

31
2ej.



UNAM

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

CAMPUS ACATLAN

“ANÁLISIS DE UNA SEÑAL INTERFEROMÉTRICA DIGITALIZADA”

**TESIS QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN MATEMÁTICAS APLICADAS
Y COMPUTACIÓN.**

PRESENTA:

BRENDA EUGENIA OLIVOS FUENTES.

ASESOR: DR. ALEJANDRO MORALES MORI.

Acatlán, Estado de México

1998

**TESIS CON
FALLA DE ORIGEN**

Trabajo apoyado por CONACYT; proyecto 1095P-E.

254606



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.

AGRADECIMIENTOS

Deseo hacer patente mi agradecimiento a las personas e instituciones que hicieron posible la realización de este trabajo: al Dr. Alejandro Morales Mori por el entusiasmo con que me asesoró, al Fís. Luis Gutiérrez por sus valiosas sugerencias técnicas, al Instituto de Física UNAM (Laboratorio de Cuernavaca) en cuyas instalaciones se llevó a cabo la investigación, a CONACYT por el respaldo económico, al Ing. Pablo González Videgaray por su apoyo académico y moral durante mi estancia en la Universidad y a mis hermanos Ulises y a Alejandro por la ayuda que me brindaron en la realización del trabajo escrito.

Por otro lado, quiero dar las gracias a mis sinodales por el tiempo que se tomaron para revisar esta tesis y por sus valiosas sugerencias: Lic. Ma. Carmen Villar Patiño, Ing. Pablo González Videgaray, Fís. Jorge Luis Suárez Madariaga y Fís. Manuel Valadez Rodríguez.

A la memoria de mi abuelita Doña Enriqueta Márquez.

A mis tías Trini, Vicky y Alejandra.

A mi tío Luis.

A Martha, a Alejandro, a Jandro y a Uli.

INDICE

Introducción	i
Capítulo 1.- Nociones de Interferometría.	1
1.1. La naturaleza ondulatoria de la luz.	1
1.1.1. Movimiento Ondulatorio.	2
1.1.2. Ondas de Luz.	10
1.2. Interferencia.	16
1.3. El Interferómetro de Michelson.	20
1.4. Análisis de una señal interferométrica.	23
Capítulo 2.- Manejo del Sistema de Adquisición de Datos.	26
2.1. Sistemas de Adquisición de Datos.	26
2.1.1. Características generales de los gabinetes multitarjetas.	27
2.1.2. El estándar CAMAC.	28
2.1.3. El CAMAC y las otras opciones.	37
2.2. Funcionamiento del sistema CAMAC que se utiliza en este trabajo.	38
2.2.1. El <i>Crate</i> .	38
2.2.2. Funcionamiento del RS-232 <i>Crate Controller</i> .	39
2.3. El CAMAC y el Interferómetro.	45
2.3.1. Configuración actual del Controlador del <i>Crate</i> .	45
2.3.2. Módulos Indispensables.	45
Capítulo 3.- Conexión de la estación de trabajo UNIX con el sistema de adquisición de datos.	47
3.1. UNIX y el lenguaje de programación C.	47
3.2. El manejo de dispositivos en UNIX (HP-UX).	48
3.3. Conexión física entre la computadora y el sistema de adquisición de datos.	54

3.4.	Conexión lógica entre la computadora y el controlador del sistema de adquisición de datos.	59
3.4.1.	El <i>driver</i> y el archivo especial.	59
3.4.2.	<i>Entry Points</i> y llamadas al sistema.	65
3.4.3.	La configuración del puerto y <i>termio.h</i> .	67
3.4.4.	Llamadas al sistema y funciones especiales para el manejo de dispositivos.	81
3.5.	Biblioteca de funciones para el intercambio de datos con el CAMAC.	87
3.5.1.	Las subrutinas del estándar.	87
3.5.2.	Biblioteca de funciones.	90
3.5.3.	Un primer programa: Control de temperatura.	93

Capítulo 4.- Determinación interferométrica de la amplitud de un oscilador armónico.

4.1.	Descripción del problema.	95
4.2.	Configuración del Equipo.	100
4.3.	Adquisición de datos.	108
4.4.	Proceso de datos.	111
4.5.	Resultados.	120

Conclusiones. 132

Apéndice 1. 135

Apéndice 2. 141

Apéndice 3. 143

Apéndice 4. 157

Apéndice 5. 171

Apéndice 6. 173

Referencias. 186

Bibliografía. 188

INTRODUCCION

No cabe duda que la Computación ha adquirido gran relevancia en el mundo actual, existen una gran variedad de campos en los cuales se aplica: la administración, las telecomunicaciones, la ingeniería, la medicina, la educación, las ciencias básicas, etc.

El presente trabajo se aboca precisamente a una de las aplicaciones de la computación al campo de las ciencias exactas como lo es la Física, concretamente al terreno de lo que se llama interferometría con láseres.

La interferometría es una técnica basada en el fenómeno de interferencia de la luz, su instrumento esencial es el interferómetro y destaca por permitir hacer mediciones sin alterar o influir significativamente el fenómeno estudiado. Constituye un método muy importante en la física experimental y tiene aplicaciones en muchas otras ramas de la ciencia. Es necesario mencionar la importancia de la interferometría en la metrología, en la holografía, en la radioastronomía, en el estudio y medición de longitudes de onda, en la espectroscopía, etc.

El propósito de esta tesis, es realizar el análisis de una señal interferométrica digitalizada, es decir, se trata de conectar las variables físicas que se obtienen de un sistema interferométrico a una computadora que las procesará; teniendo como intermediario a un sistema de adquisición de datos CAMAC (*Computer Automated Measurement and Control*). La aplicación que se eligió fué la determinación de la amplitud del movimiento de un oscilador mecánico; haciendo uso de técnicas interferométricas.

El papel del sistema de adquisición de datos es determinante ya que permite muestrear las variables físicas involucradas en forma rápida, eficiente, controlada y con mínima supervisión humana; lo que se traduce en cierto grado de automatización del experimento. El uso del CAMAC y de la computación facilita la realización de experimentos con alta precisión reduciendo al máximo los errores humanos involucrados.

En suma, las pretensiones de este trabajo, incluyen demostrar la facilidad con que pueden hacerse mediciones interferométricas haciendo uso de un sistema de adquisición de datos controlado por una estación de trabajo UNIX y al mismo tiempo hacer una aportación en terrenos de la interferometría. El trabajo se ha organizado de la manera que se describe a continuación.

En el capítulo 1 se introduce al lector en los conceptos fundamentales del comportamiento de la luz y las nociones básicas de interferometría que constituyen el marco teórico sobre el que se asienta este trabajo. Así, se presentan conceptos como: reflexión, refracción, difracción, interferencia, sin olvidar el tratamiento formal de esta última con sus implicaciones en el campo de la interferometría.

En el capítulo 2 se presenta al sistema de adquisición de datos que se utilizó; su funcionamiento general, su configuración y sus limitaciones.

En el capítulo 3 se describe la conexión física y lógica de la estación de trabajo UNIX con el sistema de adquisición de datos CAMAC. Se describe el cableado necesario, el papel de los manejadores de dispositivos (*drivers*) y de los archivos especiales, de la configuración del *kernel* y de las llamadas al sistema para el manejo de dispositivos en UNIX. Se incluye también la “biblioteca” de funciones que se desarrolló en lenguaje C para lograr la configuración del puerto de comunicaciones de la computadora, la comunicación con el CAMAC con el formato específico que éste requiere, así como el procesamiento de las respuestas CAMAC.

En el capítulo 4 se plantea y desarrolla la técnica a usar para la aplicación elegida. Se

describen los papeles que desempeñaron el sistema interferométrico, el sistema de adquisición de datos y la computadora, en el análisis del movimiento del oscilador mecánico. Así mismo se presenta el *software* desarrollado para la adquisición y proceso de datos, y los resultados obtenidos.

Cabe mencionar que a lo largo de este trabajo se utilizaron los términos anglosajones propios de la jerga computacional. En muchos casos se decidió no hacer la traducción al español; ya sea porque se observó que los términos han tenido una aceptación generalizada o bien porque no es común el uso de la traducción en la literatura al respecto; lo que dificultaría en un momento dado la búsqueda de referencias.

Esta investigación forma parte del proyecto CONACYT 1095P-E : “Calibración interferométrica por efecto optogalvánico” que se lleva a cabo en el Instituto de Física de la UNAM, laboratorio Cuernavaca. Se espera que este trabajo, además de ser de utilidad para los investigadores en el laboratorio de láseres, sirva de alguna manera a los estudiantes de los últimos semestres de la carrera de Matemáticas Aplicadas y Computación mostrándoles la perspectiva de otros usos de la computación y resaltando el carácter multidisciplinario de la carrera.

