



UNIVERSIDAD NACIONAL
AUTONOMA DE
MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
"ACATLAN"



PROGRAMA PARA RESOLVER
LA TRANSFORMADA RAPIDA
DE FOURIER

T E S I N A

Para obtener el título de:
Lic. en Matemáticas Aplicadas y Computación

PRESENTA:

José Raymundo Arteaga Luna

Asesor: M.I. Víctor J. Palencia Gómez

Acatlán, Estado de México 1996

**TESIS CON
FALLA DE ORIGEN**

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

DEDICATORIA

A una persona que fue muy importante en mi vida y que sin ella tal vez nunca hubiera acabado a tiempo la carrera a ella porque me enseñó mucho en cuanto a la vida y al estudio porque compartimos notas y estudios gracias Adi; a mi padre que me enseñó que más que un padre tengo un amigo incondicional, a mis hermanos que me enseñaron lo que es la amistad y lo llevo muy presente a cada momento; a mi esposa que me motivo mucho para seguir adelante sin importar las adversidades que uno tenga siempre hay que seguir, a dos niñas que tengo a mis dos mounstritas con las cuales he compartido muy gratos momentos; a mi hija por la cual trabajo cada vez más duro sin importar el cansancio o los problemas que surjan; y en especial a una persona que me ha ayudado toda la vida, que ha sido más que una amiga, el mejor amigo que he tenido en la vida y nunca podrá haber alguien igual, a ti mamá que eres lo que más he querido en la vida, esto es para ti.

AGRADECIMIENTO

Al Profesor Palencia, dada la carga de trabajo que tiene continuamente, me ayudo a realizar este trabajo, así como la paciencia que ha tenido hacia mi, a la UNAM por permitirme ser parte de ella y a Acatlan que fue mi casa por muchos años.

INTRODUCCION

INTRODUCCION

El Objetivo de este trabajo es elaborar un programa para computadora personal que permita realizar análisis de algunas señales mediante Filtros Digitales.

Al correr el tiempo se da uno cuenta que existen una inmensidad de fenómenos que observamos en la vida y no sabemos como suceden; toda la tecnología que existe la aplicamos día con día sin saber cual es su funcionamiento, existe una enorme variedad de aplicaciones en el mundo real, las cuales van desde la medicina, la computación, las comunicaciones, hasta llegar a los aparatos electrónicos que se usan en el hogar, la explotación de los recursos naturales; todo esto son cosas cotidianas para la mayoría de los hombres.

Algo muy diferente ocurre algunas veces en la educación, la enseñanza de métodos matemáticos y computacionales resultan algunas veces fastidiosos y difíciles, los cuales casi nunca llegamos a saber su aplicación, porque son muy abstractos, fomentan tal vez, de alguna forma que se pierda el interés en las matemáticas, tachándolas de tediosas y sin sentido, dando como resultado el temor de las mismas, lo cual la mayoría de los seres humanos lo experimentamos desde los primeros años de nuestra educación.

En el séptimo año de la carrera (MAC) nos ocurrió algo muy parecido, en Métodos Matemáticos para Sistemas Dinámicos, non empezó a parecer la clase muy teórica y formal, pero contrario a lo que muchos pensarían, el interés del grupo se mostraba día con día mayor, todo el mundo asistía a esa materia, el profesor imponía, pero con todo y eso, la clase era muy amena; al finalizar el curso muchos salimos con dudas en cuanto a la aplicación del curso, en donde íbamos a aplicar Las Series de Fourier, La Transformada de Laplace, etc.

Al buscar un tema para mi tesina, recordé aquel curso de Sistemas Dinámicos, platicando con mi asesor (Prof. Víctor Palencia) le hice mención de mi interés sobre la Transformada de Fourier, el profesor me recomendó leer sobre la Transformada Rápida de Fourier y sus aplicaciones, escogimos desarrollar un programa para la Transformada Rápida de Fourier usando Filtros Digitales, los cuales son una de las aplicaciones de la transformada.

Al leer sobre Filtros Digitales, me sorprendí al ver el campo de aplicaciones tan amplio que abarca : actualmente se utilizan los Filtros Digitales en los Electrocardiogramas, para aislar los sonidos de los demás musculos que se encuentran alrededor del corazón, con lo cual se logra mejores resultados en el estudio del corazón; también en la medicina, los ultrasonidos han sido de gran utilidad, aquí los filtros (circuitos integrados) aíslan la resonancia más tenue, lo cual da como resultado una mejor visión de los órganos; en la televisión y cine cuando se tienen problemas con la recepción de la señal, los técnicos utilizan filtros (aparatos electrónicos) con las cuales aíslan las ondas senoidales de las cosenoidales para tener una mejor visión de la imagen; en el campo de la meteorología, los científicos pueden realizar predicciones en cuanto a las condiciones del clima, utilizando filtros más específicos (software); donde yo trabajo es un Centro de Servicios en Hojalata nos dedicamos a cortar hojalata en rollo y cuadro a ciertas medidas dependiendo la necesidad del cliente, se cuenta con un aparato llamado detector de Calibres, lo que realiza el Detector de Calibres es separar las hojas que tienen un espesor diferente al cual se le indica, al comenzar a utilizarlo se le programa para que trabaje con un cierto espesor de hoja y se le da una tolerancia de ± 0.005 de pulgada, por ejemplo: se le programa para aceptar hoja con un espesor de 0.088" y el aparato va a dejar pasar hoja con espesor de 0.083" a 0.093" y las hojas que estén fuera de este rango las coloca en otro recipiente de hoja, este aparato es otro ejemplo de Filtros Digitales.

Lo que a continuación se muestra es un pequeño estudio de lo que son los Filtros Digitales y la Transformada Rápida de Fourier, esta tesis no es tal vez lo máximo en filtros y Transformada Rápida de Fourier, pero puede servir de introducción para aquellos que desean conocer más allá de estos conceptos. Una de las dificultades que se tiene en el estudio de estos temas es la falta de bibliografía en español, dado que la mayoría de los textos se encuentran en inglés lo cual podría ser un obstáculo para el estudio; el programa que se realizó, resuelve la Transformada Rápida de Fourier, este programa está hecho en Pascal, se que no es lo más usual en el medio, pero dado las carencias para adquirir otro software (Turbo C, C++, u otro) opte usar este, pero este programa se puede utilizar en Lenguaje C compilándolo nuevamente. Este tema puede ser muy abundante e interesante, falta de algún modo tener más información acerca de donde y como se puede adquirir este tipo de información, así como, quienes son los eruditos en estos temas aquí en el País, para poder platicar y que a su vez nos aconsejarán lo que se podría estudiar y realizar, todo encaminado para un mejor aprovechamiento de todos los conceptos y que se pueda contribuir en algo por bien de nuestra Universidad y porque no por el País.

INDICE

INDICE

1. Análisis de Señales	10
1.1.- Análisis de señales. 11	
1.2.- Muestreo. 12	
1.3.- Respuesta de un sistema en la Frecuencia, 13	
1.4.- Concepto de Filtro, 15	
1.5.- Tipos de Filtro, 16	
1.5.1.- Tipos de Filtro, 16	
1.5.2.- Características de los Filtros Digitales, 17	
2. Filtros Digitales	19
2.1.- Diseño de Filtros con Análisis de Fourier, 20	
2.2.- Filtro Pasa Bajas, 21	
2.3.- Filtro Pasa Altas, 22	
2.4.- Filtro Pasa Banda, 23	
2.5.- Fenómeno de Gibbs, 24	
3. Transformada Rápida de Fourier	26
3.1.- Transformada Discreta de Fourier, 27	
3.2.- Transformada Rápida, 32	
3.3.- Algoritmo del Método, 36	
3.3.1.- Algoritmo, 36	
3.3.2.- Representación Gráfica de una Señal, 36	
3.3.3.- Nodos Duales, 37	
3.3.4.- Diagrama de Flujo de la Transformada Rápida de Fourier, 39	
3.3.5.- Algoritmo de Decimación en el Tiempo, 42	
3.3.6.- Algoritmo de Decimación en Frecuencia, 47	
4. Programa de Computadora	51
4.1.- Programa de Computadora, 52	
4.2.- Ejemplo de Aplicación, 56	
Bibliografía	61

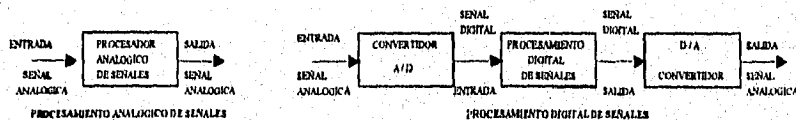
ANALISIS DE SEÑALES

I.- ANALISIS DE SEÑALES.

I.1.- Análisis de Señales.

Muchas de las señales que encontramos son analógicas por naturaleza, esto es, las señales son funciones de variable continua: en muchos casos, como el tiempo y espacio, usualmente toman un rango de valores continuo. Estas señales pueden ser procesadas directamente por sistemas analógicos apropiados tales como filtros, analizadores de frecuencia o multiplicadores de frecuencia, para la propuesta de cambio de sus características o extracción de algún ruido. En estos casos decimos que la señal ha sido procesada directamente y es una forma analógica; tanto la señal de entrada como la señal de salida son formas analógicas.

El Procesamiento Digital de Señales atañe a la sustitución de elementos analógicos, tales como amplitudes, moduladores y filtros; las ondas que deben ser procesadas son filtradas, después convertidas a representaciones digitales, para finalmente regresarlas a su forma original. El procesamiento digital de señales provee un método alternativo para el procesamiento de señales analógicas. El desarrollo de procesamientos digitales, es una necesidad para una interfase entre señales analógicas y procesadores digitales. Esta interfase es llamada **Convertidor Analógico Digital (A/D)**. La salida del convertidor es una señal digital que es apropiada para un Procesador Digital.



1.1.1.- Ventajas del Procesamiento Digital sobre el Analógico.

Un sistema Digital Programable permite flexibilidad en reconfigurar las operaciones del Procesamiento Digital de Señales simplificando y cambiando el programa; la reconfiguración en Sistemas Analógicos usualmente implica un rediseño del hardware, probando y verificando las señales apropiadas. El Procesamiento Digital provee un mejor control de las necesidades de precisión. Las tolerancias en los componentes de los circuitos analógicos los hacen extremadamente difíciles para el rediseño de sistemas y para controlar la exactitud de un sistema de Procesamiento Analógico de Señales. Un sistema digital provee mucho mejor control de los requerimientos de exactitud.

Un diseño analógico se deteriora con el tiempo y sólo es controlable en una base estática. Con un Sistema Digital Programable el diseñador construye un sistema analógico con cierta degradación lo cual provoca la pérdida de la información. La cuantificación es el proceso de representación de una muestra analógica por el nivel más cercano que corresponde exactamente a una escala integral.

1.2.- Muestreo

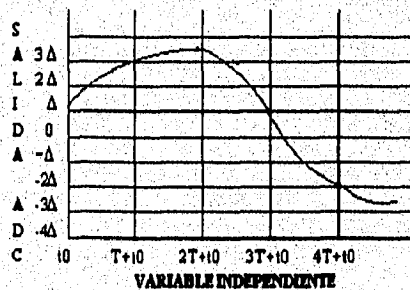
La mayoría de las señales que son procesadas con filtros digitales tienen su origen como una señal analógica antes de que sean procesadas por un filtro digital.

Este proceso de convertir una señal analógica en otra digital constituye una parte fundamental en la mayoría de los sistemas de filtraje digital. El Convertidor Analógico Digital se responsabiliza de tomar muestras de la señal a lo largo de la variable independiente (usualmente el tiempo) y también de muestrear la amplitud de la señal. La variable independiente requiere ser muestreada debido a que el proceso de muestreo debe convertir la variable independiente continua de la señal analógica a los índices discretos de la señal digital. La amplitud también debe ser muestreada, en razón de que la mayoría de los filtros son implementados sobre computadoras analógicas que no pueden representar a los números con una precisión arbitraria. A este proceso se le conoce como error de cuantización, el cual provoca un cierto grado de incertidumbre en la representación, que internamente genera la computadora de los datos.

El convertidor analógico-digital muestrea la señal analógica a lo largo de su variable independiente, evaluándola en valores discretos de esta variable y asignándole los valores cuantizados a los elementos sucesivos de la secuencia digital. En general la señal analógica es muestreada a una variable fija, de modo que las muestras queden igualmente espaciadas a lo largo de esta variable independiente. Al espaciamiento entre las muestras se denomina el periodo de muestreo, lo denotamos con T , el k -ésimo elemento de la secuencia digital corresponderá a la señal cuantizada en el instante $kT+t(0)$, esto es :

$$x(k) = Q\{x(kT + t(0))\}$$

El operador $Q\{\}$ representa el proceso de cuantización y $t(0)$ el valor de la variable independiente cuando el elemento 0-ésimo fue tomado. El proceso de muestreo lo podemos ilustrar a continuación, en donde las líneas horizontales denotan los valores que pueden ser representados por la computadora digital y las líneas verticales a los valores de la variable independiente cuando las muestras son tomadas. El k -ésimo elemento de la secuencia digital queda determinado por la intersección de la k -ésima línea vertical y la línea horizontal más próxima al valor de la señal analógica en el tiempo $kT + T(0)$. La secuencia digital resultante no representa en forma totalmente exacta a la señal analógica debido a los valores imprecisos de las amplitudes de la señal analógica; representando a la señal analógica sólo en algunos valores de su variable independiente.



