

# Obtención eficiente de espectros basada en un algoritmo mejorado de ventana ajustable

R. Núñez

*Electrónica y Telecomunicaciones-CICESE,  
Km. 107, carr. Tj.-Eda., Eda., B.C., México,  
Tel/Fax (646)175-0500/554,  
e-mail: rnunez@cicese.mx*

Recibido el 1 de septiembre de 2005; aceptado el 5 de octubre de 2006

Se presenta un algoritmo mejorado para producir espectros más precisos que los de la transformada rápida de Fourier (TRF) cuando utiliza ventanas senoidales, puesto que éste disminuye la fuga espectral de señales que lo necesitan y afecta mínimamente a las que no presentan este problema; situación completamente diferente con la utilización de las ventanas mencionadas, puesto que éstas afectan a todas las armónicas sin distinción. El algoritmo citado se basa en calcular la frecuencia fundamental de la señal adquirida y en modificar la frecuencia de muestreo por medio de técnicas de diezmado e interpolado; repitiendo este procedimiento para cada armónica cuya magnitud rebasa un umbral específico. La idea es ajustar el ancho de la ventana rectangular para que pueda contener un número entero de periodos de las señales por analizar. Para el desarrollo del algoritmo mejorado, se parte de un algoritmo de referencia (*AR*) realizado por Hidalgo, *et al.* (2002), en lenguaje C. El mismo, se programa y comprueba en lenguaje MATLAB6.5 y se observan algunas deficiencias al calcular el espectro de señales con armónicas múltiples. Por lo que, en este trabajo, se incorpora el procedimiento necesario para abatir estas deficiencias y así surge el llamado algoritmo mejorado (*AM*), el cual muestra, con base en las pruebas realizadas, ser más eficiente que el *AR*, pese a que reviste de un poco más de operaciones aritméticas. Además, el *AM* se valida con el programa comercial Dadisp de DSP, Co. y se adecúa para ayudar tanto en la localización de frecuencias enmascaradas como en la identificación de señales que presentarán fuga espectral.

*Descriptores:* TDF/TRF; fuga espectral; ventanas senoidales; diezmado; interpolado.

The improved algorithm can produce a more accurate spectrum than the sines windows and the FFT, because reduces the spectral leakage only of signals that need it; very different situation with the sines windows and the FFT utilization. The cited algorithm is based on acquired signal fundamental frequency calculation and right adjusting of the sampling frequency by software via decimation and interpolation techniques, in order to obtain a rectangular window wide with a complete number of signal periods. For to develop the improved algorithm, first is taken the algorithm of Hidalgo, *et al.* (2002) like reference, which was developed in the C language, then is developed again in MATLAB6.5 language and his spectral results showed to be deficient for the polyharmonics signals case. Therefore, in the present paper, we do the procedures to solve these deficiency problems. Therefore, the new algorithm is called improved algorithm (*IA*), which shows, in base to several achieved tests, that is more efficiently than the reference one. Finally, the mixed signals spectrum of the *IA* was validated by the Dadisp software of DSP, Co., and the algorithm could be used to look for masking frequencies and signals identification with leakage spectrum.

*Keywords:* DFT/FFT; spectral leakage; weighting windows; decimation; interpolation.

PACS: 07.05.Kf; 02.70.Hm; 33.20.Eg

## 1. Introducción

Como se sabe, el traslape y la fuga espectral se presentan en el espectro calculado por la transformada rápida de Fourier (TRF) [1-5], si la frecuencia de muestreo y el truncamiento de la señal adquirida no fueron realizados con cuidado [6,7]. El primero se produce cuando la frecuencia de muestreo es menor al doble de la frecuencia máxima de la señal adquirida y se reduce incrementando la frecuencia de muestreo o disminuyendo la frecuencia máxima de la señal. La segunda se produce por el truncamiento del periodo de la señal adquirida a un número finito de muestras, *i.e.*, por la discontinuidad entre la primera y última de éstas; se reduce multiplicando la señal muestreada por una ventana senoidal [6-9] que disminuye la discontinuidad mencionada a un valor común o muy cercano a cero. Se persigue obtener con el abatimiento de la fuga una mejor aproximación de la TRF y así un espectro más preciso. En la realidad, la fuga espectral es inevitable cuando se utiliza la TRF. Por lo que la idea fundamental del algorit-

mo propuesto en la Ref. 10, y el cual se utiliza de referencia, es disminuir la fuga espectral ajustando la duración de una ventana rectangular para que contenga un número entero de periodos de las señales adquiridas, en base a calcular la frecuencia fundamental de la señal y modificar la frecuencia de muestreo por medio de diezmado e interpolado. Dicho algoritmo, llamado de referencia (*AR*), fue desarrollado por Hidalgo, *et al.* (2002), en lenguaje C y probado con armónicas individuales y múltiples. En este estudio, el mismo algoritmo se programa en MATLAB6.5 y se constata que, para el caso de armónicas múltiples, presenta algunas deficiencias. Por tal razón este trabajo trata de subsanar éstas para que así puedan operarse señales múltiples con fuga y dejar intactas aquéllas sin este problema. En la Sec. 2 se describe lo básico sobre la fuga espectral y su abatimiento típico, y en la 3 se presenta la realización del *AR* en MATLAB6.5 y se muestra su operación deficiente para el caso de las armónicas múltiples. En la Sec. 4 se incorpora el procedimiento de conversión del mismo en el llamado algoritmo mejorado (*AM*) y se presen-

tan sus espectros resultantes; se observa que la resolución es bastante aceptable. Ya en la Sec. 5 se comparan los espectros resultantes de los algoritmos *AR* y *AM*, y también los de los provenientes de las ventanas típicas para un grupo de senoidales múltiples mezcladas, *i.e.*, con y sin fuga espectral, y, en base a esta comparación, se constata el potencial del *AM* (*v.g.*, se conocen las resoluciones y las alteraciones que impone a las señales coherentes, etc.). Además, en esta sección, se proponen algunas aplicaciones para el algoritmo propuesto, como localizador de señales enmascaradas y con posible fuga. En la Sec. 6, se concluye que el *AM* disminuye notablemente la fuga espectral de señales múltiples que así lo necesitan y afecta mínimamente a las que no lo requieren, situación completamente diferente con las ventanas senoidales, todo esto lo realiza en aras de un cálculo un poco mayor al del *AR*. En las Secs. 7 y 8 se presentan los agradecimientos y las referencias bibliográficas, respectivamente. Finalmente, en el Apéndice, se enlista el programa del estudio comparativo entre los algoritmos *AR* y *AM*.