CAPITULO 1. NOCIONES DE INTERFEROMETRIA

1.1.- La naturaleza ondulatoria de la luz.

La Optica es la parte de la Física que estudia los fenómenos luminosos (y en general la radiación electromagnética). Comenzó a desarrollarse desde tiempos muy antiguos en el afán del hombre por explicarse qué es lo que perciben sus ojos. Desde la antigüedad, con los filósofos griegos en el siglo III a.c. , hasta el siglo XVII con los científicos anteriores a Newton, muchos hombres generaron información acerca de las propiedades de la luz (como la reflexión y la refracción); sin embargo, una incógnita prevaleció en las mentes de los estudiosos: ¿qué es la luz?

Newton formuló un modelo corpuscular, es decir, para él la luz estaba constituida por pequeñas partículas o “corpúsculos” emitidos por la fuente luminosa.

Huygens, contemporáneo de Newton, objetó esa teoría y planteó un modelo ondulatorio, es decir, para él los cuerpos luminosos originaban vibraciones u ondas que se desplazaban hasta el ojo del observador.

Muchos estudios y experimentos fueron necesarios para completar la teoría acerca de la naturaleza ondulatoria de la luz (actualmente se acepta una doble naturaleza). Fue necesaria la medición de la velocidad de la luz en diversos medios y en el vacío (los primeros intentos de medir la velocidad de la luz fueron efectuados en el s. XVII, finalmente en 1983 el valor de la velocidad de la luz en el vacío fue definido), el desarrollo de la teoría electromagnética de Maxwell y su confirmación experimental realizada por Hertz (s. XIX) e incluso el advenimiento de la Teoría de la Relatividad de Einstein (1905) para terminar de aseverar: la luz es una onda electromagnética que se propaga a través del espacio aún vacío; y en su carácter de onda presenta reflexión, refracción, difracción e interferencia.

La interferometría es una técnica basada en el fenómeno de interferencia de la luz; constituye un método muy importante en la física experimental y tiene aplicaciones en muchas otras ramas de la ciencia.

En términos generales, puede decirse que la interferometría se encarga de hacer mediciones de longitud muy precisas haciendo uso de instrumentos llamados interferómetros. Permite medir igualmente grandes distancias que pequeños desplazamientos y, como se verá después, tiene la gran ventaja de hacer mediciones sin alterar o influir significativamente el fenómeno estudiado.

Debido a que este trabajo constituye en parte una modesta aportación en terrenos de interferometría, se hace necesario presentar un marco teórico básico que permita la comprensión del fenómeno de interferencia.

1.1.1.- Movimiento ondulatorio.

CONCEPTOS FUNDAMENTALES

Una **onda** puede definirse como la propagación de una perturbación en el espacio. Por la naturaleza de la perturbación pueden distinguirse dos tipos de ondas: ondas mecánicas y ondas electromagnéticas.

Ondas mecánicas.- Fueron observadas y estudiadas desde tiempos antiguos, requieren de un medio deformable o elástico para viajar; es decir, la propagación de la perturbación se da en un espacio no vacío tal como agua (ondas de agua), aire (ondas sonoras), material elástico o resortes. Se originan cuando cierta parte del medio se desplaza de su posición normal. Pueden ser transversales o longitudinales.

Ondas electromagnéticas.- Fueron estudiadas y reproducidas apenas el siglo pasado, no requieren de un medio para viajar; es decir, la propagación de la perturbación se puede dar en un medio acuoso, gaseoso y aún vacío. En forma general puede decirse que se originan en la interacción entre los campos eléctrico y magnético. En el espacio libre, son siempre transversales.

Ambos tipos de ondas (mecánicas y electromagnéticas) tienen una misma descripción matemática. De hecho, una onda mecánica simple que se mueve a lo largo de una cuerda tiene muchas propiedades en común con una onda de luz, como se indicará más adelante. Es por ello, y por simplicidad, que se explicarán en términos de ondas mecánicas los conceptos que siguen.

Por la forma en que se relaciona la dirección del movimiento de las partículas de materia con la dirección de propagación de la onda, puede distinguirse entre ondas transversales y ondas longitudinales.

Onda transversal.- El movimiento de las partículas es perpendicular a la dirección de propagación de la onda misma (Fig. 1.1).

Onda longitudinal.- El movimiento de las partículas es de vaivén a lo largo de la dirección de propagación (Fig. 1.2).

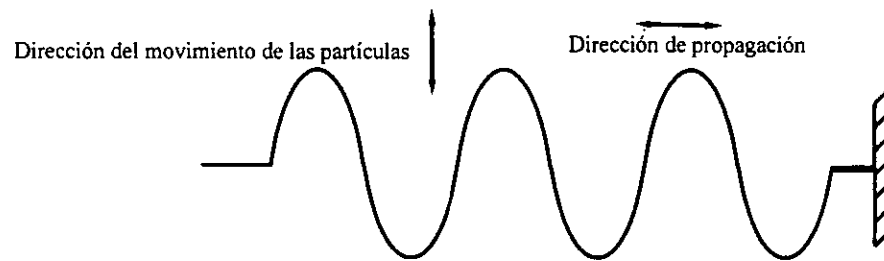


Fig. 1.1. Onda Transversal.

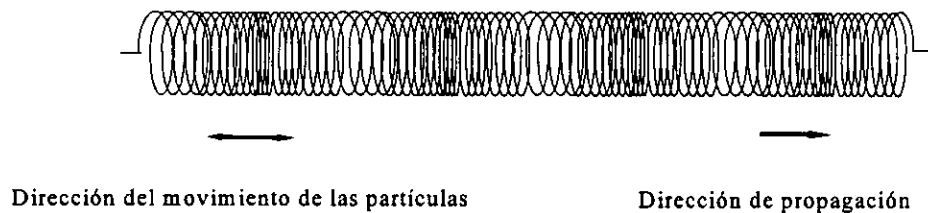


Fig. 1.2. Onda Longitudinal.

Ahora bien, es posible observar un **pulso** cuando una única perturbación es la que se propaga; cada partícula permanece en reposo hasta que el pulso llega hasta ella, luego se mueve

por un tiempo corto y finalmente vuelve al reposo (Fig. 1.3).

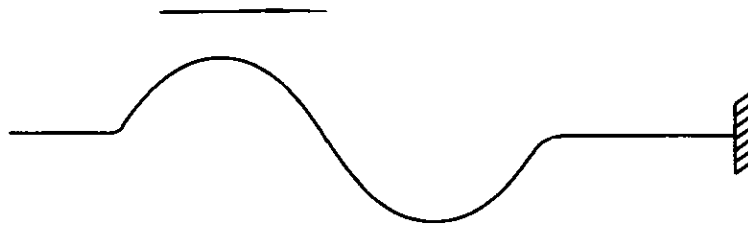


Fig. 1.3. Pulso.

También puede observarse un **tren de ondas** cuando son una serie de perturbaciones las que se propagan, si el movimiento es periódico estará observándose un **tren de ondas periódico**. El caso más sencillo de una onda periódica es una onda armónica (Fig. 1.4).



Fig. 1.4. Tren de ondas periódico.

Cualquier línea que indique la dirección del movimiento de la onda se llama **rayo**.

Se conoce como **fase** a una determinada localización de la onda (de la onda, no de las partículas del medio) (Fig. 1.5).

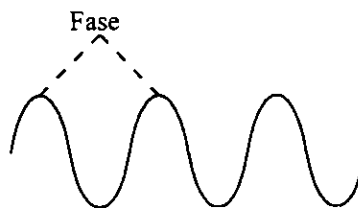


Fig. 1.5. Fase de onda.

Por la dimensión en que se propaguen, pueden ser **ondas unidimensionales** (una perturbación en una cuerda), **ondas bidimensionales** (una perturbación en un estanque) u **ondas tridimensionales** (el sonido) (Fig. 1.6).

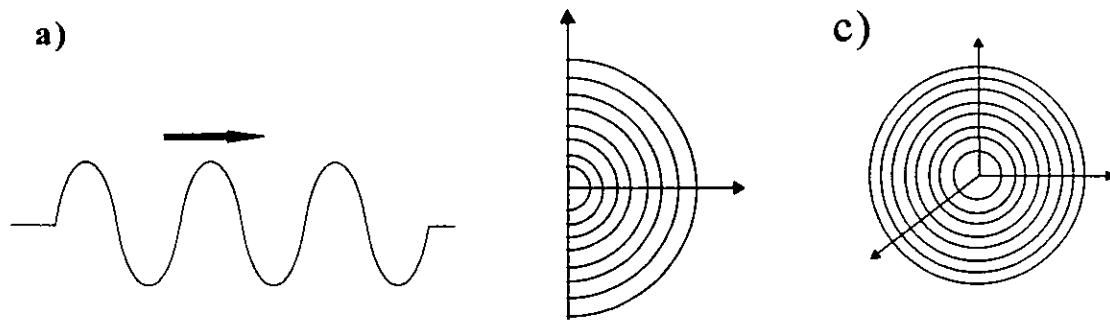


Fig. 1.6. (a) Onda unidimensional . (b) Onda bidimensional. (c) Onda tridimensional.