Las señales analógicas que varían lentamente pueden ser representadas muy aproximadamente como señales digitales derivadas a partir de velocidades de muestreo bajas, en tanto las señales que varían rápidamente requieren de velocidades de muestreo superiores.

A manera de ilustración, el muestreo de una señal coseno analógica puede generar ambigüedad, debido a que es posible para dos señales analógicas de distintas frecuencias generar señales digitales idénticas. Por ejemplo, las señales analógicas de frecuencia w y $w+2\pi/T$ proporcionarán la misma señal digital. A este proceso se le conoce como *aliasing*, ya que por el efecto del proceso de muestreo sobre señales coseno de diferentes frecuencias, se podría asumir la misma imagen digital. Para prevenir tal efecto de *aliasing* el periodo de muestreo debe ser suficientemente pequeño para acomodar las variaciones en la señal analógica.

Una limitación práctica es la velocidad de operación de los convertidores A/D y de las formas del Procesamiento Digital de Señales, lo cual es logrado a través de la muestra de señales y la cuantificación de la misma, resultando una distorsión, que nos impide reconstruir la señal analógica original de la muestra cuantificada.

1.3.- Respuesta de un Sistema en la Frecuencia

Resulta ocioso buscar una función que sea totalmente inalterada cuando pasa por un sistema, pues aunque tal función existe, la función cero, $x_k = 0$, no resulta de ninguna utilidad particular. De este modo el grupo de funciones en el que estamos pensando debe ser mínimamente afectado por cualquier sistema, es decir, se trata de funciones que son escaladas por una constante cuando pasan por el sistema. Una de tales funciones es llamada una eigenfunción y la constante de escalamiento un eigenvalor.

La función digital ek es una eigenfunción de un sistema si y sólo si la salida resultante de la aplicación de ek a la entrada del sistema es la misma función ek , multiplicada por una constante, la constante es el eigenvalor λ , y es una función de la eigenfunción particular y del sistema. Debido a que la entrada y la salida están relacionados por la sumatoria de convolución de la respuesta impulsiva, h_k , con la entrada, una eigenfunción debe satisfacer la relación :

$$\lambda ek = \sum h_k ek$$

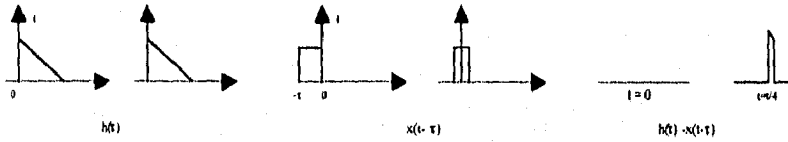
Supóngase que $h(t)$ y $x(t)$ son continuas en una función en el tiempo. La convolución de $h(t)$ y $x(t)$ está representada por :

$$y(t) = h(t) * x(t)$$

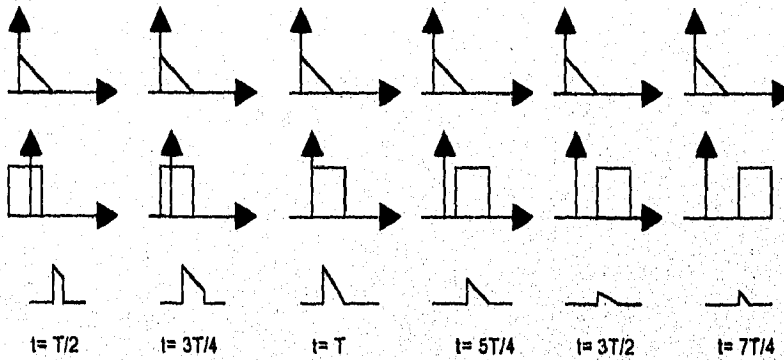
y es definida como una operación aritmética :

$$y(t) = h(t) * x(t - \tau) \text{ DIT}$$

Para evaluar la convolución de 2 funciones en un tiempo en particular t , es necesario multiplicar una de las funciones con la conversión anterior de la otra función (un instante antes), y determinar el área resultante. Este proceso se muestra en la fig. siguiente, para $x(t)$ siendo la función $\text{rect}[(t - T/2)/T]$ y $h(t)$ siendo una función cerrada de una longitud T .



Repetiendo el proceso para toda t y determinando el área sobre el tiempo, $y(t)$ es una variable compleja. Nótese que la convolución siempre invoca una mancha fuera de las ondas que fueron convolucionadas, la amplitud del resultado corresponde a la suma de las amplitudes de las funciones.



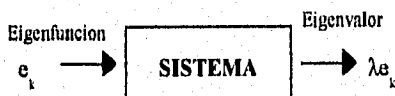
En otras palabras la convolución de una función $x(t)$ con un impulso de longitud A , posicionado en $t = t_0$, simplemente reposiciona la función de entrada $x(t)$ en la localidad del impulso, con escala A . La salida de un filtro en particular, es posible computarla para cualquier entrada dada; sin embargo siempre existen algunas entradas que representan mayor interés que otras, la secuencia impulsiva que vista en forma digital es una entrada (a diferencia de su contraparte analógica) que es 1 en el índice 0 y 0 de otra manera:

$$\delta_k = \begin{cases} 1 & \text{si } k=0 \\ 0 & \text{de otra manera} \end{cases}$$

La respuesta impulsiva de un filtro es por definición, su salida cuando la entrada es un impulso, el escalón discreto es 1 para índices positivos (incluyendo el 0) y 0 para índices negativos :

$$U_k = \begin{cases} 1 & \text{si } k > 0 \\ 0 & \text{de otra manera} \end{cases}$$

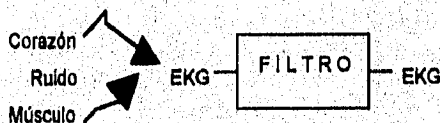
y cuando este escalón es presentado como la entrada de un filtro, la salida del mismo es referida como la respuesta escalón del filtro.



1.4.- Concepto de Filtro

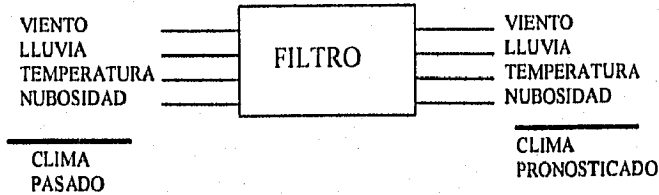
Un filtro es una caja negra con un conjunto de entradas y salidas. La caja contiene alguna forma de procesamiento que genera las salidas a partir de las entradas. En general existen dos fuertes motivaciones para ejecutar operaciones de filtraje; una es la de mejorar la calidad de las entradas, la otra es el procesamiento o extracción de información a partir de las entradas. A continuación se mencionarán dos ejemplos de utilización.

a) El electrocardiograma (EKG) es una importante herramienta en el diagnóstico y tratamiento de los problemas cardíacos. La señal EKG es simplemente el potencial eléctrico generado por los latidos del corazón; a partir de esta representación eléctrica, el especialista puede inferir muchas condiciones asociadas con el desorden cardíaco. Todos los músculos, no sólo el corazón, generan potenciales eléctricos; este ruido no es de utilidad e interfiere con el diagnóstico; por lo que es de mucha utilidad un filtro que remueva o que por lo menos atenúe la componente de ruido muscular en la señal del electrocardiograma.



b) Es nuestro propósito construir una estación meteorológica automática en la que, vista como un filtro, se pronostique el clima, la temperatura, el viento y la lluvia de mañana a partir de mediciones de la temperatura, viento, lluvia y presión barométrica anteriores. En esta aplicación

asumimos que todas las mediciones (entradas) son exactas, de modo que no hay razón para filtrarlas para mejorar la calidad de las mismas. El único propósito del filtro, en este caso, es extraer información a partir de los datos.



1.5.- Tipos de Filtros

1.5.1. -Tipos de Filtros

Como se sabe un filtro es un proceso de entradas y salidas, por estas mismas características existen dos tipos de filtro: los Filtros Analógicos y los Filtros Digitales. Los filtros analógicos difieren de los digitales por la naturaleza de la entrada y la salida. Un filtro analógico procesa entradas analógicas y genera salidas analógicas; en tanto que un filtro digital procesa y genera datos digitales. Los filtros analógicos están basados en la relación de las operaciones matemáticas de diferenciación e integración; en tanto que los filtros digitales sólo requieren desarrollar operaciones de suma, multiplicación y retardo.

La mayoría de las señales se representan en términos de dos variables, una dependiente y la otra independiente; la variable dependiente representará al dato, en tanto que la variable independiente representa el orden de este dato. La información analógica es caracterizada por una variable independiente continua que puede tomar cualquier valor, en tanto que la información digital se caracteriza por una variable independiente discreta que sólo puede tomar determinados valores.

En resumen los datos digitales son caracterizados por una variable independiente discreta. Algunos datos son digitales de modo natural, sin embargo, todos los tipos de datos pueden ser convertidos a un formato digital, restringiendo los valores de la variable independiente. Los filtros digitales, operan exclusivamente sobre entradas digitales y generan salidas digitales. Los datos o información digital son representados como una secuencia de números ordenados, esta secuencia puede ser finita con índices y fin arbitrarios :

$$X_2, X_4, X_5, \dots, X_{18}$$

o pueden iniciar en algún punto arbitrario y continuar para siempre :

$$X_{24}, X_{35}, X_{36}, \dots$$

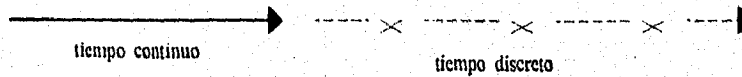
la secuencia de datos más general tiene su punto de partida en el menos infinito y continúa por siempre hasta el más infinito :

$$\dots x_{-1}, x_0, x_1, \dots$$

En resumen los datos digitales son representados por una secuencia ordenada. La secuencia es indizada por enteros y cualquier elemento de esta secuencia ordenada es denotado por x (índice). Estos índices pueden requerir un intervalo de valores finito o infinito.

1.5.2.-Características de los Filtros Digitales

(1) El procesamiento de un filtro digital es DISCRETO en el tiempo. Las operaciones, entrada/procesamiento/salida, requieren un tiempo finito para completarse; de esta manera el filtrado no puede ser continuo.



(2) El procesamiento de un filtro digital es Discreto en sus datos. Las operaciones, entrada/procesamiento/salida, tienen una resolución finita (dígitos o bits fijos); de esta manera no pueden ser continuos.



1.5.3.- Dominios para el estudio de Filtros

Los filtros digitales se pueden estudiar en el dominio analógico s , en el dominio digital z , en el dominio de la frecuencia f , y en el dominio del tiempo t .

Dominio s	Cómo se especifica un filtro analógico.
Dominio z	Cómo se implementa un filtro digital.
Dominio f	Cómo analizar un filtro analógico o un filtro digital.
Dominio t	Cómo verificar un filtro digital.

Además es posible operar las siguientes transformaciones entre esos cuatro dominios :

- $s-z$ Convierte un filtro analógico a un filtro digital equivalente.
- $s-f$ Computa la respuesta en frecuencia de un filtro analógico.

- z-f** **Computa la respuesta en frecuencia de un filtro digital.**
- t-f** **Mide la respuesta en frecuencia de un filtro digital.**
- z-t** **evalúa la respuesta en tiempo de un filtro digital.**

FILTROS DIGITALES

2. -FILTROS DIGITALES

2.1.- Diseño de Filtros con análisis de Fourier.