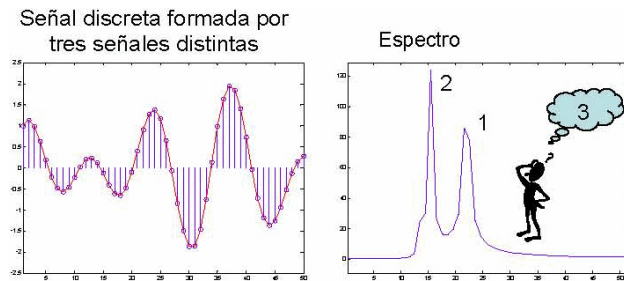


FIGURA 1. La fuga espectral puede ocultar información valiosa (*v.g.*, la  $f = 13\text{Hz}$  no se ve).

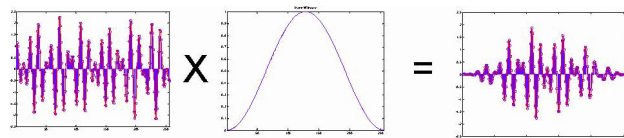


FIGURA 2. Aplicación de las ventanas senoidales (Hanning) para abatir la fuga espectral.

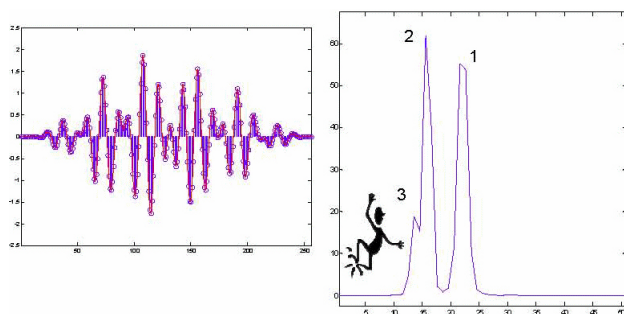


FIGURA 3. Abatiendo la fuga espectral se descubre la información oculta (*v.g.*, la  $f = 13\text{Hz}$  ya se ve).

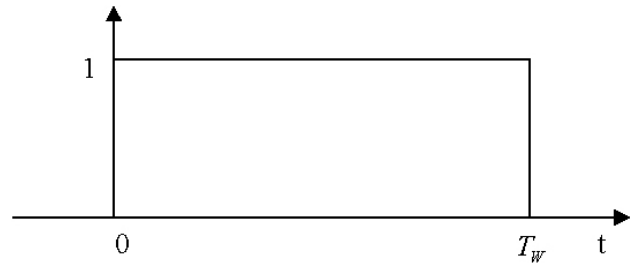


FIGURA 4. Ventana rectangular  $w(t)$  con duración  $T_w$ .

## 2. Fundamentos de la fuga espectral

Se sabe de las Refs. 6 to 9, que la fuga espectral se produce por el truncamiento necesario para convertir la señal adquirida en una secuencia de muestras de longitud finita, como se muestra en la Fig. 1. Dicha operación crea una discontinuidad entre la muestra inicial y la final de la secuencia debido a que los periodos de las señales generalmente resultan recortados. El procedimiento más común para disminuir a un nivel aceptable la citada fuga consiste en multiplicar la señal adquirida por una ventana senoidal, la cual hace converger suavemente la discontinuidad a un valor de amplitud común, tanto al inicio como al final de la secuencia. En la mayoría de los casos, este valor, se escoge muy cercano a cero (*c.f.*, con la ventana 3, o derecha, de la Fig. 2). Es claro que si la señal no presenta discontinuidad alguna, se trata de una señal coherente que no requiere de la aplicación de ninguna ventana senoidal; si ésta se realiza, la señal se distorsiona y su espectro registrará fuga.

En la Fig. 3 se observa cómo el uso de una de estas ventanas (*v.g.*, la Hanning) abate la fuga espectral y permite identificar fácilmente a la señal oculta por la falda de la propia fuga. Por ejemplo, en la Fig. 1, se tiene el caso cuando se suman las señales senoidales de las frecuencias de 13, 16 y 22.5Hz y, producto de la fuga espectral de la última, la primera, - dado que su magnitud es 4 veces menor-, pasa desapercibida en el espectro. Sin embargo, en la Fig. 3, se ve cómo la aplicación de la ventana senoidal ayuda en la detección de esta señal oculta de 13Hz. En el dominio del tiempo, las principales características de las ventanas senoidales [8,9] son: el número de términos de la función, el tipo de desplante inicial y final, la forma de acometida al centro y el área bajo la curva, y éstas están muy relacionadas con las del dominio de la frecuencia que son: la resolución, la atenuación del lóbulo principal y lateral, y la tasa de decaimiento, respectivamente. Por ejemplo, cualquier otra ventana que no sea la rectangular presentará una resolución mayor a dos bins, o armónicas, de frecuencia.

## 3. Realización del algoritmo de referencia (AR) e identificación de su deficiencia

La principal idea del *AR* [10-12] es evaluar la frecuencia fundamental de la señal de entrada y modificar la frecuencia de

muestreo por medio de programación utilizando técnicas de diezmado e interpolado, con el objeto de obtener una ventana rectangular con la medida adecuada.

Sea  $w(t)$  una ventana rectangular con ancho  $T_W$  (c.f., con la Fig. 4), el algoritmo de referencia consiste en cambiar el ancho de ésta, para que pueda contener un número entero de periodos de la señal.

La ventana está definida de la siguiente manera:

$$w(t) = \begin{cases} 1 & \text{para } 0 \leq t < T_W, \\ 0 & \text{para otro caso.} \end{cases}$$

Sea  $s(t)$  una señal senoidal con periodo desconocido  $T_0$  y un periodo de muestreo  $T_S$ .

El número de muestras  $N$  de la señal esta dado por la relación  $T_W/T_S$ , es decir:  $N = T_W/T_S$ .

La señal registrará fuga espectral excepto cuando:  $T_W = mT_0$ , y la fuga máxima se presentará cuando:  $T_W = mT_0 + T_0/2$ , en las gráficas de las Figs. 5 y 6 se ilustran estos conceptos.