Una **onda plana**, es una onda tridimensional en la que las perturbaciones viajan en una sola dirección (Fig. 1.7).

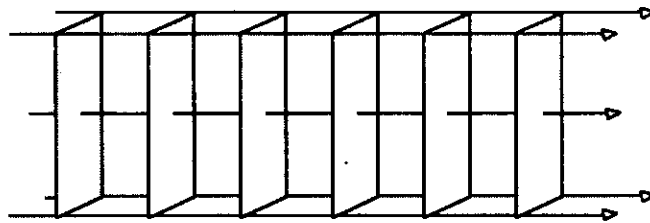


Fig. 1.7. Onda plana.

Una **onda esférica** es una onda tridimensional en la que la perturbación se propaga hacia afuera en todas direcciones desde una fuente puntual de ondas (Fig. 1.8).

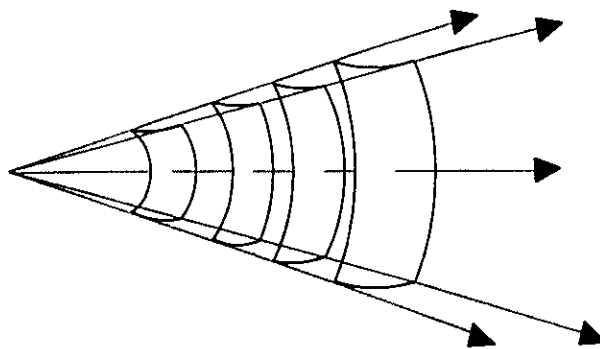


Fig. 1.8. Onda esférica.

DESCRIPCION MATEMATICA DE LAS ONDAS

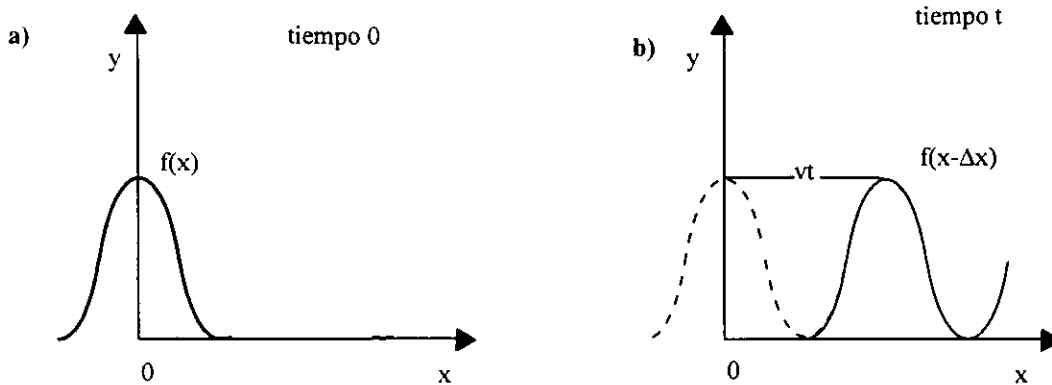


Fig. 1.9.(a)Pulso en $t=0$. (b)Pulso en $t=t$.

Tomando el caso de una onda transversal mecánica unidimensional (tal como una perturbación en una cuerda estirada) y suponiendo condiciones tales que la onda no pierde su forma mientras viaja. Supóngase primero un pulso que se mueve en dirección x positiva con una velocidad constante v (Fig. 1.9.a). En un tiempo t más tarde, el pulso se ha movido una distancia vt (pues si $v=\Delta x/t$ entonces $\Delta x=vt$) (Fig. 1.9.b)

La coordenada y indica el desplazamiento transversal de una partícula en el medio, depende tanto de la posición x como del tiempo t , así

$$y(x,t)=f(x) \text{ en } t=0.$$

La función f describe la forma de la onda en $t=0$ misma que se supone no cambia a lo largo de t (no cambia al viajar la onda). En este caso la onda que ha viajado en dirección x positiva se describe como

$$y(x,t)=f(x-\Delta x) \text{ en } t=t,$$

o bien como

$$y(x,t)=f(x-vt) \text{ en } t=t.$$

Es decir, la función $f(x-vt)$ tiene la misma forma relativa al punto $x=vt$ en el tiempo t que la función $f(x)$ la tiene al punto $x=0$ en $t=0$; precisamente ésto es lo que da la característica de movimiento de la onda (la coordenada en y es la misma en tanto crezcan tanto x como t : ocurre

un desplazamiento en el tiempo).

Ahora bien, si la onda se mueve en dirección x negativa, v se reemplaza por $-v$; se obtiene

$$y(x,t)=f(x+vt);$$

la onda se mueve hacia la izquierda.

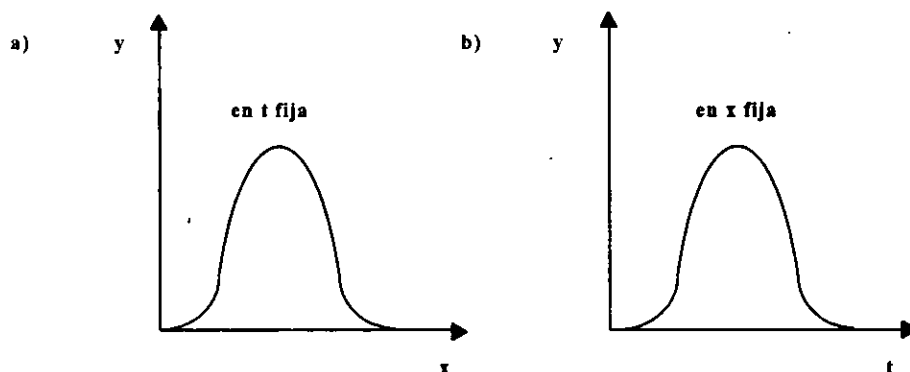


Fig. 1.10.(a)Gráfica de $y(x,t)$ con t fija. (b)Gráfica de $y(x,t)$ con x fija.

La función $y(x,t)$ contiene la descripción completa de la forma de la onda y de su movimiento. En cualquier tiempo determinado t_1 la función $y(x,t_1)$ da a y en función de x , lo que define a una curva, esta curva representa la forma real de la onda en ese tiempo (Fig. 1.10.a).

También puede ser de interés el movimiento particular de una partícula en el medio, es decir, ahora se fija la coordenada x en x_1 . Entonces la función $y(x_1,t)$ da la coordenada y de ese punto en función del tiempo (Fig. 1.10.b).

Es de especial interés para este trabajo presentar la descripción de ondas armónicas.

Una **onda armónica** es aquella que puede representarse por funciones del seno y del coseno. Así, por ejemplo, si se tiene en el tiempo $t=0$ un tren de ondas transversales dado por

$$y(x,0)=Y_m \operatorname{sen} \frac{2\pi x}{\lambda};$$

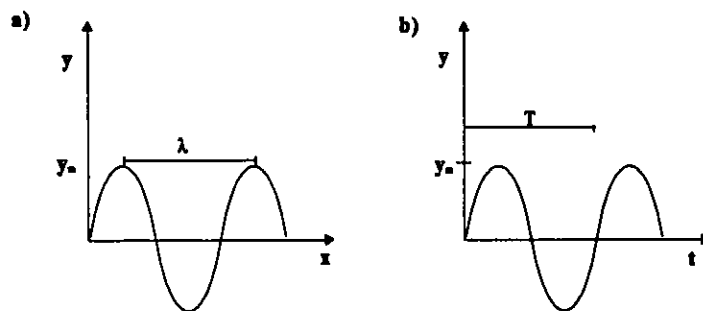


Fig. 1.11.(a) Longitud de Onda (b) Periodo.

el desplazamiento máximo Y_m se llama **amplitud** de la curva seno, el símbolo λ representa la longitud de onda del tren de ondas e indica la distancia entre dos puntos adyacentes de la onda que tengan la misma fase (fig.1.11a). Si la onda viaja en dirección x positiva con velocidad v , la ecuación de la onda es

$$y(x,t) = Y_m \operatorname{sen} \frac{2\pi(x-vt)}{\lambda},$$

es la ecuación de una onda viajera periódica, y tiene el mismo valor en $x, x+\lambda, x+2\lambda$, etc.

El **periodo** T de la onda es el tiempo necesario para que un punto en cualquier coordenada x efectúe un ciclo completo de movimiento transversal (fig. 1.11b). Durante este tiempo T , la onda viaja una distancia vT que debe corresponder a una longitud de onda λ de modo que $\lambda = vT$.