La respuesta en frecuencia de un sistema discreto lineal e invariante en tiempo es periódica con periodo 2π ; y la respuesta en frecuencia $H(e^{j\omega})$ y la respuesta impulsiva $h(n)$ están relacionadas por

$$H(e^{j\omega T}) = \sum_{n=-\infty}^{n=\infty} h(nT) e^{j\omega n T} \dots (1)$$

para un sistema causal no recursivo, como :

$$y(n) = \sum_{k=0}^{k=L-1} b_k x(n-k) \dots (2)$$

sabemos que cuando la entrada es $x(n) = e^{jn\theta}$ entonces la respuesta en estado estacionario queda especificada por :

$$y_{ss}(n) = e^{jn\theta} H(e^{j\theta}) \dots (3)$$

donde $e^{jn\theta}$ es la eigenfunción y $H(e^{j\theta})$ es el eigenvalor asociado. Sustituyendo la ec. (3) en el lado izquierdo de la ec. (2) y $x(n) = e^{jn\theta}$ en el lado derecho, obtenemos la respuesta en la frecuencia :

$$H(e^{j\theta}) = \sum_{n=0}^{n=L-1} b_n e^{jn\theta} \dots (4)$$

Por otro lado al escribir la ec. (1) para un filtro causal, esto es, $h(n) = 0$ para $n < 0$, con un número finito de retardos L , tendremos que :

$$H(e^{j\theta}) = \sum_{n=0}^{n=L-1} h(n) e^{jn\theta} \dots (5)$$

y si comparamos con la ec. (4) tenemos :

$$h(n) = b_n \dots (6)$$

la cual establece que los valores de la respuesta impulsiva $h(n)$ son iguales a los coeficientes b_n en la ecuación diferencial en el sistema para un filtro no recursivo. Al diseñar filtros iniciamos con las especificaciones de la respuesta en frecuencia deseada y la determinación de los coeficientes del filtro b_0, b_1, \dots, b_L ; la ec. (1) da la relación para la respuesta en frecuencia $H(e^{j\theta})$ en términos de los coeficientes del filtro $h(n)$; sin embargo, requerimos de una expresión para los coeficientes del filtro en términos de la respuesta en frecuencia, como :

$$h(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\theta}) e^{-jn\theta} d\theta, \quad n=0, \pm 1, \pm 2, \dots \dots (7)$$

la ec. (7) junto con la (1) constituyen el par de transformadas de Fourier, esto es, si conocemos $H(e^{j\theta})$, la otra parte del par podemos determinarla usando la sig. ec. con $\theta = -\theta$

$$H(e^{j\theta}) = \sum_{n=-\infty}^{n=+\infty} h(n)e^{jn\theta} \Leftrightarrow h(n) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} H(e^{j\theta}) e^{-jn\theta} d\theta \quad \dots(8)$$

síntesis
análisis

la ecuación para $H(e^{j\theta})$ es conocida como expresión de síntesis debido a que la respuesta en frecuencia es sintetizada a partir de la suma de los coeficientes de un número infinito de exponenciales complejas $e^{jn\theta}$, $n = 0, \pm 1, \dots$. Por otro lado la ecuación para los coeficientes $h(n)$ es una expresión de análisis y el área bajo la curva determina las $h(n)$ para todos los valores correspondientes de n . Para una señal analógica muestreada, el par lo podemos denotar como :

$$H(e^{j\omega T}) = \sum_{n=-\infty}^{n=+\infty} h(nT) e^{jn\omega T} \Leftrightarrow h(nT) = \frac{1}{W_s} \int_{-W_s/2}^{W_s/2} H(e^{j\omega T}) e^{-jn\omega T} d\omega \quad \dots(10)$$

donde $W_s = 2\pi/T$ es la frecuencia de muestreo en rad/s. De este modo, el diseño de un filtro digital lo podemos realizar con el siguiente procedimiento :

1. Determinar la respuesta en frecuencia para $H(e^{j\theta})$, la cual es obtenida para la aplicación particular del filtro.
2. Determinar la respuesta impulsiva $h(n)$, la cual produce la respuesta en frecuencia deseada; en otras palabras encontrar los valores de $h(n)$ en la ec. (7).
3. Modificar la respuesta impulsiva $h(n)$ para obtener un filtro práctico, en la ec. (8) vemos que para producir la respuesta en frecuencia $h(e^{j\theta})$ se requiere un número infinito de valores para $h(n)$. Esto es impráctico y, por lo tanto, $h(n)$ debe ser truncada a un número razonable de términos.
4. Implementar el filtro digital con un programa apropiado de computadora o un circuito integrado.

2.2.- Filtro Pasa Bajas

El filtro de pasa bajas permite sólo el paso de frecuencia bajas de una señal de entrada que pasa a través del filtro. Este filtro es el más importante de todos, ya que presenta un buen rechazo al ruido.

La función de transferencia del filtro digital en el dominio de s es :

$$\frac{1}{(Ts+1)}$$

Las ecuaciones del filtro digital en el dominio de z son :

- | | |
|--------------------------|---|
| (1) Adelanto rectangular | $Sal = Sals + (Ent - Sals) * dt / (dt + \tau)$ |
| (2) Retraso rectangular | $Sal = Sals + (Ent - Sals) * dt / \tau$ |
| (3) Trapezoidal | $Sal = Sals + (Ent + Ents - Sals - Sals) * dt / (dt + 2\tau)$ |

El filtro de pasa bajas trapezoidal normalmente proporciona los mejores resultados, el filtro pasa bajas de adelanto rectangular es el más fácil de implementar. En resumen, cuando τ es mucho mayor que dt , el término $dt / (dt + \tau)$ en la ecuación del filtro se podrá aproximar por dt / τ ; el resultado será $Sal = Sals + (Ent - Sals) * dt / \tau$ que es el filtro de pasa bajas más práctico.

Además la multiplicación se puede aproximar haciendo "un desplazamiento a la derecha" en un compositor digital para aumentar la velocidad en el proceso de filtrado.

2.3.- Filtro Pasa Altas

El filtro pasa altas permite únicamente el paso de señales de frecuencia alta, y es lo opuesto al filtro pasa bajas; la diferencia de este filtro con respecto al pasa bajas es que es muy sensible al ruido. El principal uso de los filtros pasa altas es el de remoción del error por ajuste en una señal.

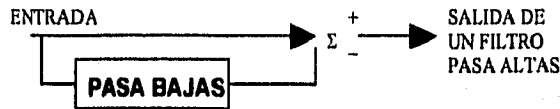
La función de transferencia en el dominio de s es $\tau s / (\tau s + 1)$. Las ecuaciones digitales en el dominio de z son :

- | | |
|----------------------------|---|
| (1) Adelanto rectangular : | $Sal = (Sals + Ent - Ents) * \tau / (dt + \tau)$ |
| (2) Retraso rectangular : | $Sal = Sals * \tau (\tau - dt) / t + (Ent - Ents)$ |
| (3) Trapezoidal | $Sal = Sals * (2\tau - dt) / (2\tau - dt) * (Ent - Ents) * 2\tau / (2t + dt)$ |

En el dominio de s, el filtro pasa altas = 1 - filtro pasa bajas

$$\tau s / (\tau s + 1) = 1 - 1 / (\tau s + 1)$$

En el dominio del tiempo un filtro pasa altas se puede construir utilizando un filtro pasa bajas con la misma constante τ , como se muestra :



En aplicaciones reales, τ deberá ser mucho mayor que dt .

2.4.- Filtro Pasa Banda

Los filtros pasa banda sólo permiten el paso de señales de media frecuencia. La banda de paso de estos filtros se diseña para ser muy estrecha, de tal manera que solamente sean reconocidas y puedan pasar las frecuencias estables. Estos filtros generalmente se usan para generar una banda estrecha de ruido en una banda ruidosa. Una banda estrecha de ruido es un tipo de ruido que contiene componentes de frecuencia en una banda de frecuencia angosta; este ruido se observa como una onda senoidal, pero su fase y magnitud cambian repentinamente. La función de transferencia en el dominio de s es :

$$\frac{4\pi f_c \cdot k \cdot s}{s^2 + 4\pi f_c \cdot k \cdot s + (2\pi f_c)^2}$$

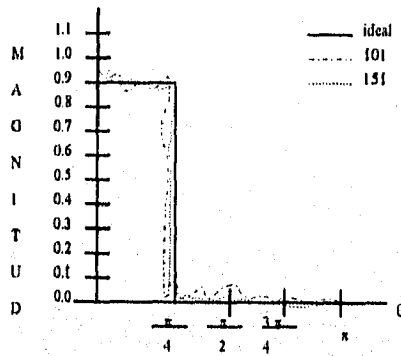
En el ajuste del filtro se realizan los siguientes pasos :

- 1.- Seleccionar la banda media de la frecuencia f_c
- 2.- Seleccionar k para controlar la respuesta en frecuencia del filtro.

A medida que se disminuye el valor de k , la respuesta será más aguda.

Características	Frecuencia	Ganancia	Fase
banda de rechazo	$< f_c$	< 1	adelantada
centro de la banda de paso	$= f_c$	1	0°
banda de rechazo	$> f_c$	> 1	defasada

coeficientes. Notamos que los valores pico cerca de la discontinuidad en ambas gráficas son aproximadamente de 1.09.



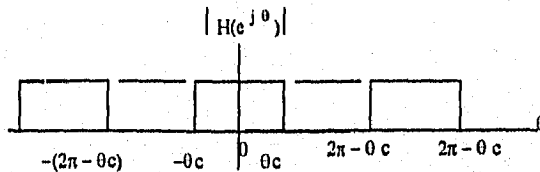
2.5.- Fenómeno de Gibbs

El fenómeno de Gibbs es muy importante en la teoría de filtros. La magnitud en la respuesta en frecuencia para un filtro ideal pasa bajas (fase cero) la observamos en la siguiente fig. y el par apropiado de la serie de Fourier es :

$$h(n) = \frac{1}{2\pi} \int_{-\theta c}^{\theta c} e^{jn\theta} d\theta = \frac{1}{n\pi} \text{sen}(n\theta c) \quad n = 0, \pm 1, \pm 2, \dots$$

con

$$H(e^{j\theta}) = \sum_{n=-\infty}^{\infty} h(n)e^{jn\theta}$$



La respuesta en frecuencia de la fig. no puede ser realizada por la ecuación 2 por dos razones :

Primero, no podemos implementar un filtro con un número infinito de términos como se requiere en la ecuación de síntesis para $H(e^{j\theta})$

Segundo cuando existe discontinuidad en la gráfica de magnitud como en la de la figura y aunque usemos un número infinito de términos, no será imposible obtener la respuesta en frecuencia para estas discontinuidades.

Para filtros de alto orden siempre habrá un sobre tiro de aproximadamente 8.95 % cerca de una discontinuidad, a esto se le conoce como Fenómeno de Gibbs. Este efecto puede verse claramente en la sig. figura, donde se muestra la aproximación de Fourier para filtros pasa bajas con 101 y 151

TRANSFORMADA RAPIDA DE FOURIER

3.- TRANSFORMADA RAPIDA DE FOURIER

3.1.- Transformada Discreta de Fourier

El desarrollo del par de Transformadas Discretas de Fourier, es sólo necesario para derivar la relación matemática de los resultados, con cada una de las modificaciones requeridas como : el dominio en el tiempo de la muestra, truncación y el dominio en la frecuencia muestreada.

Considere el par de Transformadas de Fourier de la fig. (a). Para discretizar este par de transformadas, primero es necesario muestrear la onda $h(t)$, el muestreo puede escribirse como $h(t)\Delta o(t)$ donde $\Delta o(t)$ es el dominio en el tiempo de la función muestreada (fig. b). El intervalo del muestreo es T , la función puede representarse así :

$$\begin{aligned} h(t)\Delta o(t) &= h(t) \sum_{k=-\infty}^{\infty} \delta(t-kT) \\ &= \sum_{k=-\infty}^{\infty} h(kT) \delta(t-kT) \quad \dots\dots(1) \end{aligned}$$

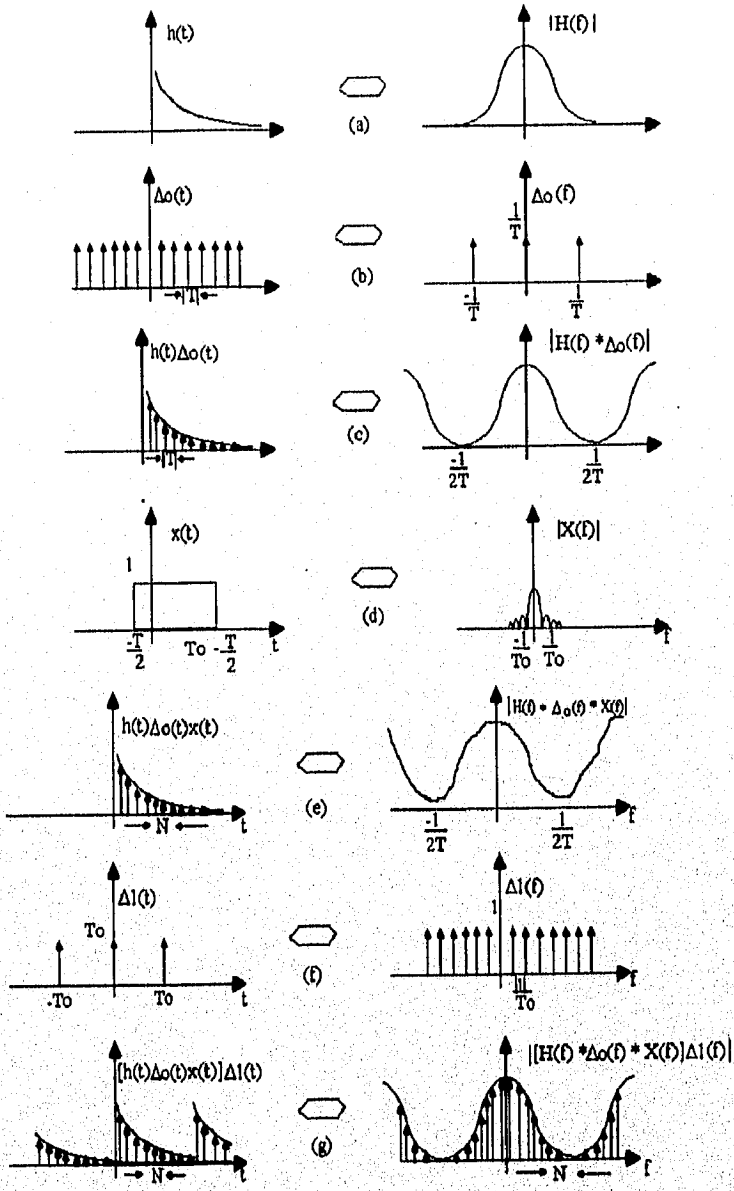
El resultado de esta multiplicación puede observarse en la fig. (c), nótese que el efecto de aliasing es resultado de T . Después la función muestreada es truncada por multiplicación con el rectángulo de la función $x(t)$, fig. (d).

$$x(t) = \begin{cases} 1 & -T/2 < t < T_0 - T/2 \\ 0 & \text{en otro caso} \end{cases} \quad \dots (2)$$

donde T_0 es la duración de la función truncada. Una pregunta obvia sobre este punto es ¿Porqué la función rectangular $x(t)$ no está centrada en cero o en $T_0/2$? El centro de $x(t)$ en cero es aludido por problemas de notación. La razón para no centrar la función rectangular en $t/2$ se puede volver obvia en el siguiente planteamiento truncando los campos :

$$\begin{aligned} h(t)\Delta o(t)x(t) &= \left[\sum_{k=-\infty}^{\infty} h(kT) \delta(t-kT) \right] x(t) \\ &= \sum_{k=0}^{N-1} h(kT) \delta(t-kT) \quad \dots\dots(3) \end{aligned}$$

donde ha sido asumido que hay N funciones impulso equidistantes falsas dentro del intervalo truncado, que es $N=T_0/T$. El ejemplo de la onda truncada y su Transformada de Fourier son ilustradas en la fig. (e). Como en el ejemplo anterior, la truncación en el dominio del tiempo resulta una ondulación en el dominio de la frecuencia.