En la Fig. 5, se muestra una señal senoidal de amplitud unitaria, con un periodo  $T_0 = 0.5$  s, y el ancho de la ventana es:  $T_W = 1$  s,  $m = T_W/T_0 = 2$ ; es decir, dentro de la ventana existen 2 periodos enteros. En este caso no existe discontinuidad, por lo tanto esta señal es coherente y su espectro no presentará fuga espectral. Sin embargo esto no ocurre en la práctica, debido a que no se conoce el periodo  $T_0$  de la señal adquirida y que el número de muestras de la trama generalmente es múltiplo de 2 (*i.e.*, debido al requisito impuesto para la obtención eficiente de la TRF).

En la gráfica de la Fig. 6 se muestra el peor de los casos. El cual se registra cuando se tienen 2.5 periodos de la señal dentro de la ventana, es decir  $m = T_W/T_0 = 2.5$ , donde  $T_W = 1.25$  s y  $T_0 = 0.5$ s; bajo estas condiciones se presenta la discontinuidad máxima, por lo tanto la señal es incoherente y su espectro presentará fuga máxima.

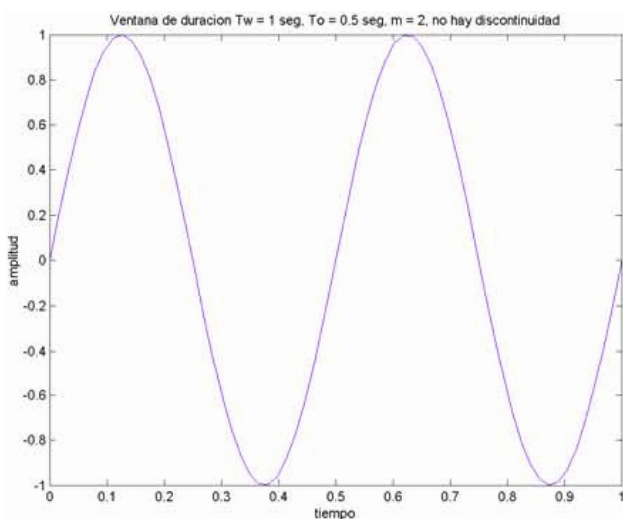


FIGURA 5. Señal senoidal  $s(t)$  coherente ( $m = 2$ ) o sin fuga espectral.

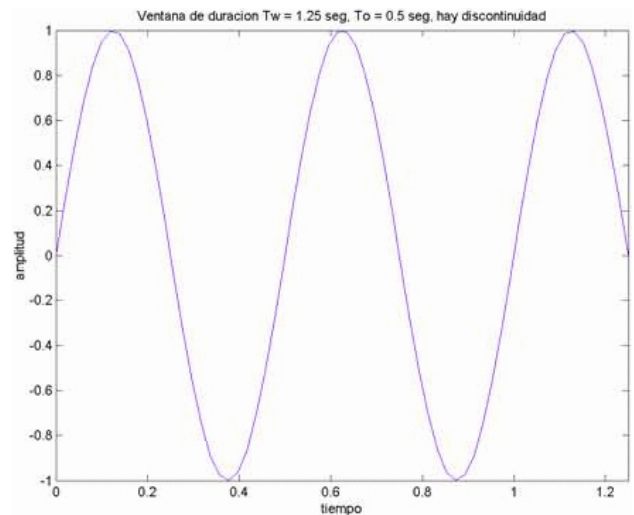


FIGURA 6. Señal senoidal  $s(t)$  no coherente ( $m = 2.5$ ) o con fuga espectral máxima.

El mencionado algoritmo modifica el ancho de la ventana hasta conseguir que un número entero de periodos de la señal se encuentren dentro de la ventana. La frecuencia de muestreo  $f_S$  está relacionada con el ancho de la ventana por medio de la fórmula:  $f_S = N/T_W$ . Otra interpretación de esto tiene que ver con que la  $f_S$  se modifica a tal grado que, con el mismo número de muestras, se obtenga un número entero de periodos de la señal por operar. Por lo que, el AR consiste en los siguientes pasos:

- a. Sea  $A_1$  la amplitud del bin más alto del espectro discreto y  $k_1$  (*i.e.*, bin 2 en la figura 7) la posición que ocupa en el espectro.
- b. Sea  $k_2 = k_1 \pm 1$  bin, cualquiera que presente la amplitud mayor para igualarlo a  $A_2$  (*i.e.*, bin 3 en la Fig. 7).

En la Fig. 7, se muestra una señal, -con casi discontinuidad máxima-, y su espectro para ilustrar la selección de  $k_1, k_2, A_1$  y  $A_2$ .

- c. Suponga que  $f_S$  es lo suficientemente grande para asegurar que no existirá traslape espectral. Entonces resulta que:

$$A_1 \text{sinc}(k_1 - C) + \text{sinc}(k_1 + C) = \frac{\cos(k_1 \cdot \pi)}{\pi} \frac{2C}{(C^2 - k_1^2)} \quad (1)$$

$$A_2 \text{sinc}(k_2 - C) + \text{sinc}(k_2 + C) = \frac{\cos(k_2 \cdot \pi)}{\pi} \frac{2C}{(C^2 - k_2^2)} \quad (2)$$

donde  $C = T_W/T_0$ , dividiendo (1) por (2) obtenemos

$$C^2 = \frac{A_1 k_1^2 + A_2 k_2^2}{A_1 + A_2}, \quad (3)$$

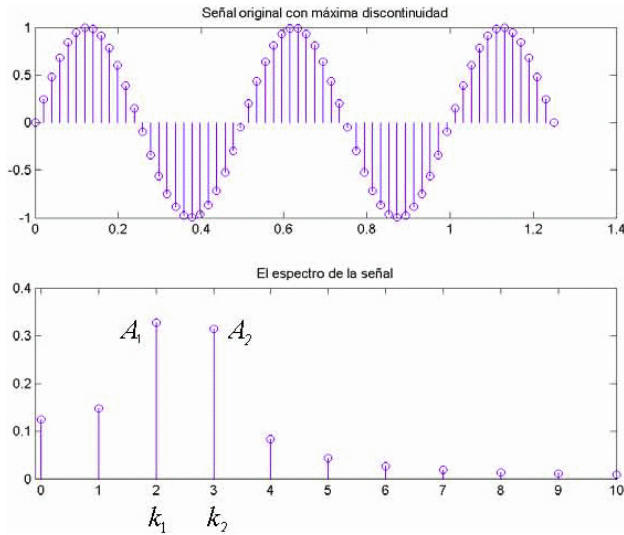


FIGURA 7. Señal senoidal con fuga espectral máxima ( $m=2.5$ ); ambos dominios.