El inverso del periodo se llama **frecuencia** f de la onda; $f = 1/T$. La frecuencia tiene unidades de ciclos por segundo o Hertz (Hz); de modo que la onda también puede expresarse como

$$y(x,t) = Y_m \operatorname{sen} 2\pi \left(\frac{x}{\lambda} - \frac{t}{T} \right);$$

así, es claro que y tiene también el mismo valor en los tiempos $t, t+T, t+2T$, etc.

En la literatura, se encuentran definidas dos cantidades, el número de onda k y la frecuencia angular ω .

$$k=2\pi/\lambda ,$$

$$\omega=2\pi/T=2\pi f ,$$

ambas son cantidades angulares y sus unidades implican radianes.

Integrando estas definiciones en la ecuación de la onda se tiene que

$$y(x,t)=Y_m \text{sen}(kx-\omega t) ,$$

y si la onda seno viaja en dirección x negativa se describe por

$$y(x,t)=Y_m \text{sen}(kx+\omega t) .$$

En general puede decirse que la velocidad v cumple con que

$$v=\lambda f=\lambda/T=\omega/k .$$

Hasta ahora se ha supuesto que el desplazamiento y es cero en la posición $x=0$, $t=0$. Esto no necesariamente es así. La expresión general para una onda senoidal que viaje en dirección x positiva es

$$y(x,t)=Y_m \text{sen}(kx-\omega t-\phi) ;$$

más formalmente, la **fase** de la onda se define como el argumento del seno (en este caso $kx-\omega t-\phi$).

Se dice que dos ondas con la misma fase (o con fases que difieran en cualquier múltiplo entero de 2π) están “en fase”; ejecutan el mismo movimiento en el mismo tiempo.

El ángulo ϕ se llama **constante de fase**. La constante de fase no afecta la forma de la onda; únicamente mueve a la onda hacia adelante o hacia atrás en el espacio o en el tiempo como una condición inicial. De esa manera, la ecuación

$$y(x,t)=Y_m \text{sen}\left[k\left(x-\frac{\phi}{k} \right) - \omega t \right]$$

mueve a la onda en el espacio, mientras que la ecuación

$$y(x,t)=Y_m \text{sen}\left[kx - \omega \left(t+\frac{\phi}{\omega} \right) \right]$$

mueve a la onda en el tiempo.

Hasta aquí se ha trabajado con descripciones de ondas “teóricas” en las que la forma de la onda no cambia con el tiempo; el medio no permite pérdidas por fricción considerables. También se ha descrito una onda periódica sumamente simple: la onda senoidal pura. En la práctica, es usual que la forma de las ondas sea más compleja, pueden ser periódicas pero no senoidales, como las ondas cuadradas para transporte de información. Asimismo, las

características del medio pueden afectar la forma de la onda.

Afortunadamente para este trabajo, la luz es una onda transversal que se describe en términos de una función seno y según las condiciones que se usarán puede decirse que no es dispersiva (no cambia al viajar).

SUPERPOSICION DE ONDAS

Cuando se observan dos o más ondas viajando en forma simultánea por la misma región del espacio, independientemente entre sí, se dice que se superponen. El **principio de superposición** postula que, cuando varias ondas se combinan en un punto, el desplazamiento de cualquier partícula en un tiempo dado es la suma vectorial de los desplazamientos que produciría cada onda individualmente.

Así por ejemplo, dos ondas viajando simultáneamente a lo largo de la misma cuerda tensada y con sus desplazamientos individuales descritos por $y_1(x,t)$ (ver fig. 1.12.a) y $y_2(x,t)$ (ver fig. 1.12.b) resultan en un desplazamiento de la cuerda dado por

$$y(x,t) = y_1(x,t) + y_2(x,t), \quad (1.1)$$

que en este caso es la suma algebraica (pues van sobre la misma cuerda) (Fig. 1.12.c).

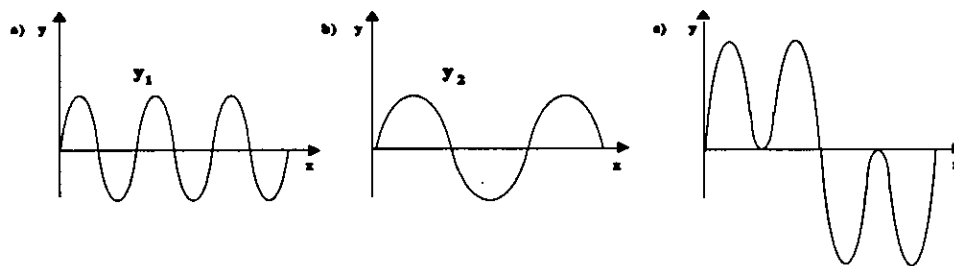


Fig. 1.12. (a) Gráfica de $y_1(x,t)$ (b) Gráfica de $y_2(x,t)$ (c) Gráfica de $y(x,t) = y_1(x,t) + y_2(x,t)$

Este principio es de suma importancia para entender el fenómeno de interferencia.

1.1.2.-Ondas de Luz

En 1865, Maxwell planteó un conjunto de ecuaciones (ecuaciones de Maxwell) con el que describe los fenómenos electromagnéticos. Según él, se produce un campo magnético en el

espacio vacío si existe un campo eléctrico variable y viceversa; el resultado neto final de los campos interactuantes es una onda de campos eléctrico y magnético que puede propagarse por el espacio: una onda electromagnética. Los campos eléctrico y magnético son perpendiculares entre sí, así como a la dirección del movimiento; además están en fase. Una onda electromagnética es pues transversal (Fig. 1.13).

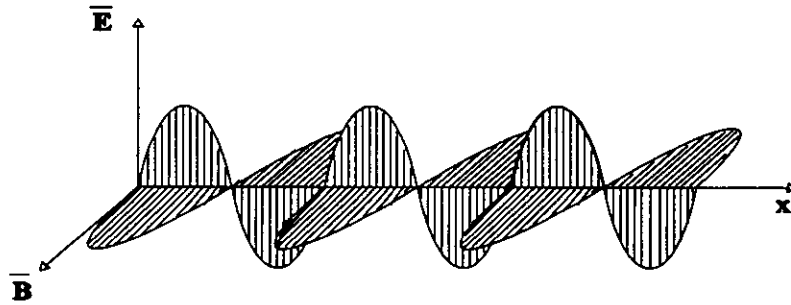


Fig. 1.13. Onda Electromagnética.

Como se dijo con anterioridad, una onda mecánica simple que se mueve a lo largo de una cuerda tiene muchas propiedades en común con una onda electromagnética. El movimiento de la partícula en la cuerda tiene su análogo en la variación del campo eléctrico o del campo magnético, variación que se da en dirección perpendicular a la dirección de propagación.

Maxwell demostró que cada componente del campo eléctrico y magnético obedece a la ecuación de onda y que juntos resultan en una onda electromagnética que viaja a través del espacio vacío a la velocidad

$$c = 1/\sqrt{\epsilon_0\mu_0} = 3 \times 10^8 \text{ m/s}$$

donde $\mu_0 = 4\pi \times 10^{-7} \text{ Tm/A}$ es la constante de permeabilidad en el vacío y $\epsilon_0 = 8.85418782 \times 10^{-12} \text{ C}^2/\text{Nm}^2$ es la constante de permitividad en el vacío. Este valor de c coincidía con las mediciones hechas sobre la velocidad de la luz. Ello corroboró que la luz es una onda electromagnética.

Así la luz, al igual que toda onda electromagnética, puede describirse como una onda por medio de la ecuación de la magnitud de campo eléctrico

$$E(x,t)=E_m \text{sen}(kx-\omega t-\phi) , \text{ con } k/\omega=c ; \quad (1.2)$$

donde c es la velocidad de la luz en el vacío actualmente definida como $c=299\,792\,458$ m/s.

También se puede describir por la ecuación del campo magnético B pero por regla general se usa E más que B ya que los efectos sobre el ojo humano y sobre varios detectores de luz es más significativa.

Es muy importante hacer notar que la descripción matemática anterior, misma que seguirá usándose a lo largo de este trabajo, es válida únicamente en ondas luminosas linealmente polarizadas; es decir, aquellas cuyo campo E permanece en una dirección fija al propagarse la onda. En fuentes luminosas ordinarias, los átomos se comportan independientemente y emiten ondas donde la dirección de E varía en forma aleatoria. Esta onda es pues transversal pero no polarizada. Se puede obtener luz linealmente polarizada por medio de un polarizador e incluso algunos tipos de laser pueden entregar luz polarizada.

Todas las ondas electromagnéticas se caracterizan porque su velocidad de propagación, en el vacío, es la constante c . La totalidad de las ondas electromagnéticas, clasificadas por su longitud de onda, se conoce con el nombre de espectro electromagnético.

