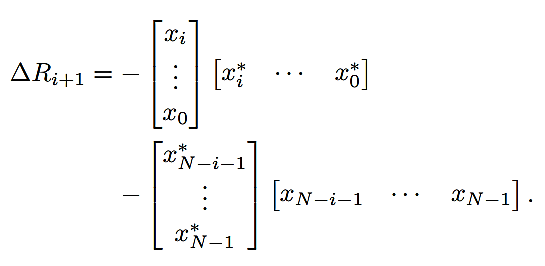
1. Computación de la reflexión de del coeficiente:
2. Actualización de los coeficientes de predicción:
3. Incremento del contador:
4. If

Endif

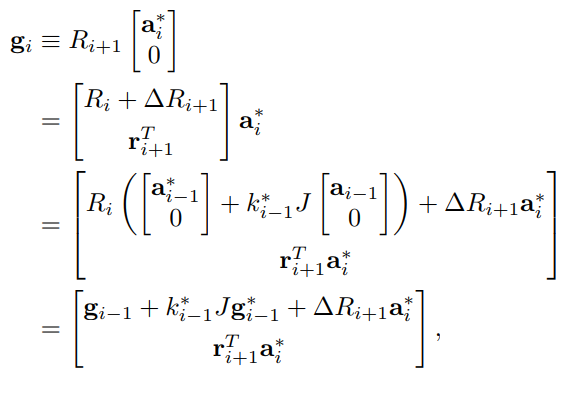
1. Actualizar



1. Computarizar



1. Actualizar



1. Repetir desde 1. [3]

# Resolucion PRÁCTICA

Código de la función en Matlab [4]:

function varargout= mi\_pburg( x, orden, nfft, ventana, Fs)

%PBURGW Estimación Espectral de Potencia usando Burg enventanando la señal

% Pxx = PBURGW(x,Orden,nfft,Ventana) es la estimación de la D.E.P.(Densidad

% Espectral de Potencia) de la señal x, usando el método de Burg con

% enventanado. Orden es el orden del modelo Auto Regresivo de predicción lineal.

% NFFT es la longitud de la FFT que determina las frecuencias a que se

% estima la D.E.P. Ventana es el nombre de la función Matlab que realiza

% una determinada ventana, se introduce en comillas simples como texto Ej.: 'blackmanharris'.

% Pxx tendrá longitud (NFFT/2+1) para NFFT par, (NFFT+1)/2 para NFFT impar,

% y NFFT si x es una señal compleja. NFFT por defecto vale 256.

% Si no se especifica la ventana, se utiliza por defecto la ventana

% rectangular(rectwin).

%

% [Pxx,F] = PBURGW(x,Orden,nfft,Ventana,Fs) devuelve además, el vector de

% frecuencias en que se evalúa en la D.E.P., para una frecuencia de muestreo Fs.

% Por defecto usa una Fs de 2 [Hz].

%

% PBURGW sin argumentos de salida grafica la Densidad Espectral de Potencias.

%1 - Revisión de los parámetros de entrada

narginchk(2,5);

if isempty(orden)

error('Necesita indicar el orden del modelo.');% Se tiene que especificar el orden del modelo

end

if nargin < 5, Fs = []; end

if nargin < 4, ventana='rectwin'; end % Ventana rectangular

if nargin < 3, nfft = []; end

if isempty(nfft), nfft = 256; end

if isempty(Fs), Fs = 2; end

%2 - Inicialización de los parámetros

N = length(x);

fi\_t = x(2:end); % fragmento de vector de errores de predicción hacia adelante

gi\_t = x(1:end-1); % fragmento de vector de errores de predicción hacia atrás

a = 1; % Inicialización del vector de estimadores de los coeficientes (a\_k) del filtro Auto Regresivo

Epsilon = x\*x'/N; % (Epsilon\_0)(Error de mín. cuadrados)

K = zeros(orden,1); % Estima de los coeficientes de reflexión

%3 - Cálculo de estimadores de K, E.

%-----Algoritmo Iterativo/

for i=1:orden

% Cálculo de la ventana:

% Se puede usar cualquier ventana que este implementada como una función Matlab

% Pero se recomienda utilizar la por defecto:

vent = ['v=' ventana '(length(fi\_t));'];

eval(vent);

v = v(:)'; % Conversion a vector fila(tanto si era columna como fila)

K(i)=-((v.\*fi\_t)\*gi\_t')/(((v.\*fi\_t) \* fi\_t' + (v.\*gi\_t) \* gi\_t')/2);

a = [a;0] + K(i) \* [0;flipud(conj(a))]; % Actualización de los {ak}

fi\_t\_nuevo = fi\_t + K(i) \* gi\_t;

gi\_t\_nuevo = K(i)' \* fi\_t + gi\_t;

fi\_t = fi\_t\_nuevo(2:end);

gi\_t = gi\_t\_nuevo(1:end-1);

Epsilon(i+1) = (1-K(i)\*K(i)') \* Epsilon(i); % Error de mín. cuadrados

end

% Fin del algoritmo Iterativo.

%4 - Estimación de la DEP usando la FFT(periodograma)

Af = abs( fft( a, nfft ) ) .^ 2; % Función de trasferencia del denominador del filtro

Pxx = Epsilon(end) ./ Af; % Función de trasferencia teniendo en cuenta la entrada(ruido)

%5 - Control de la salida

% Si la entrada es real, se selecciona sólo la primera mitad

if isreal(x)

select = (1:floor(nfft/2)+1)'; %Siempre es unilateral(Solo frecuencias positivas)

Pxx = Pxx(select);

%Para que sea unilateral hay que hacer una corrección

Pxx(1)=Pxx(1)/2;

Pxx(end)=Pxx(end)/2;

else

select = (1:nfft)';

Pxx = Pxx(select);

end

ff = (select - 1)\*Fs/nfft;

%Argumentos de salida:

if nargout == 0

newplot;

plot(ff,10\*log10(Pxx)); grid on;

xlabel('Frecuencia [Hz]');

ylabel('Módulo de la Densidad Espectral de Potencia [dB]');

title('Estimación Espectral de Burg');

end

if nargout >= 1

varargout{1} = Pxx;

end

if nargout >= 2

varargout{2} = ff;

end [5]

# Resultados y conclusiones

# Referencias

|  |  |
| --- | --- |
| [1] | J. Guerrero Martínez, «Tema 5: Estimación Espectral,» 2010. [En línea]. Available: http://ocw.uv.es/ingenieria-y-arquitectura/1-5/ib\_material/IB\_T5\_OCW.pdf. [Último acceso: 21 Julio 2019]. |
| [2] | M. Bayas Paredes, «MODELOS AUTORREGRESIVOS DE ANÁLISIS ESPECTRAL,» Diciembre 1984. [En línea]. Available: https://bibdigital.epn.edu.ec/bitstream/15000/5721/1/T607.pdf. [Último acceso: 21 Julio 2019]. |
| [3] | K. Vos, «A Fast Implementation of Burg’s Method,» Agosto 2013. [En línea]. Available: https://www.opus-codec.org/docs/vos\_fastburg.pdf. [Último acceso: 24 Julio 2019]. |
| [4] | J. G. Proakis y D. G. Manolakis, Tratamiento Digital de Señales, principios, algoritmos y aplicaciones, Mexico: Prentice-Hall, 1997. |
| [5] | J. de la Torre Peláez, «Windowed Burg algorithms,» 6 Marzo 2002. [En línea]. Available: https://www.mathworks.com/matlabcentral/fileexchange/1565-windowed-burg-algorithms. [Último acceso: 24 Julio 2019]. |