Derivación Gráfica del par de Transformadas Discretas de Fourier

El paso final en la modificación del par de Transformadas de Fourier originales, es mostrar la Transformada de Fourier de la ec. (3) en el dominio del tiempo, este producto es equivalente a convolucionar la onda truncada mostrada en la ec. (3) y la función tiempo $A_1(t)$, ilustrada en la fig. (f). La función $A_1(t)$ está dada por :

$$A_1(t) = T_0 \sum_{k=0}^{N-1} \delta(t - kT_0) \quad \dots\dots(4)$$

La relación deseada es $[h(t)\Delta_0(t)x(t)]\Delta_1(t)$ por lo tanto

$$\begin{aligned}
 [h(t)\Delta_0(t)x(t)] * \Delta_1(t) &= \left[\sum_{k=0}^{N-1} h(kt) \delta(t - kT) \right] * \left[T_0 \sum_{r=0}^{N-1} \delta(t - rT_0) \right] \\
 &= \dots + T_0 \sum_{k=0}^{N-1} h(kt) \delta(t + T_0 - kT) \\
 &\quad + T_0 \sum_{k=0}^{N-1} h(kt) \delta(t - kT) \\
 &\quad + T_0 \sum_{k=0}^{N-1} h(kt) \delta(t - T_0 - kT) + \dots \quad (5)
 \end{aligned}$$

Nótese que es una función periódica con periodo T_0 . En una notación más compacta que se puede escribir como :

$$h(t) = T_0 \sum_{k=0}^{N-1} h(kT) \delta(t - kT - rT_0) \quad \dots(6)$$

Se escogió la notación de $h(t)$ para decir que es una aproximación de $h(t)$. Nótese que el resultado de la convolución ec. (6) es una función periódica con periodo T_0 , el cual está formado de N muestras. Si la función rectangular ha sido escogida, tal valor muestreado coincide con el último punto de la función rectangular, la convolución de la función rectangular con impulsos espaciados en el intervalo T_0 , puede resultar un aliasing en el dominio del tiempo. Esto es el punto N de un periodo puede coincidir con (y sumarse a) el primer punto del siguiente periodo. Para estar seguro que el aliasing en el dominio del tiempo no ocurrirá, es necesario escoger el intervalo de truncación (fig. d), nótese que los puntos finales de la función de truncación mienten en el punto medio de dos valores muestreados adyacentes permitiendo el aliasing en el dominio del tiempo).

La Transformada de Fourier de una función periódica $h(t)$ es una secuencia de impulsos equidistantes

$$H(n/T_0) = \sum_{n=-\infty}^{\infty} \alpha_n \delta(f - n f_0) \quad f_0 = 1/T_0 \quad \dots(7)$$

donde

$$\alpha_n = 1/T_0 \int_{-T/2}^{T/2} h(t) e^{j2\pi n t / T_0} dt \quad n=0, \pm 1, \pm 2, \dots \quad (8)$$

Substituyendo la (ec. 6) en la (ec. 8) se tiene

$$\alpha_n = \frac{1}{T_0} \int_{-T/2}^{T_0-T/2} \sum_{k=0}^{N-1} h(kT) \delta(t-kT-rT_0) e^{j2\pi n t / T_0} dt$$

la integración es sólo sobre un periodo, entonces :

$$\begin{aligned} \alpha_n &= \int_{-T/2}^{T_0-T/2} \sum_{k=0}^{N-1} h(kT) \delta(t-kT) e^{j2\pi n t / T_0} dt \\ &= \sum_{k=0}^{N-1} h(kT) \int_{-T/2}^{T_0-T/2} e^{j2\pi n t / T_0} \delta(t-kT) dt \\ &= \sum_{k=0}^{N-1} h(kT) e^{-j2\pi n k / T_0} \quad \dots(9) \end{aligned}$$

como $T_0 = NT$, la ec.(9) puede reescribirse como :

$$\alpha_n = \sum_{k=0}^{N-1} h(kT) e^{-j2\pi n k / T_0} \quad n = \pm 1, \pm 2 \dots (10)$$

y la transformada de Fourier de la ec. (6) es :

$$H(n/NT) = \sum_{n=-\infty}^{\infty} \sum_{k=0}^{N-1} h(kT) e^{-j2\pi n k / N} \quad \dots(11)$$

Con una evaluación precipitada de la ec.(11) no es obvio que la transformada de Fourier $H(n/NT)$ no es periódica, como se ilustra en la fig. (g), porque hay sólo N distintos valores complejos computables de la ec. (11). Para establecer este factor haremos $n=r$ donde r es un entero arbitrario, entonces :

$$H(r/NT) = \sum_{k=0}^{N-1} h(kT) e^{-j2\pi r k / N} \quad \dots(12)$$

Ahora $n = r + N$; nótese que :

$$\begin{aligned} e^{-j2\pi k(r+N)/N} &= e^{-j2\pi r k / N} e^{-j2\pi k} \\ &= e^{-j2\pi r k / N} \quad \dots(13) \end{aligned}$$

donde $e^{jnk} = \cos(2pk) - j\sin(2pk) = 1$ para k tomando valores enteros. Esto para $n = r + N$

$$\begin{aligned} \frac{H(r+N)}{NT} &= \sum_{k=0}^{N-1} h(kT) e^{j2nk(r+N)/N} \\ &= \sum_{k=0}^{N-1} h(kT) e^{j2nk r/N} \\ &= H(\underline{r}) \quad \dots(14) \end{aligned}$$

Por lo tanto, hay sólo N valores distintos los cuales (ec. 11) pueden ser valuados; $H(n/NT)$ es periódica con un periodo de N muestras. La transformada de Fourier puede expresarse también como :

$$H(\underline{n}) = \sum_{k=0}^{N-1} h(kT) e^{j2nk/N} \quad n=0,1,\dots,N-1 \dots(15)$$

La ec. (15) es la transformada de Fourier deseada; la expresión tiene N muestras de tiempo y N muestras de frecuencia por la forma continua de la Transformada de Fourier. Si se asume que las N muestras de la función original $h(t)$ es un periodo de la onda periódica, la transformada de Fourier de esta función periódica está dada por las N muestras computadas (ec. 15). La notación $H(n/NT)$ es usada para indicar que la transformada de Fourier discreta es una aproximación a la transformada de Fourier continua. Normalmente se escribe :

$$G(\underline{n}) = \sum_{k=0}^{N-1} g(kT) e^{j2nk/N} \quad n=0,1,\dots,N-1 \dots(16)$$

donde la transformada de Fourier de la función periódica muestreada $g(kT)$ es idéntica a $G(n/NT)$.

3.2.- Transformada Rápida

Considere la transformada discreta de Fourier

$$X(n) = \sum_{k=0}^{N-1} x_0(k) e^{j2\pi nk/N} \quad n=0,1,\dots,N-1 \dots(1)$$

donde tenemos que reemplazar kT por k y n/NT por n para conveniencias de notación. Notemos que (ec. 1) describe N ecuaciones computables. Por ejemplo si $N=4$ y si permitimos que

$$W = e^{j2\pi/N} \quad \dots (2)$$

entonces la ec. (1) puede escribirse como :

$$\begin{aligned} X(0) &= x_0(0) W_0 + X_0(1) W_0 + x_0(2) W_0 + x_0(3) W_0 \\ X(1) &= x_0(0) W_0 + X_0(1) W_1 + x_0(2) W_2 + x_0(3) W_3 \\ X(2) &= x_0(0) W_0 + X_0(1) W_2 + x_0(2) W_4 + x_0(3) W_6 \dots(3) \\ X(3) &= x_0(0) W_0 + X_0(1) W_3 + x_0(2) W_6 + x_0(3) W_9 \end{aligned}$$

La ec. (3) puede ser más fácil representándola en forma de matriz :

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W_0 & W_0 & W_0 & W_0 \\ W_0 & W_1 & W_2 & W_3 \\ W_0 & W_2 & W_4 & W_6 \\ W_0 & W_3 & W_6 & W_9 \end{bmatrix} \begin{bmatrix} X_0(0) \\ X_0(1) \\ X_0(2) \\ X_0(3) \end{bmatrix} \dots (4)$$

y más compacto como :

$$X(n) = W_{nk} X_0(k) \dots(5)$$

Examinando la ec. (4) revela que W y posiblemente $x_0(k)$ son complejas, entonces N^2 es una multiplicación compleja y $(N)(N-1)$ es una adición compleja necesaria para los requerimientos computacionales.

Desarrollo Intuitivo.

Para ilustrar el algoritmo de la Transformada Rápida de Fourier, es conveniente escoger el número de puntos muestrales $x_0(k)$ de acuerdo a la siguiente relación $N=2^r$, donde r es un entero. Más tarde el desarrollo removerá esta restricción. El resultado de la ec. (4) es resultado de escoger $N=4=2^2$, por lo tanto podemos aplicar la Transformada Rápida de Fourier a la ec. (4).

El primer paso para desarrollar la Transformada Rápida de Fourier, para este ejemplo es reescribir la ec. (4). como

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_1 & W_2 & W_3 \\ 1 & W_2 & W_0 & W_3 \\ 1 & W_3 & W_2 & W_1 \end{bmatrix} \begin{bmatrix} X_0(0) \\ X_0(1) \\ X_0(2) \\ X_0(3) \end{bmatrix} \dots (6)$$

La matriz de la ec.(6) fue derivada de la ec.(4) usando la relación $W_{nk} = W_{nk \bmod N}$. Recuerde que $[nk \bmod N]$ es el residuo de la división de nk entre N ; si $N=4$, $n=2$, y $k=3$ entonces

$$W_6 = W_2 \dots (7)$$

donde

$$\begin{aligned} W_{nk} &= W_6 = \text{EXP} \{(-j2\pi/4)(6)\} = \exp[-j3\pi] \\ &= \exp[-j\pi] = \exp \{(-j2\pi/4)(2)\} = W_2 = W_{nk \bmod N} \dots (8) \end{aligned}$$

El segundo paso en el desarrollo es factorizar la matriz cuadrada en :

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & W_0 & 0 & 0 \\ 1 & W_2 & 0 & 0 \\ 0 & 0 & 1 & W_1 \\ 0 & 0 & 1 & W_3 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & W_0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & W_2 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & W_0 \\ W_0 & 0 \\ 0 & W_2 \\ W_2 & 0 \end{bmatrix} \begin{bmatrix} X_0(0) \\ X_0(1) \\ X_0(2) \\ X_0(3) \end{bmatrix} \dots (9)$$

El método de factorización está basado en la teoría de la Transformada Rápida de Fourier. Es suficiente mostrar que la multiplicación de las dos matrices cuadradas (ec. 9) dan como producto una matriz cuadrada (ec. 6), con excepción de los renglones 1 y 2 que han sido intercambiados. (la numeración es 0, 1, 2, 3). Nótese que este intercambio ha sido tomado en cuenta (ec. 6) reescribiendo el vector columna $X(n)$; denotando el intercambio del vector por :

$$X(N) = \begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} \dots (10)$$

Se tiene que aceptar que el factor (ec. 9) es correcto, aunque los resultados estén revueltos, se debe de examinar el número de multiplicaciones requeridos para computar la ecuación, primero tenemos :

$$\begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & W_0 & 0 \\ 0 & 1 & 0 & W_0 \\ 1 & 0 & W_2 & 0 \\ 0 & 1 & 0 & W_2 \end{bmatrix} \begin{bmatrix} X_0(0) \\ X_0(1) \\ X_0(2) \\ X_0(3) \end{bmatrix} \dots (11)$$