Las ondas electromagnéticas se distinguen unas de otras por su longitud de onda y más definitivamente por su frecuencia. Se ha demostrado que la luz visible tiene frecuencias desde 4.0×10^{14} Hz y 7.5×10^{14} Hz lo que incluye los colores violeta, azul, verde, amarillo, naranja y rojo.

Las ondas se llaman monocromáticas siempre que contienen una sola frecuencia. Ninguna perturbación física real tiene esta forma aunque existen algunas aproximaciones en grado diferente que son llamadas cuasi-monocromáticas.

Al igual que cualquier otra forma de onda, una onda electromagnética puede transportar energía de un lugar a otro (luz o calor) es por ello que pueden interactuar con un detector (una película fotográfica, la retina, etc). La energía fluye en la dirección en que avanza la onda (en la dirección $\vec{E} \times \vec{B}$). Así pues, dos conceptos que cobrarán importancia son Potencia e Intensidad.

Se conoce como **potencia** P a la cantidad de flujo de energía por unidad de tiempo. La **intensidad** I está definida como la potencia promedio (en un periodo de movimiento) por unidad de área, transmitida a través de un área A normal a la dirección en que viaja la onda $I=P/A$; sus

unidades son potencia por unidad de área (watts/metro²). Esta, en el caso de ondas electromagnéticas está dada por:

$$I = E_m^2 / 2\mu_0 c, \quad (1.3)$$

donde c es la velocidad de la luz en el vacío, μ_0 es la constante de permeabilidad en el vacío y E_m es la amplitud de campo eléctrico. De modo que la intensidad I de una onda electromagnética es proporcional al cuadrado de la amplitud E_m de su campo eléctrico.

REFLEXION Y REFRACCION DE ONDAS DE LUZ

La reflexión y refracción de ondas de luz son fenómenos que han sido estudiados por la Óptica Geométrica la cual se limita a las situaciones donde los efectos resultantes de la naturaleza ondulatoria de la luz son despreciables. Esta simplificación equivale a exigir una propagación rectilínea en un medio homogéneo. Aún cuando a estos dos fenómenos se les puede dar un tratamiento siguiendo las ecuaciones de Maxwell, se ha demostrado que una descripción aproximada es completamente adecuada.

Así pues, las tres leyes básicas de la reflexión y de la refracción son las siguientes.

Cuando un haz de luz incide sobre la superficie de separación de dos medios (interfase) se divide en dos, uno de ellos regresa al mismo medio (se refleja) y el otro pasa al segundo medio (se transmite o refracta) (Fig. 1.14)

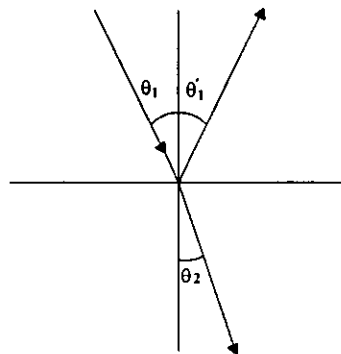


Fig. 1.14. Reflexión y refracción de un rayo.

Suponiendo un rayo incidente monocromático se cumple que:

1. Los rayos incidente, reflejado y transmitido están todos localizados en el mismo plano,

conocido como plano de incidencia, el cual es normal a la superficie de separación.

2. El ángulo de incidencia θ_1 es igual al ángulo de reflexión θ_1' .

3. Las direcciones de los rayos incidentes y transmitidos guardan una relación que viene dada por la ley de Snell:

$$n_1 \text{sen } \theta_1 = n_2 \text{sen } \theta_2 ,$$

donde n_1 es el índice de refracción del medio 1 y n_2 es el índice de refracción del medio 2. El índice de refracción del medio i está dado por la velocidad de la luz en el vacío sobre la velocidad de la luz en el medio i

$$n_i = c/v_i .$$

La frecuencia de una onda no se altera cuando pasa de un medio a otro, en cambio sí se altera la longitud de onda λ_i y consecuentemente la velocidad de la propagación.

La longitud de onda en un medio i λ_i está dada por

$$\lambda_i = \lambda / n_i ,$$

donde λ es la longitud de onda en el vacío.

La acción de los componentes que integran un sistema óptico (lentes, espejos, prismas) puede describirse por la óptica geométrica, ésto es, por rayos de luz.

DIFRACCION E INTERFERENCIA DE ONDAS DE LUZ

Ambos fenómenos han sido estudiados por la llamada Óptica Física, la cual, en contraste con la Óptica geométrica, se ocupa de los efectos resultantes de la naturaleza ondulatoria de la luz.

La característica principal de la difracción es la desviación, respecto de la propagación rectilínea, que surge cuando una onda es obstruida de alguna manera; por ejemplo cuando se hace pasar a la onda por un orificio (fig. 1.15).

Evidentemente, la difracción resulta considerable sólo cuando el tamaño del obstáculo o del orificio es del orden de la longitud de onda.

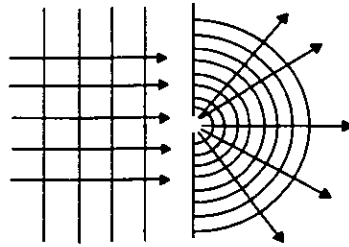


Fig. 1.15. Difracción.

Por otro lado, la interferencia se presenta cuando en una región del espacio inciden dos o más movimientos ondulatorios, luz en este caso. Según el principio de superposición y dependiendo de las fases relativas en que se encuentren las ondas, puede pasar que exista un refuerzo casi completo en la amplitud de las dos ondas (interferencia constructiva) (Fig. 1.16)

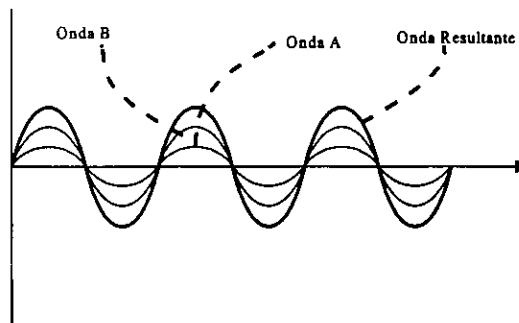


Fig. 1.16. Interferencia constructiva.

También puede ocurrir una cancelación casi completa en cada punto y la amplitud resultante estará cerca de cero (interferencia destructiva)(Fig. 1.17).

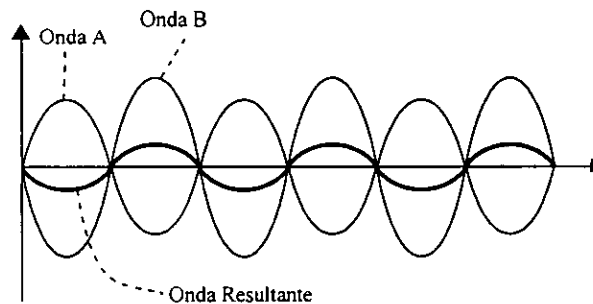


Fig. 1.17. Interferencia destructiva.

Como el concepto de interferencia es muy importante para este trabajo, se tratará más ampliamente.

1.2.- Interferencia.

Cuando dos o más ondas se combinan en un punto determinado se dice que interfieren. Supóngase que se tienen dos haces de luz monocromática linealmente polarizada, idénticos excepto porque no están en fase y tienen amplitudes diferentes, descritos por

$$E_1(x,t) = A \sin(kx - \omega t - \phi_1),$$

$$E_2(x,t) = B \sin(kx - \omega t - \phi_2)$$

donde A y B son las amplitudes correspondientes a cada haz.