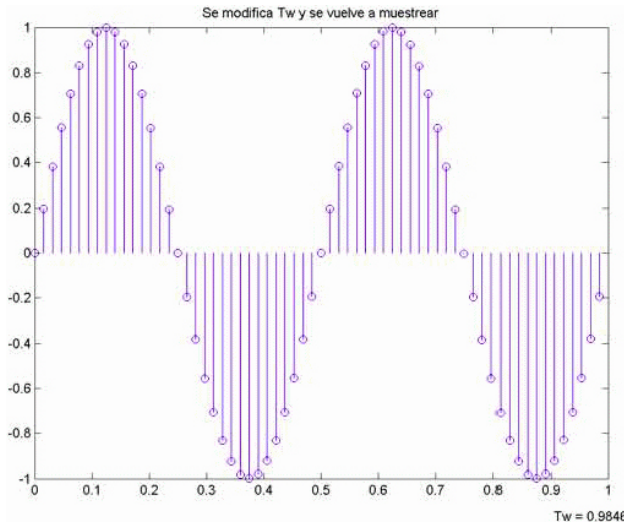


FIGURA 8. Señal senoidal modificada y ajustada a un número entero de periodos ( $T_w'$ ).

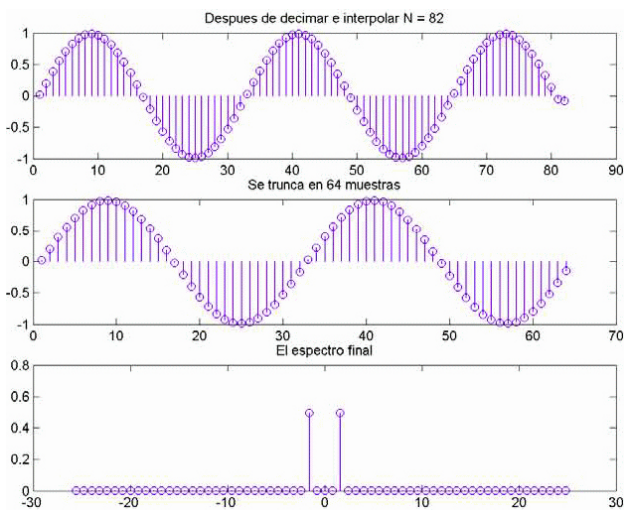


FIGURA 9. Señales senoidales, original y modificada (gráfica del medio), y espectro de la modificada.

- d. Se resuelve (3) para  $T_0$  y se define un nuevo ancho de la ventana  $T'_W$  que es exactamente un múltiplo  $m$  de  $T_0$ :

$$T'_W = mT_0 = \text{int}(C) \frac{T_W}{C}, \quad (4)$$

donde  $\text{int}(C)$ , representa la parte entera de  $C$ , según el mando  $\text{floor}(C)$  del MATLAB6.5.

En la Fig. 8, se presenta la señal obtenida al muestrear durante un tiempo  $T'_W = 0.9846$  s y se observa que la discontinuidad entre la primera y última muestra ha desaparecido. Este nuevo ancho de la ventana  $T'_w$ , también se puede obtener muestreando de nuevo la señal a un número de muestras, o con una frecuencia de muestreo, diferente. Pero en la práctica es difícil realizar estos cambios, ya que el número de muestras está limitado a múltiplos de 2, -para cualquier algoritmo eficiente de la TRF-, y el periodo de muestro se debe modificar por circuiteria. Consecuentemente, no es sencillo estar variando continuamente a la  $f_s$  o a  $N$ .

Para obtener  $T'_W$ , se modifica la frecuencia de muestreo por programación a un valor nuevo dado por

$$f'_S = \frac{N}{T'_W}. \quad (5)$$

Las alteraciones al periodo de muestreo son realizadas por un interpolador y un diezmador [4,5].

- e. Sea  $L - 1$  el número de muestras insertadas por el interpolador entre dos muestras de la secuencia original. Sea  $M - 1$  el número de muestras removidas por el diezmador entre muestras de la secuencia original. Entonces:

$$f'_S = \frac{L}{M} f_S, \quad (6)$$

de (5) y (6),

$$\frac{L}{M} = \frac{C}{\text{int}(C)}, \quad (7)$$

Adicionalmente,  $L$  y  $M$  deben ser números enteros ambas. Se puede fijar el valor de  $M$  primero y calcular  $L$  después. Mientras más grandes sean  $M$  y  $L$ , mayor será la aproximación de la Ec. (7) y la fuga espectral será menor. Cuando se cambia la frecuencia de muestreo por medio de interpolación o diezmado, el número de muestras cambia, pero las muestras en donde se encuentra la información útil deben ser siempre las  $N$  primeras; las sobrantes deben descartarse.

- f. El último paso es aplicar directamente la TRF a la nueva secuencia de muestras, como se representa en la Fig. 9. En la ventana 1, o superior, se muestra la señal obtenida después de interpolar y diezmar, y se observa como el número de muestras se ha incrementado (v.g.,  $N=82$  muestras), así como en la ventana 2 se presenta la señal que resulta después de eliminar las muestras sobrantes (v.g.,  $N=64$  muestras), y en la ventana



3, o inferior, se muestra el espectro final; observe que éste ya no registra la fuga espectral. Las gráficas de las Figs. 5, 8 y 9 representan señales reales muestreadas e interpoladas de 128 muestras, espaciadas a 1/128 de segundo, y con la presentación normal de sus espectros en 64 bins y a 1Hz de resolución. En la Fig. 10 y el Apéndice, se presenta el diagrama de flujo de la operación y el programa en MATLAB6.5 (c.f., con la primera parte) del AR, respectivamente.