Este es el vector columna $x_1(k)$ que es igual al producto de las dos matrices de la derecha (ec. 9). El elemento $x_1(0)$ es computado por una multiplicación compleja y una suma compleja.

$$x_1(0) = x_0(0) + W_0 x_0(2) \dots (12)$$

El elemento $x_1(1)$ es también determinado por una multiplicación compleja y una suma. Sólo una suma compleja es requerida para computar $x_1(2)$:

$$\begin{aligned} x_1(2) &= x_0(0) + W^2 x_0(2) \\ &= x_0(0) - W^0 x_0(2) \dots (13) \end{aligned}$$

donde la multiplicación compleja $W_0 x_0(2)$ ha sido ya computada en el determinante de $x_1(0)$ (ec. 12). Por el mismo razonamiento, $x_1(3)$ es computada por una sola suma compleja y no multiplicaciones. El vector intermedio está entonces determinado por cuatro sumas complejas y dos multiplicaciones complejas. Sigamos con el desarrollo del método (ec. 9).

$$\begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} = \begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \\ X_2(3) \end{bmatrix} = \begin{bmatrix} 1 & W_0 & 0 & 0 \\ 1 & W_2 & 0 & 0 \\ 0 & 0 & 1 & W_1 \\ 0 & 0 & 1 & W_3 \end{bmatrix} \begin{bmatrix} X_0(0) \\ X_0(1) \\ X_0(2) \\ X_0(3) \end{bmatrix} \dots (14)$$

El término $x_2(0)$ es determinado por una multiplicación compleja y una suma.

$$x_2(0) = x(0) + W_0 x_1(1) \dots (15)$$

El elemento $x_2(1)$ es computado por una suma porque $W_0 = -W_2$. De forma similar $x_2(2)$ es determinado por una multiplicación y una suma compleja, y $x_2(3)$ sólo por una suma.

Computar $X(n)$ significa que la ec. (10) requiere un total de 4 multiplicaciones complejas y 8 sumas complejas. Calcular $X(n)$ por la ec. (4) requiere 16 multiplicaciones complejas y 20 sumas complejas. Nótese que el proceso de factorización de la matriz introduce ceros en los factores de la matriz y, como un resultado, reduce el número de multiplicaciones requeridas. Para este ejemplo, el proceso de factorización de la matriz, reduce el número de multiplicaciones por un factor de dos. Para el cálculo el tiempo está mayormente gobernado por el número requerido de multiplicaciones, Aquí se ve la eficiencia del algoritmo.

Para $N=2^r$ el algoritmo de la transformada rápida de Fourier es simplemente un procedimiento para factorizar matrices de $N \times N$ en r matrices (cada una de $N \times N$). Cada factor de la matriz tiene propiedades especiales de minimizar el número de sumas y multiplicaciones complejas. Si se extiende esto al ejemplo anterior, entonces la transformada rápida de Fourier requiere $Nr/2 = 4$ multiplicaciones complejas y $Nr=8$ sumas complejas, donde el método directo (ec. 4) requiere N^2 multiplicaciones complejas y $N(N-1)$ sumas complejas. Si se asume que el tiempo de cálculo es proporcional al número de multiplicaciones, entonces la relación aproximada del cálculo de la transformada rápida de Fourier está dada por :

$$\frac{N^2}{Nr/2} = \frac{2N}{r} \dots (16)$$

el cual para $N=1024 = 2^{10}$ es una reducción computacional de más del 200 %. En la siguiente figura se ilustra la relación entre el número de multiplicaciones requeridas por la transformada rápida de Fourier y el método directo.

El procedimiento de factorización de la matriz introduce una discrepancia. Recuerde que el

resultado $\overline{X(n)}$ toma el lugar de $X(n)$ esto es :

$$\overline{X(n)} = \begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} \text{ en lugar de } X(n) = \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} \dots (17)$$

Este nuevo arreglo es inherente en el proceso de factorización y es un problema menor porque es sencillo generalizar una técnica para descifrar $X(n)$ para obtener $\overline{X(n)}$.

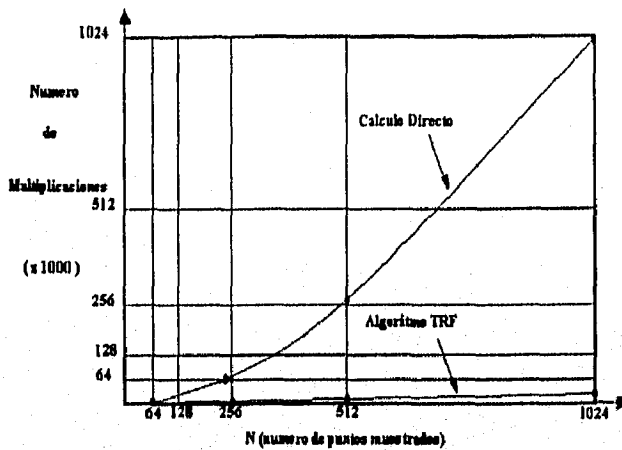
Reescribiendo $\overline{X(n)}$ reemplazando el componente binario n con su equivalente binario :

$$\begin{bmatrix} X(0) \\ X(2) \\ X(1) \\ X(3) \end{bmatrix} \text{ se vuelve } \begin{bmatrix} X(00) \\ X(10) \\ X(01) \\ X(11) \end{bmatrix} \dots (18)$$

Obsérvese que si el argumento binario de la ec.(18) cambia de golpe, o revierte binariamente (esto es que 01 cambia a 10 y 10 cambia a 01) se tiene :

$$\overline{X(n)} = \begin{bmatrix} X(00) \\ X(10) \\ X(01) \\ X(11) \end{bmatrix} \text{ cambia a } \begin{bmatrix} X(00) \\ X(01) \\ X(10) \\ X(11) \end{bmatrix} \dots (19)$$

Que es una forma rara de descifrar el desarrollo generalizado de la transformada rápida de Fourier.



3.3.- Algoritmo del Método.

3.3.1.- Algoritmo.

La evaluación directa de la Transformada Discreta de Fourier requiere N multiplicaciones complejas y $N-1$ sumas complejas para cada valor de $X(k)$, y donde hay N valores para determinar, son necesarias N^2 multiplicaciones y $N(N-1)$ sumas. Consecuentemente, para valores de N muy grandes, como 1000, la evaluación directa requiere considerablemente a la computadora.

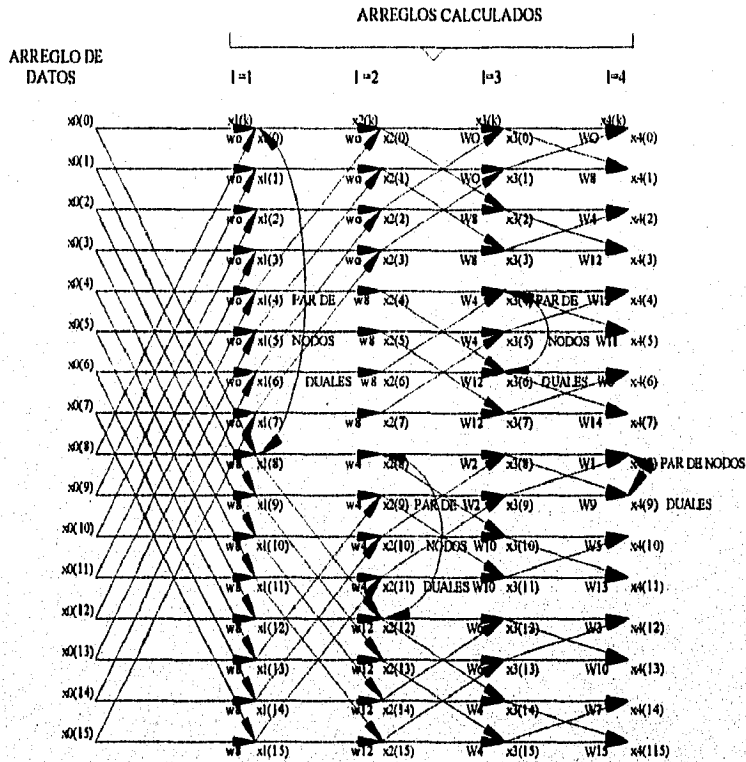
La eficiencia y la moderna aproximación para la evaluación de la Transformada Discreta de Fourier es a través del algoritmo de la Transformada Rápida de Fourier.

3.3.2.- Representación Gráfica de una Señal.

Convertimos la ec. (9) en una representación gráfica, como se muestra a continuación; representamos el vector o arreglo de datos por una columna vertical de nodos a la izquierda de la figura. El segundo arreglo vertical de nodos es el vector $x_1(k)$ calculado en la ec. (11) y el siguiente arreglo vertical corresponde al vector $x_2(k) = X(n)$, ec. (11). En general debe de haber $\log_2 N$ arreglos calculados donde $N = 2^y$.

La interpretación de la figura es como sigue. Cada nodo tiene dos líneas sólidas las cuales representan las rutas de cada nodo. La ruta transmite o brinda una calificación proveniente de un arreglo, multiplica la calificación por W^p , e introduce el resultado en el nodo del siguiente arreglo. El factor W^p aparece cerca de las flechas de ruta, abstenerse de este factor implica que $W^p = 1$. El resultado entra al nodo proveniente de la suma de dos direcciones.

La representación gráfica es entonces un método para representar los requerimientos computacionales en factores de matrices para el algoritmo de la Transformada Rápida de Fourier. Cada columna de la gráfica corresponde a un factor de la matriz; son requeridos arreglos verticales de N puntos ($N=2^p$). La utilización gráfica nos permite describir fácilmente los procesos de factorización de la matriz de tamaño N .

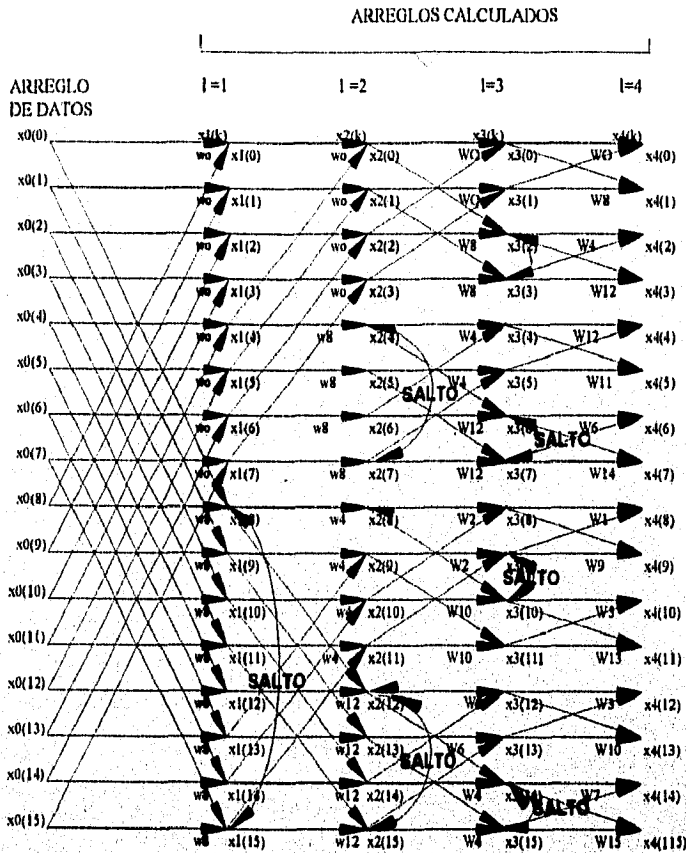


3.3.3.-Nodos Duales

Inspeccionando la siguiente figura, nos revela que en cada arreglo podemos encontrar dos nodos cuya ruta de transmisión se deriva del mismo par de nodos pero de un arreglo previo. Por ejemplo, el nodo $x_1(0)$ y $x_1(8)$ son computados en términos de los nodos $x_0(0)$ y $x_0(8)$. Nótese que los nodos $x_0(0)$ Y $x_0(8)$ no quedan dentro de los cálculos de cualquier otro nodo. Estos nodos los definimos como *par de nodos duales*.

Como el cálculo de un par de nodos duales es independiente de los demás, es posible desarrollarlo en forma computacional. Obsérvese en la fig. que simultáneamente se puede calcular $x_1(0)$ y $x_1(8)$

en términos de $x_0(0)$ y $x_0(8)$ y retornar el resultado en localidades de almacenamiento previamente ocupadas por $x_0(0)$ y $x_0(8)$.