Describir la onda resultante, que por el principio de superposición es $E(x,t) = E_1(x,t) + E_2(x,t)$; resulta más sencillo si se definen las cantidades α y β :

$$\alpha = 1/2 [2(kx - \omega t) - \phi_1 - \phi_2],$$

$$\beta = \frac{\phi_2 - \phi_1}{2}.$$

Las cantidades α y β cumplen con que

$$kx - \omega t - \phi_1 = \alpha + \beta$$

$$kx - \omega t - \phi_2 = \alpha - \beta.$$

La ecuación $E(x,t) = A \sin(kx - \omega t - \phi_1) + B \sin(kx - \omega t - \phi_2)$ bien puede expresarse como

$$E(x,t) = \text{Im}(A e^{i(kx - \omega t - \phi_1)} + B e^{i(kx - \omega t - \phi_2)}),$$

sustituyendo los valores α y β , se tiene que

$$E(x,t) = \text{Im}(A e^{i(\alpha + \beta)} + B e^{i(\alpha - \beta)})$$

desarrollando:

$$E(x,t) = \text{Im}(e^{i\alpha} (A e^{i\beta} + B e^{-i\beta}))$$

$$E(x,t) = \text{Im}(e^{i\alpha} (A e^{i\beta} + A e^{-i\beta} + (B-A) e^{-i\beta}))$$

$$E(x,t) = \text{Im}(2A e^{i\alpha} \cos \beta + (B-A) e^{i(\alpha - \beta)})$$

$$E(x,t) = 2A \cos \alpha \cos \beta + (B-A) \sin(\alpha - \beta)$$

sustituyendo α y β por sus valores correspondientes, se tiene

$$E(x,t) = 2A \sin(kx - \omega t - (\phi_2 + \phi_1)/2) \cos 1/2(\phi_2 - \phi_1) + (B-A) \sin(kx - \omega t - \phi_2); \quad (1.4)$$

obsérvese que el último término de la expresión no depende de la diferencia entre las fases $\Delta\phi$ de las dos ondas, con $\Delta\phi=(\phi_2-\phi_1)$. Para efectos de este trabajo, el último término carece de importancia por las razones que se harán evidentes más adelante en este capítulo; es por ello que puede considerarse que la amplitud de la onda resultante E_0 es:

$$E_0=2 A |\cos(\Delta\phi/2)| \quad (1.5).$$

Es fácil observar que la onda resultante sigue siendo senoidal, conserva su valor λ y conserva su frecuencia. Si la diferencia de fase $\Delta\phi$ es muy pequeña, la amplitud resultante es casi el doble de la amplitud de la onda individual (interferencia constructiva), y si es cercana a π , la amplitud resultante es casi cero (interferencia destructiva).

Como la intensidad I es proporcional al cuadrado de la amplitud, puede verse la interferencia en términos de intensidad; así, cuando la intensidad neta es mayor que las intensidades individuales, se está hablando de interferencia constructiva. En cambio, cuando la intensidad neta es menor que las intensidades individuales, se está hablando de interferencia destructiva. El hecho de que la interferencia sea constructiva o destructiva depende de la diferencia de fase $\Delta\phi$ y está condicionada que ésta no cambie con el tiempo. Esta condición está relacionada con el concepto de coherencia.

Coherencia.-Una fuente coherente es aquella a la que las ondas abandonan guardando la misma relación de fase entre sí en todo momento. Dos haces son coherentes, ya sea que estén en fase o fuera de fase, siempre que mantengan todo el tiempo una relación de fase constante.

Una fuente coherente deberá producir variaciones completamente periódicas de E y de B con el tiempo y un tren de ondas infinito o de extensión muy grande (esta extensión será el límite de las distancias para las cuales la luz permanece coherente).

Actualmente se cuenta con fuentes de luz visible altamente coherentes: los láseres (láser, *light amplification through stimulated emission of radiation*). La longitud de coherencia de la luz láser puede ser de decenas o centenas de metros.

Cuando ondas coherentes interfieren, la intensidad de la onda combinada en cualquier punto en el espacio no cambia con el tiempo. Por lo general sólo se logra esta coherencia dividiendo la luz de una sola fuente en dos componentes y tratarlos como si fueran fuentes de luz

independientes.

Así fué como Young (1801) realizó el experimento de la doble rejilla y observó un patrón de interferencia llamado franjas de interferencia (franjas blancas y oscuras que corresponden a máximos y mínimos en la intensidad de la luz).

Un tren de ondas planas de luz (por ejemplo procedentes de un láser) incide sobre una barrera en la cual están cortadas dos rendijas angostas separadas por una distancia d , de modo que la luz se difracta e ilumina la pantalla (Fig. 1.18)

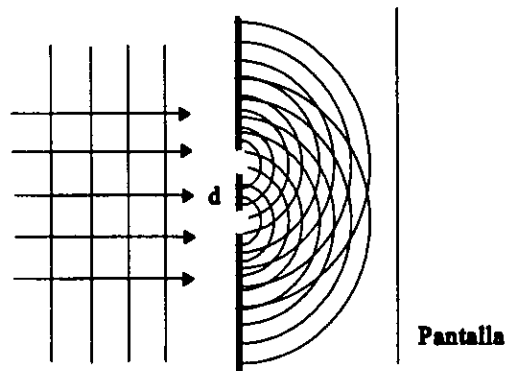


Fig. 1.18. Onda plana pasando por una doble rejilla.

Sobre la pantalla pueden apreciarse franjas de interferencia.

En tiempos de Young todavía no existían los láseres así que él creó una fuente de luz coherente al permitir que la luz solar incidiera sobre una abertura angosta adicional (Fig. 1.19)

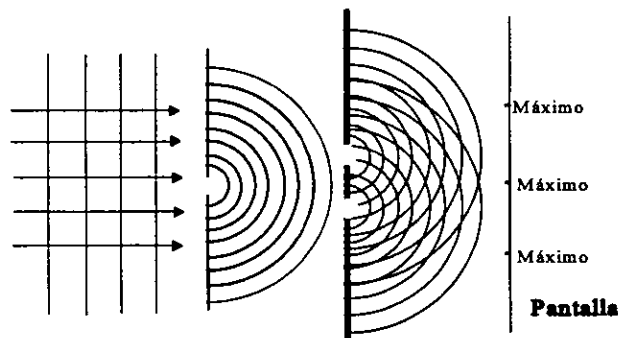


Fig. 1.19. Experimento de la doble rejilla.

Nótese que la interferencia está dada por la diferencia entre las trayectorias que recorre cada rayo que incide en la pantalla. La relación con la diferencia de fases es:

$$\Delta\phi/2 \pi = \Delta x/\lambda \quad (1.6)$$

donde Δx es la diferencia de trayectorias.

La coherencia del haz de láser ha dado por resultado un número de aplicaciones prácticas en las cuales el haz se divide en dos haces que recorren caminos diferentes y luego se hace que se recombinen, momento en que ocurre la interferencia.

Como ése será el caso que se atenderá, es preciso presentar la intensidad I en términos de la diferencia de fases y/o de la diferencia de trayectoria de los dos haces luminosos.

Si I_θ es la intensidad de la onda resultante e I_0 es la intensidad que produciría cada una de las ondas actuando por sí misma, entonces, según la ecuación (1.3)

$$\frac{I_\theta}{I_0} = \frac{E_\theta^2/2\mu_0 c}{E_0^2/2\mu_0 c}$$

$$\frac{I_\theta}{I_0} = \frac{E_\theta^2}{E_0^2}$$

sustituyendo E_θ por la ecuación (1.5)

$$\frac{I_\theta}{I_0} = \frac{(2E_0 \cos(\Delta\phi/2))^2}{E_0^2}$$

$$\frac{I_\theta}{I_0} = \frac{4 E_0^2 \cos^2(\Delta\phi/2)}{E_0^2}$$

por lo tanto

$$I_\theta = 4 I_0 \cos^2(\Delta\phi/2)$$

sustituyendo $\Delta\phi/2$ según la ecuación (1.6)

$$I_\theta = 4 I_0 \cos^2(\pi\Delta x/\lambda) \quad (1.7)$$

de modo que se observan máximos de intensidad cuando $\cos^2(\Delta\phi/2) = \cos^2(\pi\Delta x/\lambda) = 1$ y eso ocurre cuando

$$\Delta\phi = 2m\pi \text{ con } m=0, \pm 1, \pm 2, \dots$$

o bien cuando

$$\Delta x = m\lambda \text{ con } m=0, \pm 1, \pm 2, \dots$$

y los mínimos de la intensidad ocurren cuando $\cos^2(\Delta\phi/2) = \cos^2(\pi\Delta x/\lambda) = 0$ y eso ocurre cuando

$$\Delta\phi=(2m+1)\pi \quad \text{con } m=0,\pm 1,\pm 2,\dots$$

o bien cuando

$$\Delta x=(m+\frac{1}{2})\lambda \quad \text{con } m=0,\pm 1,\pm 2,\dots$$

(Ver figura 1.20)

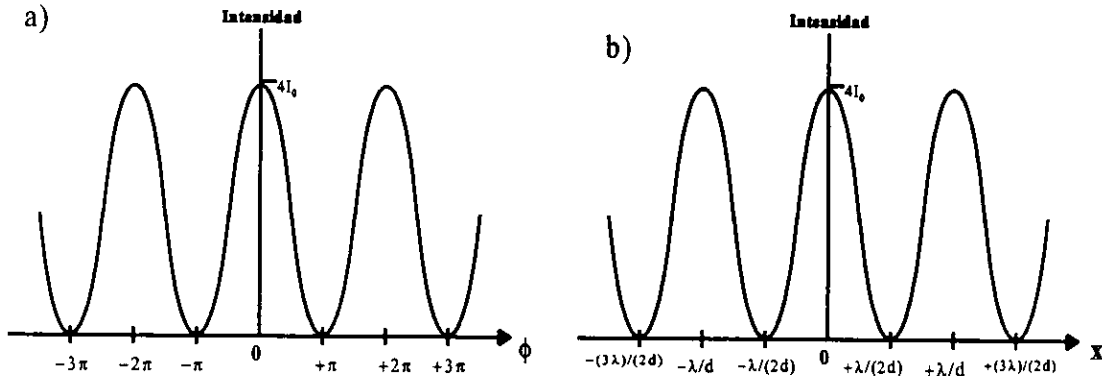


Fig. 1.20. a) Gráfica de interferencia respecto a ϕ . b) Gráfica de interferencia respecto a x .