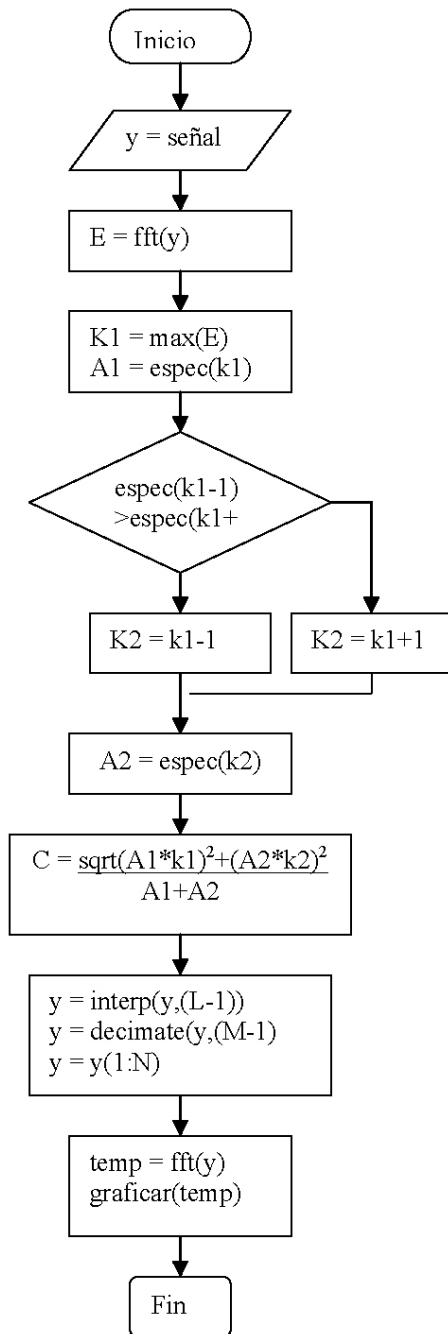


FIGURA 10. Diagrama de flujo del algoritmo de referencia (AR) realizado en MATLAB6.5.

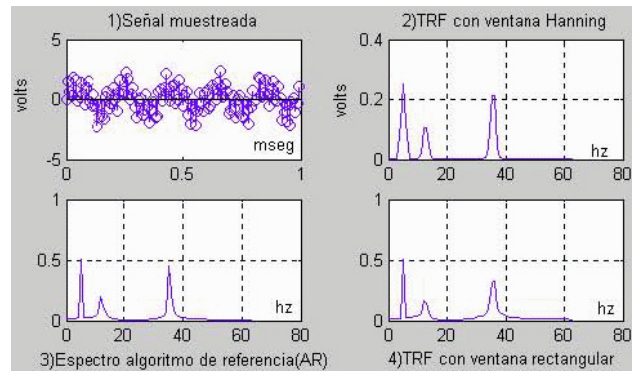


FIGURA 11. Resultados espectrales deficientes (ventana 3) del algoritmo de referencia para el caso de armónicas múltiples ( 5, 12.5, 35.5Hz); el AR procesa sólo a la armónica de mayor magnitud.

### 3.1. Identificación de la deficiencia del AR

Operar solamente sobre la frecuencia de mayor magnitud de la secuencia es la principal deficiencia del AR [10], como puede observarse en el espectro de la ventana 3 de la Fig. 11, el cual corresponde a tres señales de magnitudes 1, 0.5 y 1Vp y con frecuencias de 5, 12.5 y 35.5Hz, respectivamente. Fuera de la frecuencia de mayor magnitud (*i.e.*, de la de 5Hz), el algoritmo AR deja casi igual al resto de las armónicas; aunque los autores aseguran que trabaja correctamente para todas las armónicas. En la Sec. 5, se presenta un estudio comparativo entre los algoritmos AR y AM, bajo las mismas condiciones y basado en el programa que se enlista en el Apéndice.

### 4. Realización del algoritmo mejorado (AM)

En esta sección se realiza un procedimiento para mejorar el AR, es decir hacerlo que opere sobre todas las armónicas y de paso que no altere a las señales coherentes. Para realizar esto el algoritmo se aplica a cada componente de frecuencia cuya magnitud rebasa un umbral o nivel predeterminado, y así va seleccionando y salvaguardando estas armónicas de información, evitando así considerar las menores al umbral o ruidosas, como lo indica el cuadro: picos= E> nivel, en el diagrama a flujo de la Fig. 12. El salvaguardo de armónicas, se lleva acabo en un vector H= s\*N (c.f., con cuadro: Guardar datos y con los grupos de instrucciones **b)** y **c)** del enlistado de las Fig. 10 y el Apéndice (segunda parte), respectivamente), que se utiliza al final para construir y aproximar al espectro final, donde s corresponde al número de muestra. Para el caso particular, se realiza la búsqueda de los picos de la TRF y sólo se procesan los mayores al umbral mencionado, el cual puede fijarse según la tasa de señal a ruido que se considere. El AM utiliza un arreglo matricial para memorizar la magnitud de las armónicas seleccionadas y su posición relativa en la frecuencia, y también realiza los ajustes correspondientes para presentar la mejor aproximación de la señal en el tiempo y en la frecuencia (c.f., con los grupos de instrucciones **b)**-**e)** del enlistado en el Apéndice, segunda parte). Para mostrar su operación, se utilizan las mismas señales de la Fig. 11 y los

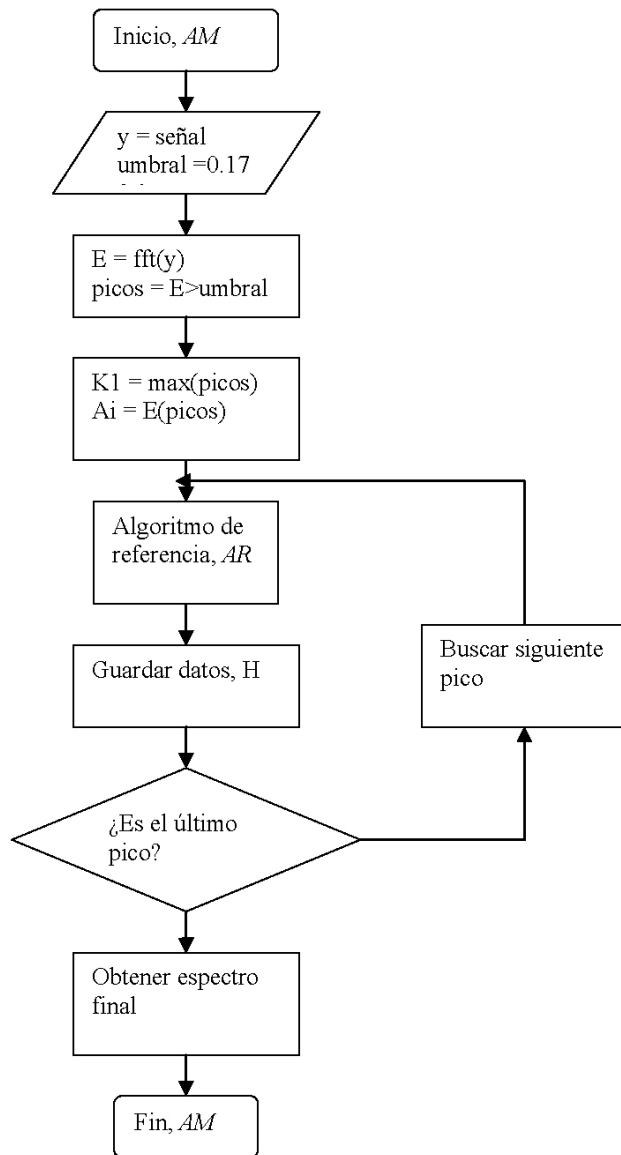


FIGURA 12. Diagrama de flujo del algoritmo mejorado (AM) realizado en MATLAB6.5.