Veamos ahora el espacio entre un par de nodos duales. Con respecto a la fig. anterior, primero, en el arreglo $l=1$, hay un par de nodos duales, es decir $x_1(0)$; $x_1(8)$ es separado por $k=8=N/2^1$. En el arreglo $l=2$, hay un par de nodos duales, es decir $x_2(8)$, $x_2(12)$, es separado por $k=4=N/2^2$. De forma similar, un par de nodos duales aparece en $x_3(4)$ y $x_3(6)$, en el arreglo $l=3$, es separado por $k=2=N/2^3=N/2^4$. Generalizando este resultado, observamos que el espaciamento entre nodos duales en un arreglo l está dado por $N/2^l$. Así, si consideramos un nodo dual en particular $x_1(k)$, entonces su nodo es $x(k + N/2^l)$. Esta propiedad nos permite identificar fácilmente un par de nodos duales.

El cálculo de un par de nodos duales requiere sólo de una multiplicación. Para aclarar esto, considérese el nodo $x_2(8)$ y su dual $x_2(12)$, ilustrado en la fig. anterior. La ruta de transmisión del nodo $x_2(12)$ es multiplicada por W^4 y W^6 primordialmente a los nodos de entrada $x_2(8)$ y $x_2(12)$

respectivamente. Es importante notar que $W^4 = -W^{12}$ y sólo es requerida una multiplicación. donde el mismo dato $x(12)$ será multiplicado por estos términos. En general, si el factor de peso en un nodo es W^p , entonces el factor de peso de un nodo dual es $W^{p(N/2)}$, sólo es requerida una multiplicación en el cálculo de un par de nodos duales. El cálculo de cualquier par de nodos duales está dado por el par de ecuaciones :

$$x_i(k) = x_{i+1}(k) + W^p x_{i+1}(k + N/2^l)$$

$$x_i(k + N/2) = x_{i+1}(k) - W^p x_{i+1}(k + N/2^l) \dots (21)$$

En el cálculo de un arreglo, normalmente empezamos con $k=0$ y secuencialmente hacia abajo el arreglo. Como un estado previo, la dualidad de cualquier nodo en el mismo arreglo siempre es $N/2^l$ hacia abajo en el arreglo. Donde el espaciamiento es $N/2^l$ por lo que debemos saltar después de cada $n/2^l$ nodos. Para ilustrar este punto, consideremos el arreglo $l=2$ de la siguiente figura. Si empezamos con el nodo $k=0$, entonces el nodo dual se encuentra en $k=N/2^2=4$ que se puede verificar inspeccionando la figura. Recorriendo el arreglo, notamos que el nodo dual se encuentra siempre localizado debajo de la localidad 4 del arreglo hasta que se alcance el nodo 4. En este punto ya tenemos un conjunto de nodos previamente encontrados; esto es, estos nodos son los duales para los nodos $k=0, 1, 2$ y 3 . es necesario saltar sobre los nodos $k=4, 5, 6$, y 7 . Los nodos $8, 9, 10$ y 11 siguen con la idea original de los nodos duales encontrándose por debajo de la localidad 4 del arreglo. En general, si trabajamos de la parte de arriba hacia abajo del arreglo l , entonces calcularemos la ec. (21) para los primeros $N/2^l$ nodos, saltando el siguiente $N/2^l$, etc. Sabemos que tenemos que parar de saltar cuando el índice de un nodo sea mayor que $N-1$.

Determinando W debemos definir las propiedades de cada arreglo con excepción del valor de p . El valor de p es definido por : a) la escritura del índice k en binario que se forma con l bits, b) escalando o deslizando este número

3.3.4.- Diagrama de Flujo de la Transformada Rápida de Fourier

Se sabe de lo anterior que lo que primero computamos es el arreglo $l=1$ empezando con el nodo $k=0$ y trabajando hacia abajo del arreglo. En este nodo k , calculamos el par de ec. (21) donde p es determinado por el procedimiento descrito. Se continúa calculando el par de ec. (21) hasta que encontremos una región de nodos la cual será saltada. Saltamos sobre los nodos apropiados y continuamos hasta que se termine de calcular todo el arreglo. Entonces se procede a calcular los arreglos restantes usando el mismo procedimiento. Finalmente descifraremos el último arreglo para obtener el resultado deseado. A continuación explicaremos el diagrama de flujo de la Transformada Rápida de Fourier.

Fig. 1.- Describe los datos de entrada necesarios. Se asume que el vector $x_0(k)$ es complejo y su índice será $k=0, 1, \dots, N-1$. Si $x_0(k)$ es real, entonces la parte imaginaria debe ser cero. El número de puntos muestreados N debe satisfacer la relación $N=2^r$, donde r es un número real.

Fig. 2.- El parámetro l es el número de arreglos que deben ser considerados. Se empieza con el arreglo $l=1$. El espacio entre los nodos duales está dado por el parámetro $N2$; para el arreglo $l=1$, $N2=N/2$. El parámetro $NU1$ es el salto a la derecha requerido cuando se determina el valor de p en la ec. (21); $NU1$ es inicializado con $y-l$. El índice k del arreglo es inicializado con $k=0$; así trabajaremos de la parte de arriba a la parte de abajo del arreglo.

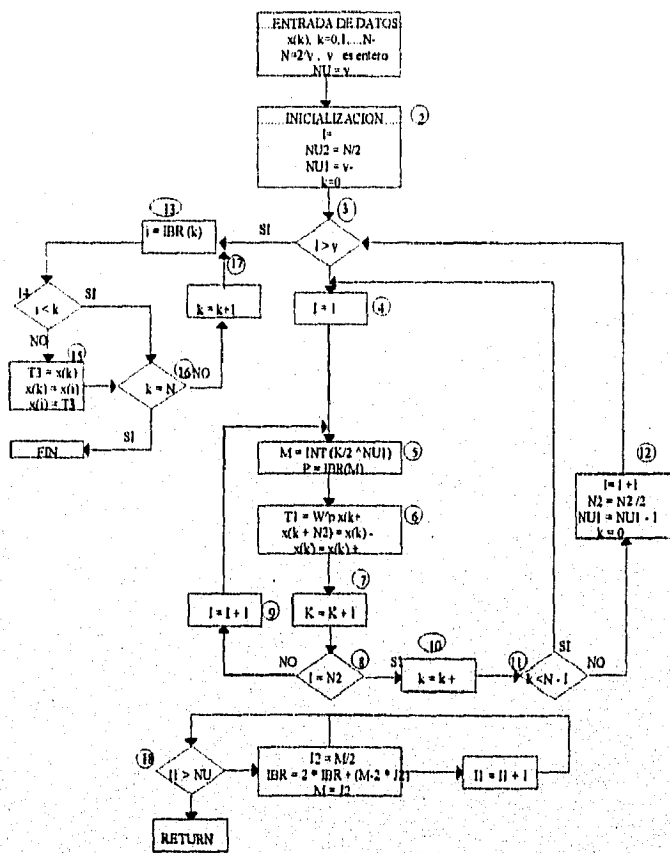
Fig. 3.- Verifica que el arreglo l , el cual es calculado, sea mayor a r . Si esto sucede entonces pasa a la fig. (14) para ir a obtener el resultado. Si no han sido calculados todos los arreglos, se procede a ir a la fig. (4).

Fig. (4) .- Pone el contador $l-1$. Este contador monitorea el número de pares de nodos duales que van a ser considerados. Recuerdese que es necesario ordenar los nodos de tal forma que los nodos que han sido considerados no se encuentren una segunda vez. El contador l es el control para determinar cuando el programa va a saltar.

Fig. (5) y (6) .- Aquí se desarrolla la ec. (21). Una vez que k e l han sido inicializados con 0 y l respectivamente, el nodo inicial considerado es el primer nodo del primer arreglo. Para determinar el factor p para este nodo, primero debemos de escalar (dividir en partes iguales) el número binario k para recorrerlos $r-l$ partes. Para realizar esto, debemos calcular el valor entero de $k/2^{(r-l)} = k/2^{(NU1)}$ y poner el resultado en M como se muestra en la fig. (5). De acuerdo a este procedimiento para determinar p , debemos reversar M bits, donde M está representado por $r=Nu$ bits. La función $IBR(M)$ denotada en la fig. (5) es una rutina especial bit inversión.

Fig. (6) es el cálculo de la ecuación (21). Calculamos el producto $W x(k + N2)$ y asignamos el resultado a una localidad temporal de almacenamiento. Después sumamos y restamos este término de acuerdo a la ec. (21). El resultado es un nodo dual de salida.

Cuando el procedimiento recorre el arreglo al siguiente nodo (fig. 7) k se incrementa en l . Para eludir al cálculo otra vez de un nodo que ha sido considerado previamente, verificamos (fig. 8) si l es igual a $N/2$, para el arreglo l el número de nodos que pueden ser considerados consecutivamente sin saltar alguno, es igual a $N/2=N2$. En la fig. (8) se determina esta condición. Si l no es igual a $N2$, entonces procedemos a recorrer en uno al arreglo e incrementar l como se muestra en la fig. (9). Recuerdese que ya habíamos incrementado k en la fig. (7). Las figuras (5) y (6) se repiten para los nuevos valores de k .



Si $l=N2$ (fig. 8), entonces tenemos que tomar un nodo previamente considerado. Sabemos que debemos saltar $N2$ nodos poniendo $k = k + N2$. Porque k ha sido previamente incrementado en l , en la fig. (7), es suficiente saltar los nodos previamente considerados incrementando k en $N2$.

Después de que se desarrolló el requerimiento computacional indicado en las figuras (5) y (6) para el nuevo nodo $k = k + N2$, debemos verificar primero en no exceder el tamaño del arreglo. Como se muestra en la fig. (11), si k es menor que $N-1$ (recuérdese que k es un índice que va de 0 a $N-1$) entonces inicializamos el contador l igual a 1 en la fig. (4) y repetimos en las figuras (5) y (6).

Si $k > N-1$ en la fig. (11), sabemos que tenemos que proceder con el siguiente arreglo. De aquí como se muestra en la fig. (12), l es incrementado con 1. El nuevo espaciamento $N2$ es simplemente $N2/2$ (recuérdese que el espaciamento es $N/2$). $NU1$ es decrementado con 1 ($NU1$ es igual a $v-1$), y k se inicializa con cero. Entonces verificaremos en la fig. (3) si todos los arreglos han sido calculados. Si esto sucede, procedemos a descifrar el resultado final. Estas operaciones son desarrolladas en las fig. (13) a la (17).

En la figura (13) se tiene $i = \text{IBR}(k)$. Otra vez utilizamos la función bit-inversión $\text{IBR}(k)$. Hay que recordar que para descifrar la Transformada Rápida de Fourier simplemente intercambiamos $x(k)$ y $x(i)$. Esta manipulación es desarrollada en la fig. (15). Para desarrollar la fig. (15) es necesario determinar, como se muestra en la fig. (14), si i es menor que k . Este paso es necesario para prohibir la alternancia de nodos previamente utilizados.

En la fig. (16) se determina cuando todos los nodos han sido descifrados y en la fig. (17) simplemente incrementamos k . En la fig. (18) describimos la lógica de la función bit-inversión $\text{IBR}(k)$; Tenemos que implementar el procedimiento bit-inversión, tomando en cuenta que las variables $x(k)$ y W^p son números complejos.

3.3.5.-Algoritmo de Decimación en el Tiempo.

Describamos la transformada discreta :

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{-kn} \quad \text{donde } W_N = e^{j2\pi/N}$$

y se asume que

$$N = 2^r$$

donde r es un entero. La suma puede factorizarse en dos partes :

$$X(k) = \sum_{n=0}^{N/2-1} x(n)W_N^{-kn} + \sum_{n=N/2}^{N-1} x(n)W_N^{-kn}$$

alternamente

$$X(k) = \sum_{n=0}^{N/2-1} x_{10}(n)W_N^{-2kn} + W_N^{-k} \sum_{n=0}^{N/2-1} x_{11}(n)W_N^{-2kn} \quad \dots(1)$$

donde

$$x_{10}(n) = x(2n) \quad x_{11}(n) = x(2n+1) \quad \dots(2)$$

para $0 \leq n \leq N/2-1$. Desde que :

$$W_N^{-2kn} = W_N^{-kn/N/2}$$

La ec. (1) puede ser expresada como :

$$X(k) = \sum_{n=0}^{N/2-1} x_{10}(n)W_N^{-kn/N/2} + W_N^{-k} \sum_{n=0}^{N/2-1} x_{11}(n)W_N^{-kn/N/2}$$