1.3.- El interferómetro de Michelson

En términos generales puede afirmarse que un interferómetro es un dispositivo que, por medio de componentes ópticos (lentes, espejos, prismas), es capaz de producir franjas de interferencia a partir del haz que emite una fuente luminosa (radiación).

Un sistema interferométrico es aquel que entre sus elementos cuenta con al menos un interferómetro.

Como se dijo en un principio, la interferometría es una técnica cuyo instrumento esencial es el interferómetro. En algunas aplicaciones el interés está en conocer las características de la fuente misma, como la visibilidad de las franjas de interferencia que produce y su distribución (o posición); en otras aplicaciones lo que interesa es el estudio de un fenómeno ajeno a la fuente de

luz como en el caso de la holografía y la metrología. A continuación se mencionan algunas aplicaciones de la interferometría.

Estándares de longitud y medición de distancias.

Dado un haz de luz cuya frecuencia y longitud de onda es conocida, pueden medirse distancias en términos de longitudes de onda. En 1960 el metro patrón fué definido como 1650763.73 longitudes de onda de la luz rojo-anaranjada que emite el gas Kriptón⁸⁶. Este metro patrón fué bastante práctico y permitió hacer medidas con una precisión de una parte en 10^8 m. (Una cienmillonésima de metro). La actual definición del metro se basa en la unidad de tiempo (el segundo) y en el valor definido para la velocidad de la luz (1983); así casi cualquier línea láser puede devenir en un estandar de longitud.

Asimismo pueden medirse distancias grandes o pequeñas en términos de longitudes de onda. Pueden medirse longitudes de onda de láseres de menor precisión, para efectos de calibración por ejemplo. Es posible medir desplazamientos muy pequeños como los causados por dilatación (expansión térmica). La medición de la velocidad de la luz en diferentes medios y en el vacío ha sido una aplicación importante de la interferometría basada en la diferencia de trayectoria de los haces.

Análisis de superficies.

Por medio de técnicas interferométricas es posible hacer análisis de las irregularidades en superficies, puede detectarse si la superficie es plana, esférica, etc. Esta característica puede usarse en los terrenos de la óptica para evaluar lentes, con la ayuda de microscopios puede usarse también para examinar la textura de ciertos materiales en el área de la microtopografía, y para realizar actividades de alineación.

Holografía.

Se logra la reproducción tridimensional de imágenes, guardando un patrón de interferencia en una placa fotográfica (holograma) el cual, al ser iluminado de cierta manera reconstruye la imagen. Un observador la percibe como si el objeto todavía estuviera ahí.

Espectroscopía.

La espectroscopía es el estudio de la distribución de la radiación electromagnética; es decir, busca la separación de las radiaciones provenientes de una fuente luminosa en un espectro

de sus colores o longitudes de onda constituyentes. Métodos de interferencia se usan en aquellas regiones del espectro donde pueden emplearse detectores de potencia.

Radioastronomía.

La radioastronomía estudia el universo a través de las ondas de radiación que penetran en la atmósfera de la tierra; pueden usarse interferómetros como detectores de la radiación y una de las aplicaciones es la medición de diámetros estelares.

Por otro lado, existen diversos tipos de interferómetros. Por el número de haces que interfieren pueden ser de dos haces o de múltiples haces; el patrón de franjas resulta diferente en cada caso. Por el método usado para separar los haces de luz pueden ser de división de frente de onda o de división de amplitud; en el primer caso, se utilizan haces de luz que viajan en distintas direcciones pero que provienen de la misma fuente (como el experimento de la doble rejilla), en el segundo caso, la radiación es separada por medio de un divisor de haz.

Se conoce como interferómetro de Michelson a un tipo de interferómetro de división de amplitud; implica un divisor de haz y únicamente dos haces interfiriendo. Este es el tipo de interferómetro que se usará (Fig. 1.21).

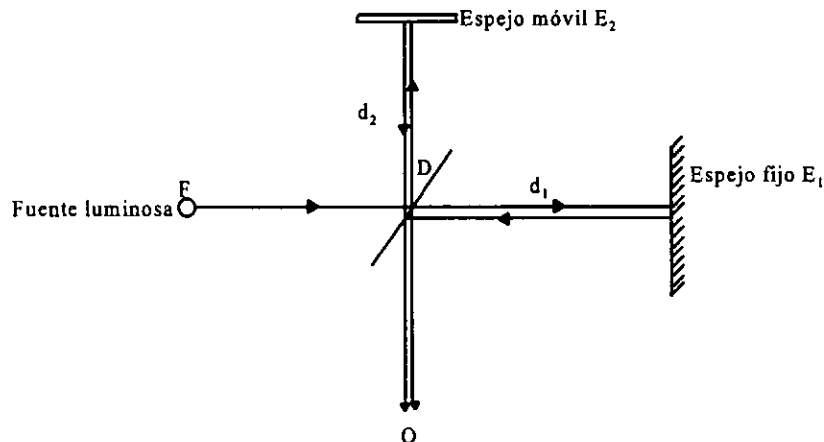


Fig. 1.21. Interferómetro de Michelson

PRINCIPIO BASICO DE FUNCIONAMIENTO

Un haz de luz monocromática proveniente del punto F de la fuente incide sobre un divisor de haz D, que puede ser un espejo semiplataado, cuya característica es que transmite la mitad de la luz y refleja la otra mitad. En D la luz se divide en dos ondas. Una continúa por transmisión hacia el espejo E_1 , la otra viaja por reflexión hacia E_2 . Las ondas se reflejan en cada uno de los espejos retornando a lo largo de sus direcciones de incidencia, entrando finalmente cada onda al ojo O del observador.

Si el espejo E_2 se mueve hacia atrás o hacia adelante, el efecto es un cambio en el patrón de interferencia ya que está produciéndose una diferencia de trayectorias Δx (con su equivalente $\Delta\phi$) Δx puede verse como

$$\Delta x = 2(d_2 - d_1)$$

donde d_2 es la distancia entre D y E_2 y d_1 es la distancia entre D y E_1 .

Por lo que se vió anteriormente, si $\Delta x = m\lambda$ con $m=0, \pm 1, \pm 2, \dots$ el patrón de franjas original se repite, es decir si se observaba un máximo de intensidad, volverá a observarse un máximo de intensidad después de que E_2 se desplazó Δx .

El interferómetro de Michelson debe su nombre a su inventor, el científico de origen polaco Albert Abraham Michelson (1852-1931); hombre que dedicó una buena parte de su vida en aclarar la incógnita acerca de la velocidad de la luz.

1.4.- Análisis de una señal interferométrica.

Cualquier intento de analizar una señal interferométrica, requiere en primera instancia de un detector de la energía electromagnética, éste puede ser la retina del ojo humano, una película fotográfica, una celda fotoeléctrica, en fin, algún medio para convertir la energía en información.

El sistema interferométrico que se usará en el presente trabajo será utilizado para medir la magnitud del desplazamiento del espejo móvil bajo ciertas condiciones (mismas que se especificarán en el capítulo 4). Se utilizará como fuente de luz linealmente polarizada, un láser de Helio-Neón cuya longitud de onda es conocida e igual a $\lambda=633$ nm. La configuración real del

sentido opuesto del haz original. Las placas L1 y L2 optimizan la polarización del haz luminoso.

Para detectar la señal, el divisor de haz BM3 desvía el haz hacia el fotodiodo PH1. El filtro SF2 mejora la señal que llega al fotodetector (fotodiodo).

Como es un problema que la luz láser regrese a su fuente, existe un polarizador P1 que la detiene.

Así pues, cuando la luz láser incide en el fotodiodo, éste produce una señal eléctrica que posteriormente es amplificada, ésta es una señal interferométrica producto de la recombinación de los dos haces. El amplificador elimina el componente de la señal correspondiente al segundo término de la ecuación (1.4), así pues, entrega una señal eléctrica que corresponde a la intensidad I descrita por la ecuación (1.7), adecuada a los requerimientos del sistema de adquisición de datos.

Aún cuando la disposición de los componentes ópticos es más complicada, es completamente equivalente al esquema presentado en la figura 1.21; tan sólo el ojo se reemplaza por un fotodiodo. En lo subsecuente y por simplicidad se hará referencia a aquel esquema toda vez que sea necesario.