resultados se presentan en la Fig. 13. De los espectros resultantes, se puede observar la mejora incorporada al AR; sobre todo al comparar los espectros de las ventanas 3 de las Figs. 11 y 13, respectivamente. Así como también, al comparar los espectros de las señales pesadas por las ventanas rectangular y de Hanning mostradas en las ventanas 2 y 4, respectivamente, con el de la señal ajustada mostrada en la ventana 3 de la Fig. 13. El AM reduce aún más la fuga espectral (c.f., las ventanas 2 y 3 de la Fig. 13) y altera en menor medida la magnitud de las señales sin fuga (c.f., las ventanas 4 y 3 de la Fig. 13), alteración que sí realizan las ventanas senoidales. De la ventana 1 a la 4, de la Fig. 14, se presentan las mismas señales senoidales de 5, 12.5 y 35.5Hz (v.g., la señal muestreada), su espectro original calculado por la TRF, el espectro producido por el AM y la señales senoidales originales y operadas por el AM, respectivamente. En esta última, las señales se encuentran encimadas y al compararlas se

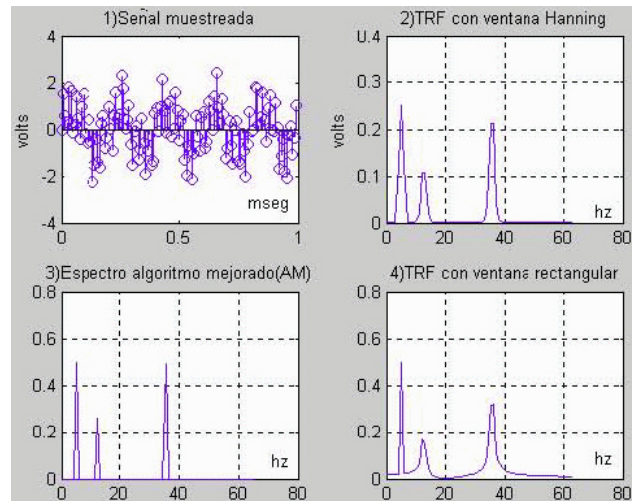


FIGURA 13. Resultados espectrales eficientes (ventana 3) del algoritmo mejorado para el caso de armónicas múltiples (5, 12.5 y 35.5Hz); el AM opera sobre todas las armónicas de información cuya magnitud es mayor al umbral seleccionado.

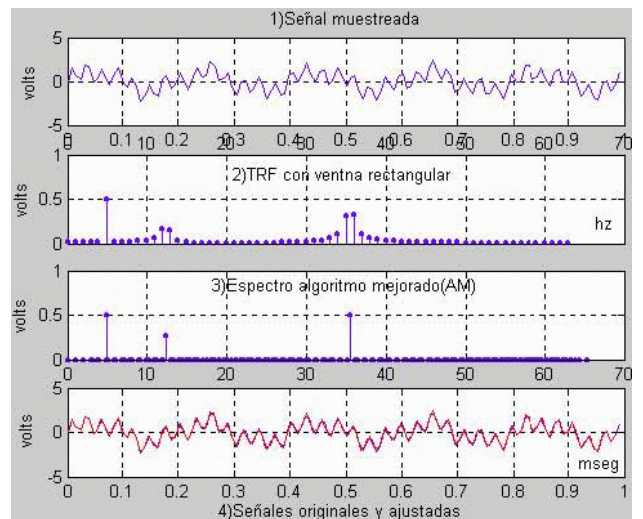


FIGURA 14. En la ventana 1, se presentan las señales senoidales sumadas de 5, 12.5 y 35.5Hz, y en la ventana 4, se despliegan en forma encimada las señales originales y las ajustadas. En las ventanas 2 y 3, se presentan los espectros de la TRF y del AM, respectivamente.

observa la eliminación de las muestras que realiza el algoritmo implementado. En la ventana 3, -de la misma figura-, se presentan las armónicas ajustadas y como se aproximan a sus valores verdaderos. En la Fig. 15 se aplica una señal que es la suma de varias senoidales de magnitud unitaria y con frecuencias de 10.5, 20.5, 30, 40.5, 50.5 y 60Hz, y se considera un umbral de 0.17V de c.d. como nivel de selección. De la ventana 1 a la 4 se despliegan: las señales senoidales originales, y los espectros de las señales operadas con: la ventana de Hanning, el AM y la ventana rectangular, respectivamente. Al comparar el espectro resultante del AM con los de las ventanas de Hanning y rectangular podemos constatar el rendimiento del primero.

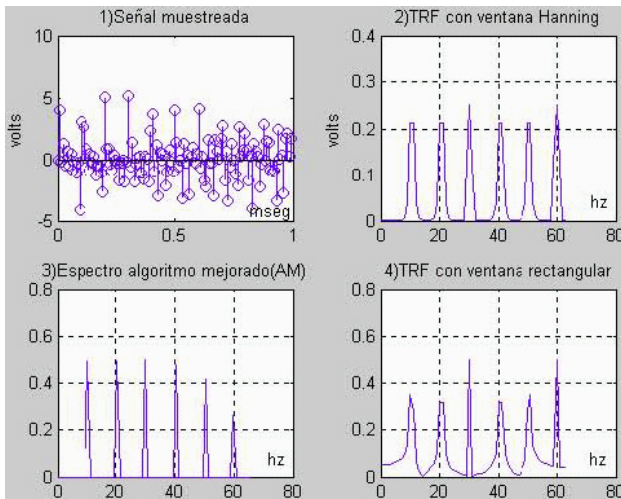


FIGURA 15. Espectro eficiente (ventana 3) del AM, para el caso de armónicas múltiples de 10.5, 20.5, 30, 40.5, 50.5 y 60Hz.