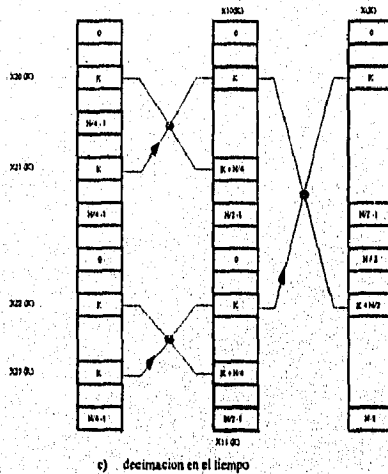
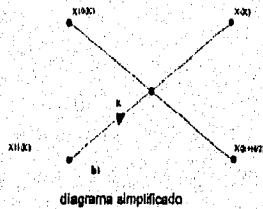
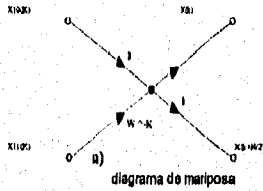
claramente

$$X(k) = X_{10}(k) + W_N^{-k} X_{11}(k) \quad \dots(3)$$

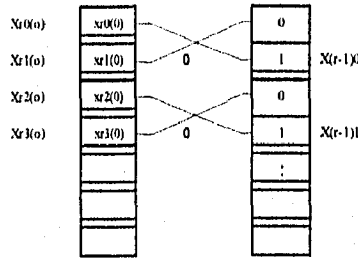
donde $x_{10}(k)$ y $x_{11}(k)$ son periódicos, cada uno con un periodo de $N/2$, y tenemos que :

$$X(k+N/2) = X_{10}(k+N/2) + W_N^{-(k+N/2)} X_{11}(k+N/2) = X_{10}(k) - W_N^{-k} X_{11}(k) \quad \dots(4)$$

Las ecuaciones 3 y 4 pueden ser representadas por una gráfica de mariposa, como la siguiente (a), donde la señal negativa en $\pm W_N^{-k}$ es pertinente al computar $X(k+N/2)$. Esta representación gráfica puede simplificarse en el diagrama (b) para una mejor conveniencia. Tenemos que recorrer un camino arduo para expresar el elemento N deseado en la Transformada Discreta de Fourier como una función de dos $(N/2)$ elementos de la Transformada Discreta de Fourier. Asumiendo que el valor de $X_{10}(k)$ y $X_{11}(k)$ están disponibles en sus correspondientes arreglos, los valores de $X(k)$ pueden ser realmente computados como una representación en la fig. (c).



algoritmo de la TRF * primero y segundo ciclo *



(d) n ciclo

$X_{10}(k)$ y $X_{11}(k)$ pueden ser ahora expresados en términos de $(N/4)$ -elementos de la Transformada Discreta de Fourier representando este ciclo. para $X_{10}(k)$ se puede escribir

$$\begin{aligned}
 X_{10}(k) &= \sum_{n=0}^{N/2-1} x_{10}(n) W_N^{kn} \\
 &= \sum_{n=0}^{N/4-1} x_{10}(2n) W_N^{2kn} + \sum_{n=0}^{N/4-1} x_{10}(2n+1) W_N^{-k(2n+1)} \\
 &= \sum_{n=0}^{N/4-1} x_{10}(n) W_{N/4}^{kn} + W_N^{-2k} \sum_{n=0}^{N/4-1} x_{11}(n) W_{N/4}^{-kn} \quad \dots(5)
 \end{aligned}$$

y, similar para $X_{11}(k)$

$$\begin{aligned}
 X_{11}(k) &= \sum_{n=0}^{N/2-1} x_{11}(n) W_N^{kn} \\
 &= \sum_{n=0}^{N/4-1} x_{11}(n) W_{N/4}^{kn} + W_N^{-2k} \sum_{n=0}^{N/4-1} x_{10}(n) W_{N/4}^{-kn} \quad \dots(6)
 \end{aligned}$$

donde :

$$\begin{aligned}
 X_{10}(n) &= X_{10}(2n) & X_{11}(n) &= X_{10}(2n+1) \\
 X_{11}(n) &= X_{11}(2n) & X_{10}(n) &= X_{11}(2n+1) \quad \dots(7)
 \end{aligned}$$

Para $0 \leq n \leq N/4 - 1$.

Consecuentemente de las ecuaciones (5 y 6)

$$X_{10}(k) = X_{20}(k) + W_N^{-2k} X_{21}(k) \quad X_{10}(k + N/4) = X_{20}(k) - W_N^{-2k} X_{21}(k)$$

$$X_{11}(k) = X_{22}(k) + W_N^{-2k} X_{23}(k) \quad X_{11}(k + N/4) = X_{22}(k) - W_N^{-2k} X_{23}(k)$$

Así si los valores de X_{20} , X_{21} , X_{22} y X_{23} están disponibles, Los valores de $X_{10}(k)$ y $X_{11}(k)$ y los valores de $X(k)$ pueden computarse como se muestra en la fig. (c).

En la misma forma, el m -ésimo ciclo del procedimiento es :

$$X_{(m-1)0}(k) = X_{m0}(k) + W_N^{-2m-k} X_{m1}(k)$$

$$X_{(m-1)0}(k+N/2m) = X_{m0}(k) - W_N^{-2m-k} X_{m1}(k)$$

$$X_{(m-1)1}(k) = X_{m2}(k) + W_N^{-2m-k} X_{m3}(k)$$

$$X_{(m-1)1}(k+N/2m) = X_{m2}(k) - W_N^{-2m-k} X_{m3}(k)$$

donde

$$\begin{aligned} X_{m0}(n) &= X_{(m-1)0}(2n) \\ X_{m1}(n) &= X_{(m-1)1}(2n+1) \\ X_{m2}(n) &= X_{(m-1)2}(2n) \\ X_{m3}(n) &= X_{(m-1)3}(2n+1) \end{aligned} \quad \dots (8)$$

para $0 \leq n \leq N/2^m - 1$.

Claramente el procedimiento termina con el r -ésimo ciclo ($N=2^r$), dados $x_0(n)$, $x_1(n)$,... se reduce un elemento de secuencia, en cuyo caso

$$X_{ri}(0) = x_{ri}(0)$$

para $i=0,1,\dots, N-1$, los valores de la penúltima Transformada Discreta de Fourier pueden ser obtenidos por la ecuación de arriba como :

$$X_{(r-1)0}(0) = x_{r0}(0) + W_N^0 x_{r1}(0)$$

$$X_{(r-1)0}(0) = x_{r0}(0) + W_N^0 x_{r1}(0)$$

$$X_{(r-1)1}(0) = x_{r1}(0) + W_N^0 x_{r2}(0)$$

$$X_{(r-1)}(0) = x_{r1}(0) + W^0_N x_{r2}(0)$$

Asumiendo la secuencia $\{x_{r0}(0), x_{r1}(0), \dots\}$, es posible que en un arreglo los valores de $X_{(r-1)}(k)$ para $i=0, 1, \dots$ puedan ser computados como en la fig. (d). Entonces los valores de $X_{(r-2)}(k)$, $X_{(r-3)}(k)$, ... pueden ser computados en una forma secuencial, para que después puedan ser obtenidos los valores de $X(k)$.

La única tarea que permanece pendiente en este punto es la de identificar los elementos $x_{r0}(0), x_{r1}(0), \dots$. Afortunadamente esto es fácil; como se puede mostrar ya que $x_{rp}(0)$ está dado por :

$$x_{rp}(0) = x(q)$$

donde q es la r -bit representación binaria de p inversa. Por ejemplo, si $N=16$, $r=4$, y :

$$x_{r0}(0) = x(0)$$

$$x_{r4}(0) = x(8)$$

$$x_{r2}(0) = x(4)$$

$$x_{r15}(0) = x(15)$$

En efecto, la secuencia $\{x_{r0}(0), x_{r4}(0), \dots\}$ es una versión reordenada de la secuencia $\{x(0), x(1), \dots\}$.

A manera de ejemplo construyamos el algoritmo de decimación en el tiempo para $N=8$

$$x_{10} = \{x(0), x(2), x(4), x(6)\}$$

$$x_{11} = \{x(1), x(3), x(5), x(7)\}$$

y la ecuación 7 da :

$$x_{20} = \{x(0), x(4)\}$$

$$x_{21} = \{x(1), x(6)\}$$

$$x_{22} = \{x(2), x(5)\}$$

$$x_{23} = \{x(3), x(7)\}$$

y finalmente de la ecuación 8

$$x_{30} = \{x(0)\}$$

$$x_{31} = \{x(4)\}$$

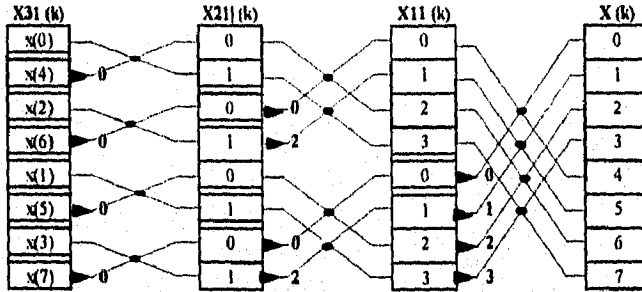
$$x_{32} = \{x(2)\}$$

$$\dots$$

$$x_{37} = \{x(7)\}$$

El algoritmo puede ser fácil de programar. Como se puede observar en la siguiente figura, una de las salidas de cada entrada son computadas, el elemento de entrada no necesita ser grande para ser

procesado y puede ser representado por su correspondiente salida. Cuando se procede de la misma forma de izquierda a derecha, al final de los cálculos el arreglo de las entradas contiene los elementos deseados por la Transformada Discreta de Fourier. En general, cada ciclo del algoritmo envuelve $N/2$ mariposas, como se muestra en la fig., y cada mariposa requiere una multiplicación (compleja). Dado que hay r ciclos computados y $r = \log_2 N$, el número total de multiplicaciones es $(N/2) \log_2 N$ opuesto al N^2 en el caso de la valuación directa.



Algoritmo de decimación en el tiempo de la Transformada Rápida de Fourier para $N=8$

3.3.6.- Algoritmo de Decimación en Frecuencia

Este procedimiento está dado por una secuencia de separación del índice presente y del viejo de los elementos. Este procedimiento es repetido en cada nueva secuencia hasta que es obtenido un elemento secuencial. Una alternativa del algoritmo de la Transformada Rápida de Fourier puede ser desarrollada dividiendo la secuencia dada a través de su punto medio y repetir el mismo procedimiento para cada secuencia resultante.

podemos escribir:

$$X(k) = \sum_{n=0}^{N/2-1} x(n)W_N^{-kn} + \sum_{n=N/2}^{N-1} x(n)W_N^{-kn}$$

$$= \sum_{n=0}^{N/2-1} [x(n) + W_N^{kN/2}x(n + N/2)] W_N^{-kn}$$

y con $k=2k$ ó $2k+1$

$$X(2k) = \sum_{n=0}^{N/2-1} x_{10}(n)W_{N/2}^{-kn}$$

$$X(2k+1) = \sum_{n=0}^{N/2-1} x_{11}(n)W_{N/2}^{-kn}$$

donde $x_{10}(n) = x(n) + x(n + N/2) \quad \dots(1)$

$x_{11}(n) = [x(n) - x(n + N/2)]W_N^{-kn} \quad \dots(2)$

para $0 \leq n \leq N/2-1$. Así el nuevo y los viejos valores de $X(k)$ están dados por las Transformadas Discretas de Fourier de $x_{10}(n)$ y $x_{11}(n)$, respectivamente. Asumiendo que los valores de $x(n)$ son almacenados secuencialmente en un arreglo, los valores de $x_{10}(n)$ y $x_{11}(n)$ pueden ser computados como se ilustra en la siguiente figura, donde la mano izquierda de la mariposa representa a la ec. (1) y (2).

El mismo ciclo ahora puede aplicarse a $x_{10}(n)$ y $x_{11}(n)$. Para $x_{10}(n)$

$$X(2k) = \sum_{n=0}^{N/4-1} [x_{10}(n) + W_N^{-kN/2} x_{10}(n + N/4)] W_{N/2}^{-kn}$$

y de forma similar para $x_{11}(n)$

$$X(2k+1) = \sum_{n=0}^{N/4-1} [x_{11}(n) + W_N^{-kN/2} x_{11}(n + N/4)] W_{N/2}^{-kn}$$

de aquí con $k=2k$ ó $2k+1$

$$X(4k) = \sum_{n=0}^{N/4-1} x_{20}(n) W_{N/4}^{-kn}$$

$$X(4k+2) = \sum_{n=0}^{N/4-1} x_{21}(n) W_{N/4}^{-kn}$$

$$X(4k+2) = \sum_{n=0}^{N/4-1} x_{22}(n) W_{N/4}^{-kn}$$

$$X(4k+2) = \sum_{n=0}^{N/4-1} x_{23}(n) W_{N/4}^{-kn}$$

donde $x_{20}(n) = x_{10}(n) + x_{10}(n+N/4)$

$$x_{21}(n) = [x_{10}(n) - x_{10}(n+N/4)] W^{-2n}$$

$$x_{22}(n) = x_{11}(n) + x_{11}(n+N/4)$$

$$x_{23}(n) = [x_{11}(n) - x_{11}(n+N/4)] W^{-2n}$$

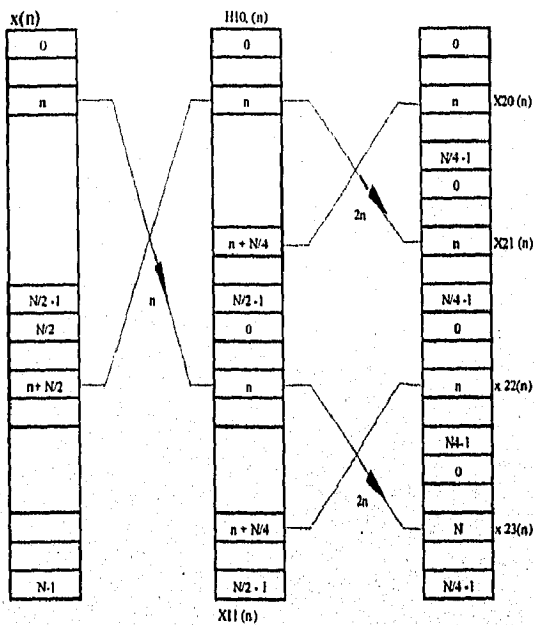


fig. (a)

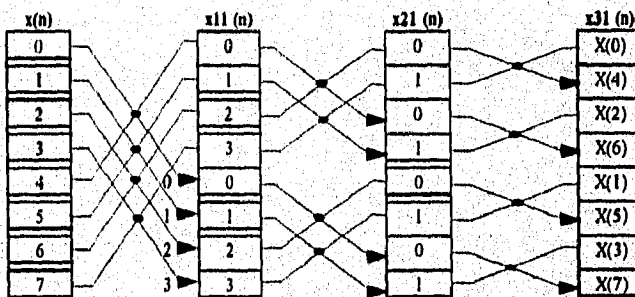


fig. b

Algoritmo de decimación en el tiempo de la Transformada Rápida de Fourier para $N=8$

para $0 \leq n \leq (N/4-1)$. Los valores de $x_{10}(n)$, $x_{21}(n)$, ... pueden ser computados como en la fig. (a). La Transformada Discreta de Fourier de cada una de estas secuencias da un cuarto del valor de $X(k)$.

De igual forma, el ciclo énesimo

$$X(2k) = \sum_{n=0}^{N_1-1} x_{m0}(n) W_{N_1}^{-kn}$$

$$X(2k+2) = \sum_{n=0}^{N_1-1} x_{m1}(n) W_{N_1}^{-kn}$$

donde $N_1 = N/2^m$ y

$$x_{m0}(n) = x_{(m-1)0}(n) + x_{(m-1)0}(n + N/2^m)$$

$$x_{m1}(n) = \{x_{(m-1)0}(n) - x_{(m-1)0}(n + N/2^m)\} W_{N_1}^{2^{m-1}n}$$

para $0 \leq n \leq N/2^m - 1$

En el caso previo, el procedimiento termina cuando $m=r$, al mismo tiempo $xr0(n)$, $xr1(n)$, ... reduce a cada uno de los elementos de secuencia, dando el valor deseado por la Transformada Discreta de Fourier.

$$X(0) = x_{r0}(0)$$

$$X(2) = x_{r1}(0)$$

El algoritmo completo para $N=8$ es ilustrado en la fig. (b). Como podemos observar, los valores de $X(k)$ aparecen en el arreglo desordenado de la salida. Se observa que, puede ser fácil reordenar lo realizado, restando la representación binaria r -bit de la localización indexada al final del cálculo. La ventaja de este algoritmo es que los valores de $x(n)$ son localizados en el arreglo de entrada secuencial.

PROGRAMA DE COMPUTADORA

4.- PROGRAMA DE COMPUTADORA

4.1.- Programa de computadora.

A continuación se mencionara los procedimientos que se utilizaron para realizar el programa de computadora.

Funcion Eleva .-Dado que Pascal no cuenta con una función que eleve un numero a una potencia diferente a dos, se crea esta función con la cual se puede aproximar en forma casi exacta el calculo deseado.

Procedure Lectura .- En este procedimiento se introducen los valores a calcular, se condiciona que el numero de valores sea mayor a cero y menor a 1024 ya que es una condición del método (TRF).

Procedure Bit_Reverse .- Aquí se intercambian los renglones de la matriz de datos, el cual es un punto clave para el desarrollo de la Transformada Rápida de Fourier, con lo cual se realizan menor numero de operaciones que con la tradicional Transformada Discreta de Fourier.

Procedure Calcula .- En este procedimiento se realiza lo que es el calculo de la transformada, el cual representa la eficiencia y moderna aproximación de la Transformada Discreta de Fourier.

Procedure Limpia .- Limpia la matriz de resultados, para que no contenga basura al momento de utilizarse.

Procedure Filtro .- Calcula el Filtro después de haberse calculado la Transformada Rápida de Fourier, el cual es un filtro de Primer Grado de tipo Retraso Rectangular.

Program Fourier;

Uses Crt;

Var

R1,R2,I,Ia,Ib,Ic,I1,I2 : Integer;

L,M,N,K: Integer;

Ray,C1,C2,Pi,T1,T2,W,W1,A,A1,C3,C4,C5,B,B1,K1: Real;

G1,G2,F1:Array[0..1023] Of Real;

Function Eleva(R,S:Real): Real;

Var

F:Real;

Begin

F:=Ln(R);

Eleva:=Exp(S*F)

End;

Procedure Lectura;

Begin

N:=0;

Repeat

Clrscr;

Gotoxy(5,2);Write('Numero De Puntos');

Read(N);

Until ((N>0) Or (N<1024));

K:=3;

Gotoxy(10,4);Write('Entre Los Valores');

For I:=0 To N-1 Do

Begin

Write('G[I,']');Read(G1[I]);

Write('G1[I,']');Readln(G2[I]);

End;

End;

Procedure Bit_Reverse;

Begin

L:=1;

For I:=1 To N-1 Do

Begin

If (I<L) Then

Begin

T1:=G1[L-1];

T2:=G2[L-1];

G1[L-1]:=G1[I-1];

G2[L-1]:=G2[I-1];

G1[I-1]:=T1;

G2[I-1]:=T2;

```

End;
M:=N Div(2);
If (M<L) Then
Begin
  While (M<L) Do
  Begin
    L:=L-M;
    M:=M Div (2);
  End;
End;
L:=L+M;
End;
End:

```

```

Procedure Calcula;
Begin
  Pi:=3.141592;
  W:=Cos(2*Pi/N);
  W1:=-Sin(2*Pi/N);
  For Ic :=1 To K Do
  Begin
    R1:=Round(Eleva(2,K-Ic));
    For Ib:=1 To R1 Do
    Begin
      R2:=Round(Eleva(2,Ic-1));
      For Ia:=1 To R2 Do
      Begin
        I1:=(Ia-1)+(Ib-1)*Round(Eleva(2,Ic));
        I2:=(Ia-1)+(Ib-1)*Round(Eleva(2,Ic))+Round(Eleva(2,Ic-1));
        A:=G1[I1];
        A1:=G2[I1];
        B:=G1[I2];
        B1:=G2[I2];
        C1:=(Ia-1)*Eleva(2,K-Ic);
        G1[I1]:=A+B*Eleva(W,C1);
        G2[I1]:=A1+B1*(-Eleva(-W1,C1));
        G1[I2]:=A-B*Eleva(W,C1);
        G2[I2]:=A1-B1*(-Eleva(-W1,C1));
      End;
    End;
  End;
End;
End:

```

```

Procedure Limpia;
Begin
  For I:=0 To 1023 Do
  F1[I]:=0;
End;

```



```
Procedure Filtro;  
Begin  
  Limpia;  
  For I:= 1 To N Do  
    Begin  
       $F1[I]:=F1[I-1]+(G1[I-1]-F1[I-1])*0.1;$   
    End;  
End;
```

```
Begin  
  Lectura;  
  Bit Reverse;  
  Calcula;  
  Filtro;  
  Clrscr;  
  For I:=1 To N Do  
    Begin  
      Gotoxy(5,2+I);Write('F(',I,')=';F1[I]);  
    End;  
End.
```

4.2.- Ejemplo de Aplicación.

Evalue la Transformada Rapida de Fourier y el Filtro Pasa Bajas para la funcion retardo e^{-t} donde $t = 0.1, 0.2 \dots, 0.8$

Filtro Pasa Bajas a usar :

$$\text{Sal} = \text{Sals} + (\text{Ents} - \text{Sals}) * dt/\tau$$

CORRIDA

NUMERO DE PUNTOS : 8

VALORES

G[1] = 1.105
G[2] = 1.221
G[3] = 1.349
G[4] = 1.491
G[5] = 1.648
G[6] = 1.822
G[7] = 2.013
G[8] = 2.225

VALORES DE LA TRANSFORMADA

REALES

G[1] = 1.296
G[2] = -1.51
G[3] = -0.889
G[4] = -0.236
G[5] = -0.641
G[6] = -0.178
G[7] = -0.247
G[8] = -0.118

PROGRAMA

G[1] = 1.285
G[2] = -1.566
G[3] = -0.936
G[4] = -0.296
G[5] = -0.654
G[6] = -0.199
G[7] = -0.267
G[8] = -0.129

VALORES DEL FILTRO

REALES

G[1] = 1.276
G[2] = 1.001
G[3] = 0.8060
G[4] = 0.678
G[5] = 0.555

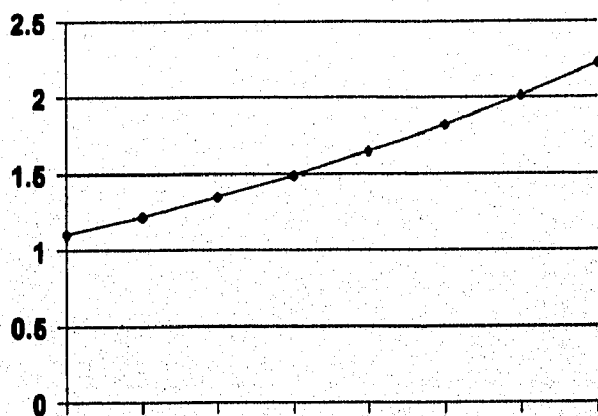
PROGRAMA

G[1] = 1.287
G[2] = 1.0027
G[3] = 0.8079
G[4] = 0.698
G[5] = 0.563

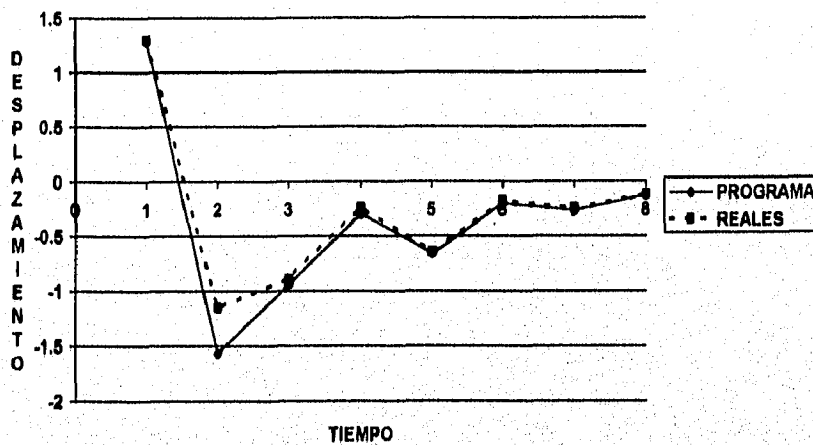
G[6] = 0.461
G[7] = 0.408
G[8] = 0.328

G[6] = 0.488
G[7] = 0.412
G[8] = 0.358

DATOS INICIALES

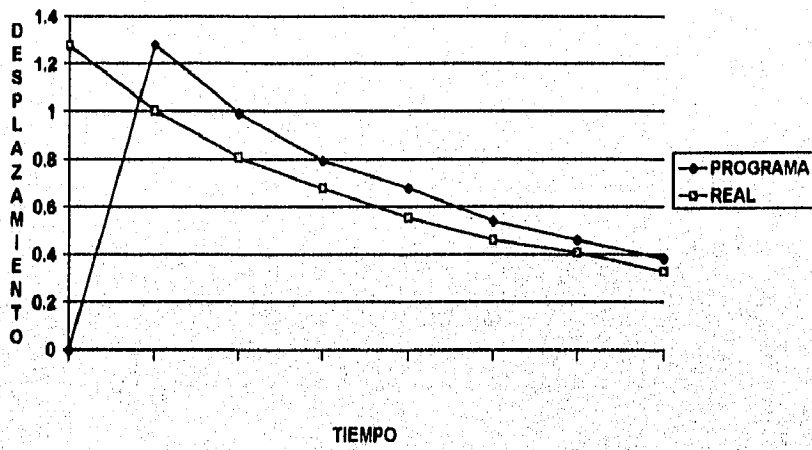


VALORES DE LA TRANSFORMADA



ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

VALORES DEL FILTRO



BIBLIOGRAFIA

BIBLIOGRAFIA :

Introducción a los Filtros Digitales
Jesús Barrios Romano
Universidad Autónoma Metropolitana, 1992

Digital Signal Processing Design
Andrew Bateman, Warren Yates
Computer Science Press, 1990

Digital Filters
R.W. Hamming
Prentice Hall, 1990

The Fourier Transform and its application
Ronald N. Bracewell
Mcgraw Hill, 1986

The Fast Fourier Transform
Brigham
Prentice hall, 1992

Digital Filters, Analysis and Design
L. S. Garrett
Prentice Hall, 1991

Analysis de Fourier
Hwei P. Hsu
Addison Wesley Iberoamericana, 1990