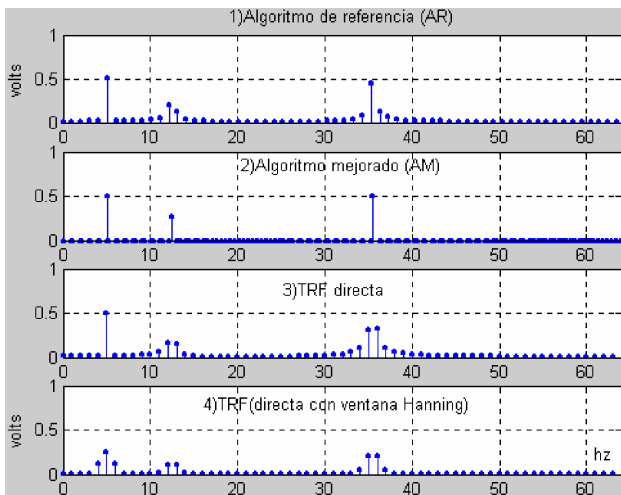


FIGURA 16. Comparando los resultados espectrales entre: el AR, el AM, la TRF(directa) y la TRF(con ventana Hanning) presentados desde la ventana 1 a la 4, respectivamente, para las señales sumadas de 5, 12.5 y 35.5Hz.

### 5. Comparaciones de los algoritmos y otras aplicaciones

El programa en MATLAB6.5 en el Apéndice nos ayuda en la realización de un estudio comparativo entre los espectros de los algoritmos AR y AM, y cuyos resultados se muestra en la Fig. 16. En dicha figura, se presenta el caso ya estudiado de las señales mezcladas de 5, 12.5 y 35.5Hz. A partir de la ventana 1 a la 4 se muestran los espectros del AR, del AM, el directo y el con ventana Hanning, respectivamente. En la ventana 2, se contempla el resultado del AM como el más efi-

ciente, aunque para obtenerse se requiere de un número operaciones aritméticas equivalente al de las armónicas que rebasan el umbral mencionado. Estos resultados fueron validados con el programa profesional de análisis digital de señales Dadis [13].

Otras aplicaciones recomendadas para el AM son: **a)** ayudar en la localización de frecuencias enmascaradas, ya que al incrementar la resolución del espectro las señales ficticias se mueven de posición (*i.e.*, cambian de bin) y este incremento se realiza cambiando el periodo de muestreo  $T_s$  o disminuyendo el ancho de la ventana rectangular  $T_w$ , y **b)** ayudar en la detección si la señal presentará fuga espectral, ya que incrementando la resolución este problema disminuye, es decir, la amplitud de la componente espectral correcta aumenta mientras la fuga disminuye.

### 6. Conclusiones

El algoritmo mejorado (AM) disminuye notablemente la fuga espectral de señales con armónicas múltiples que así lo necesitan y afecta mínimamente a las que no la requieren, situación completamente diferente con las ventanas senoidales, puesto que éstas afectan a todas las armónicas sin distinción. El AM facilita la variación del ancho de la ventana rectangular de truncamiento y esto ayuda en la localización de frecuencias enmascaradas, ya que al incrementar la resolución las señales ficticias se mueven de posición o del bin dentro del espectro. Este incremento, se realiza cambiando el periodo de muestreo  $T_s$  o el ancho de la ventana rectangular  $T_w$ . Debido a la incorporación del nivel de umbral, el AM detecta sólo aquellas armónicas cuyas magnitudes rebasan éste y elimina así las ruidosas, también puede manifestar si la señal presentará fuga espectral, ya que incrementando la resolución la fuga varía, es decir, la amplitud de la componente espectral correcta aumenta mientras que la fuga disminuye. Es claro que hay que tener en cuenta que, bajo ciertas condiciones, la fuga espectral puede producir traslape espectral y viceversa. Finalmente, se observa que cuando se tienen señales mezcladas, *i.e.*, coherentes e incoherentes, el algoritmo propuesto produce espectros más precisos que los provenientes de las ventanas senoidales y su TRF, aunque para ello requiere de realizar un poco más de operaciones aritméticas que las correspondientes al AR. Los resultados fueron validados por el programa profesional Dadis de DSP, Co.

### Agradecimientos

Agradecemos al CONACYT por apoyar económicamente el presente, a través del proyecto: 7453, dirigido por el Dr. J. Alvarez G.

## Apéndice

Enlistados en MATLAB6.5 de algoritmos de referencia (AR) y mejorado (AM) de ventana ajustable (comparación de resultados espectrales).

### ALGORITMOS MEJORADO (AM) y DE REFERENCIA(AR) DE VENTANA AJUSTABLE

R. Núñez, DET-CICESE

```
% Comparación entre los algoritmos de referencia,el mejorado, el de la
% TRF(directa) y el de la TRF(con ventana Hanning).
% % Algoritmo de referencia (AR). *****
clear;
close all;
clc;
% Datos
nivel_normal=0;
nivel_mejorado=.05;
N=128; % %Numero de muestras
NC=128;
Tm=1/128; % periodo de muestreo
Tm1=Tm;
% Senal de prueba: aplicaciones (secciones 3 y 4) % *****
n=linspace(0,Tm*(N-1),N); ** instrucciones a)
y = 10*sin(2*pi*7*n)/7 +10*sin(2*pi*21*n)/21 +10*sin(2*pi*35*n)/35 + 10*sin(2*pi*49*n)/49;
espec=abs(fft(y))/N; *****
% Encontrar A1,A2,k1,k2 para C
k1 = find (espec==max(espec));
k1=k1(1);
A1=espec(k1);
if espec(k1-1)>espec(k1+1)
    k2=k1-1;
else
    k2 = k1+1;
end
A2=espec(k2);
k1=k1-1;
k2=k2-1;
C = sqrt( (A1*k1. 2 + A2*k2. 2) / (A1 + A2) );
M = 500; % Muestras interpoladas
L = floor( M * C / floor(C) );
Tm = M*Tm/L;
n=linspace(0,Tm*(N-1),N);
y1 = interp(y,(L-1));
y1 = decimate(y1,(M-1)); *
y1 = y1(1:N);
temp=abs(fft(y1,NC)/NC);
temp2=zeros(1,length(temp));
for l=1:1:length(temp)
    if temp(1,l)>nivel_normal
        temp2(1,l)=temp(1,l);
    end
end
% grafica la ventana con el algoritmo
figure(1)
subplot(4,1,1);
stem((1/(NC*Tm):1/(NC*Tm):(1/(2*Tm)))-1/(NC*Tm),temp2(1:NC/2),' ');
title(' Algoritmo de referencia (AR)');
```

**Algoritmo de referencia (AR)**



```

axis([0,length(espec)/2,0,1]);
grid
figure(2)
subplot(4,1,1);
plot((1/(NC*Tm):1/(NC*Tm):(1/(2*Tm)))-1/(NC*Tm),temp2(1:NC/2));
title(' Algoritmo de referencia (AR)');
axis([0,length(espec)/2,0,1]);
grid
% % Algoritmo mejorado (AM) % % *****
m=1; % Se ajusta la ventana a m por Tm * instrucciones a)
g=m; *****
% Senal; selección de armonicas basada en umbral ( seccion 3, proceso de mejora) % ***
n=linspace(0,Tm*(N-1),N);
tam=length(espec);
x1=0;
x2=0; * instrucciones b)
s1=1;
for s=2:tam
    if ((espec(s);x1)&(x1<x2))
        picos(s1)=s-1;
        s1=s1+1;
    end
    x2=x1; *
    x1=espec(s);
end
picos=picos(find(nivel_mejorado;espec(picos))); *
pic=length(picos);
picl=pic/2;
for s=1:picl
    Tm=Tm1; % Se ocupa el Tm original para encontrar el Tm de la ventana ajustable *****
    % Encontrar A1,A2,k1,k2 para C
    k1 = picos(s);
    A1=espec(k1);
    if espec(k1-1) < espec(k1+1) *
        k2=k1-1;
    else
        k2 = k1+1;
    end
    A2=espec(k2);
    k1=k1-1;
    k2=k2-1;
    C = sqrt( (A1*k1. 2 + A2*k2. 2) / (A1 + A2) ); * instrucciones c)
c(s)=C; % Vector C donde se colocan las frecuencias mas aproximadas de la senal
M = 500; % Muestras interpoladas
L=floor(M*C/floor(C));
Tm=M*Tm/L;
Tm=g*Tm; % g es un factor para que ajustar el periodo de cada una de las secuencias *
tm(s)=Tm; % tm es un vector periodo
h=interp(y,(L-1)); % Se realiza el interpolado a L-1
h=decimate(h,(M-1)); % Se realiza el diezmado a M-1
h=h(1:N);
H(s,:)=h; % Se guarda en cada secuencia dentro de un vector H de s X N *
w(s,:)=(1/(NC*Tm):1/(NC*Tm):(1/(2*Tm)))-1/(NC*Tm); % Vector de frecuencia
w(s,:)=w(s,:)*g;
tempo(s,:)=(fft(h,NC)/NC);
end

```

```

H1=zeros(picl,length(h));
for s=1:picl
    H1(s,:)=tempo(s,:).*(abs([1:length(h)]-(picos(s)-1))i=1);
    H1(s,:)=H1(s,:)+tempo(s,:).*(abs([1:length(h)]-(picos(pic-s+1)+1))i=1);
end
h1=zeros(picl,length(h));
for s=1:picl
    for z=2:tam
        if((abs(H1(s,z))<nivel_mejorado))
            h1(s,z)=H1(s,z);
        end
    end
end
end
*****
figure(1)** secciones 3 y 4 , figura 14
subplot(4,1,2);
stem(w(1,:),abs(h1(1,1:NC/2)),'.');
for s=2:picl
    hold on
    stem(w(s,:),abs(h1(s,1:NC/2)),'.');
    hold off
end
title('Algoritmo mejorado (AM)')
axis([0,length(espec)/2,0,1]);
grid
figure(2)
subplot(4,1,2);
plot(w(1,:),abs(h1(1,1:NC/2)));
for s=2:picl
    hold on
    plot(w(s,:),abs(h1(s,1:NC/2)));
    hold off
end
title('Algoritmo mejorado (AM)')
axis([0,length(espec)/2,0,1]);
grid
figure(1)
subplot(4,1,3)
stem([0:N/2-1],espec(1:NC/2),'.');
axis([0,length(espec)/2,0,1]);
title('TRF directa')
grid;
figure(2)
subplot(4,1,3)
plot([0:N/2-1],espec(1:NC/2));
axis([0,length(espec)/2,0,1]);
title('TRF directa')
grid;
ventana=hann(N);
ventana=ventana';
y2=ventana.*y;
especv=abs(fft(y2))/N;
figure(1)
subplot(4,1,4)
stem([0:N/2-1],especv(1:NC/2),'.');
axis([0,length(espec)/2,0,1]);
title('TRF(directa con ventana Hanning)')

```

Algoritmo mejorado (AM)

\*

\*

\*

\* instrucciones d) y e)

\*

\*

\*

\*

\*

```

grid;
figure(2)
subplot(4,1,4)
plot([0:N/2-1],especv(1:NC/2));
axis([0,length(espec)/2,0,1]);
title('TRF(directa con ventana Hanning)')
grid;
*****

```

- 
1. J.W. Cooley and J.W. Tukey, *Math. Computation* **19** (1965) 297.
  2. E.O. Brigham, *The Fast Fourier Transform, 2<sup>nd</sup>* (Englewood Cliffs, NJ: Prentice Hall, 1988).
  3. Digital Signal Processing Com., *Programs for Digital Signal Processing* (N.Y.: IEEE P., 1979).
  4. A.V. Oppenheim and R.W. Schaffer, *Discrete-time signal processing* (Pren.-Hall, 1989).
  5. J. Proakis and D. Manolakis, *D.S.P: Principles, Algorithms, and Applics. 2<sup>nd</sup>* (N.Y., 1992).
  6. R. Lyons, *Understanding digital signal processing 1<sup>rd</sup>*. (Ed. Addison-Webley, 1997).
  7. R. Núñez, *Aplicaciones en instrumentación del análisis digital de señales*(curso: **ET631**) y *Procesamiento digital de señales*(curso: **ET535**) (DET, CICESE, 2004).
  8. F. Harris, *PIEEE* **66** (1978) 51.
  9. A.H. Nuttall, *IEEE T. ASSP* **29**(1981).
  10. R.M. Hidalgo, J.G. Fernández, R.R. Rivera, and H.A Larrondo, *IEEE T. I&M* **51** (2002).
  11. Y. Neuvo, C.Y. Dong, and S.K. Mitra, *IEEE Trans. On Acoust. Speech and Signal Proc.* **32** (1984) 563.
  12. J. Schoukens, Y. Rolain, G. Simon, and R. Pintelon, *IEEE T. I&M* **52** (2003).
  13. Dsp, Inc., "The Dadisp **V7.0**, data analysis and display software", user manual, 2004.