Una vez que se tiene la señal amplificada, puede hacerse con ella lo que sea necesario; para contar franjas por ejemplo, es necesario contar máximos de intensidad. Esta es la base sobre la que se asienta este trabajo.

Esta, es una señal continua que puede ser la señal de entrada de diversos instrumentos para su análisis; puede hacerse una apreciación visual conectándola a un osciloscopio, puede conectarse a convertidores analógico-digitales o a contadores que detecten los cruces por cero de la señal.

Este trabajo propone hacer el análisis de la señal interferométrica a través de un equipo de cómputo, una estación de trabajo UNIX para ser precisos. La señal pasará por un sistema de adquisición de datos (del cual se hablará ampliamente en el siguiente capítulo) y de ahí se enviará a la computadora para su proceso. La aplicación que se ha elegido es la determinación de la amplitud del movimiento de un oscilador mecánico; es decir, interferometría aplicada a la determinación de distancias.

Se pretende demostrar la facilidad con que pueden hacerse mediciones interferométricas haciendo uso de un sistema de adquisición de datos controlado por una computadora bajo UNIX y también, por qué no decirlo, hacer una pequeña aportación en el campo de la interferometría.

CAPITULO 2.- MANEJO DEL SISTEMA DE ADQUISICION DE DATOS

2.1.- Sistemas de adquisición de datos.

En la industria, en los laboratorios científicos, en las universidades, se utilizan sistemas de adquisición de datos para convertir en información útil las señales que se derivan de sistemas físicos tales como vehículos, máquinas, estructuras o procesos físicos. Para el presente trabajo, por ejemplo, el sistema físico es el sistema interferométrico.

Puede ser deseable monitorear, medir y/o controlar una o más magnitudes involucradas en un sistema físico (como la temperatura, la presión, el voltaje, la fuerza, etc.), es por ello que el hombre ha construido, a lo largo de su historia, instrumentos que le auxilian en esta tarea.

La eficiencia y precisión con que se hagan las medidas tiene una gran importancia. En la industria, por ejemplo, pueden ser factores determinantes para definir la calidad de un producto o el nivel de desempeño de una máquina; en un laboratorio científico, son decisivos para el control y/o evaluación de experimentos.

Actualmente es posible hablar de tres opciones en sistemas de adquisición de datos, en donde las mediciones son realizadas por medio de:

- a) Instrumentos con los que el usuario interactúa, que trabajan individualmente e independientes entre sí (tales como voltímetros, osciloscopios o cronómetros). La adquisición de datos de esta forma realizada, es recomendable para aplicaciones de monitoreo o evaluación esporádica.
- b) Instrumentos conectables a una computadora (tarjetas) que trabajan individualmente e independientes entre sí. El usuario requiere de software que le permita interactuar con el (los) instrumento(s). La adquisición de datos realizada de esta forma, ofrece más ventajas que la anterior; permite aplicaciones de monitoreo, evaluación y/o control en forma más dedicada. Puede ser más o menos económico adquirir y mantener un sistema de adquisición de datos de esta naturaleza, pero el número de instrumentos se limita a los slots libres con que cuente la

computadora.

c) Instrumentos modulares conectables a un gabinete, capaces de trabajar conjuntamente y en colaboración. El gabinete se conecta a una computadora y el usuario puede programar las actividades de cada instrumento según sus necesidades. Un sistema de adquisición de datos de esta naturaleza, es caro de adquirir y de mantener pero es adecuado para aplicaciones tanto de monitoreo y evaluación como de control en forma dedicada y **constante**, que requieran alta precisión o resultados en tiempo real. Este sistema de adquisición de datos, no necesariamente está sujeto al desempeño, capacidad o limitaciones de un modelo de computadora específico.

En el presente trabajo, se utilizarán instrumentos modulares en un gabinete multitarjetas, es por ello que se les dedica este capítulo.

2.1.1.- Características generales de los gabinetes multitarjetas.

Los gabinetes multitarjetas, están constituídos de un soporte (también conocido con los nombres de *Crate* o Chasis) que cuenta con un bus de datos al que se le pueden conectar múltiples instrumentos en forma modular (con interfaz estándar), que provee una única fuente de poder, permite la transferencia de datos a través de un bus único (*Dataway*), y que puede conectarse a una computadora y/o a otro gabinete.

Un sistema de este tipo requiere de uno o varios controladores que no son otra cosa que módulos dedicados a manejar toda comunicación entre los instrumentos y el *dataway*, y también entre el *crate* y la computadora (u otro controlador externo); ver fig. 2.1.

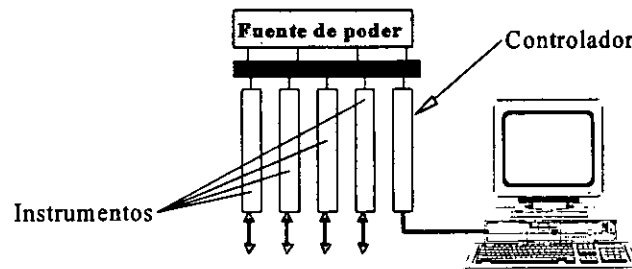


Fig. 2.1. Representación esquemática de un gabinete multitarjetas.

La aparición de este tipo de estándares (al principio de los años 70's) respondió a la necesidad de tener instrumentos compatibles (en laboratorios y organizaciones industriales) que trabajaran bajo una misma interfaz. Anteriormente, varios instrumentos se conectaban entre sí y con la computadora, a través de varias interfaces y cada uno usando su propia fuente de poder. El número de interfaces necesarias podía llegar a ser muy grande.

Con la utilización de un gabinete multitarjetas se estandarizan interfaces mecánicas y eléctricas y se puede lograr instrumentación modular capaz de ligar dispositivos (como sensores, contadores o convertidores de señal) con controladores digitales o computadoras. El uso de estándares permite que los productos sean ofrecidos por diversos fabricantes, hace posible configurar y/o reconfigurar el sistema según las necesidades específicas del usuario toda vez que lo requiera, y están especialmente diseñados para resolver necesidades de adquisición y control en tiempo real (a altas velocidades).

2.1.2.- El Estándar CAMAC

CAMAC (*Computer Automated Measurement and Control*) constituye un sistema modular de instrumentación, definido por el estándar ANSI/IEEE std 583-1982, con las características de gabinete multitarjetas mencionado [2.1]. No es el único que existe en el mercado, aunque fue uno de los primeros y llegó a ser muy popular. Actualmente ha sido superado por otros estándares mucho más poderosos y de mayor velocidad de transferencia como VXI (del que se hablará después).

Sin embargo, aquí se enfoca la atención hacia los sistemas CAMAC ya que se trabajará con uno de ellos para conectar el sistema físico (interferómetro) con la computadora.

A continuación se detallan las características de un sistema CAMAC. Un sistema CAMAC ha sido diseñado para consistir de un *Crate*, así como de instrumentos conectables o módulos. El sistema es usualmente conectado a una computadora en forma *on-line*; sin embargo el uso de la computadora es opcional.

EL CRATE.

El *CRATE* CAMAC es un gabinete o soporte estándar que cuenta con un bus de datos

Dataway, así como con 25 estaciones donde se insertan todos los módulos que compartirán el bus, cuenta con un sistema de ventilación y con una fuente de poder única (fig. 2.2). Conjuntos de *Crates* pueden ser interconectados en serie o en paralelo por medio de *highways* paralelos o seriales.

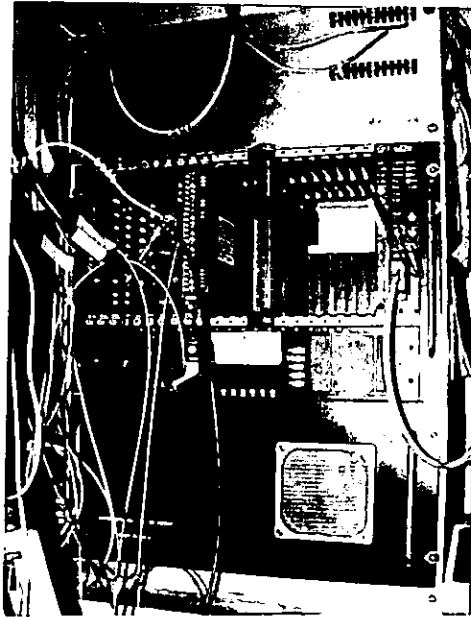


Fig. 2.2. El *Crate* CAMAC.

El *DATAWAY* es un bus paralelo consistente de líneas de señal y de energía que conforman un total de 86 líneas. Este bus, transportará los datos, las señales de control y la energía de todos los módulos. Así, 24 de las líneas se usarán en comandos de lectura y otras 24 para comandos de escritura, de modo que puede manejar palabras de hasta 24 bits en una simple transferencia de datos; el resto de las líneas transportan las señales de control y la energía. La comunicación entre los instrumentos a través del *dataway* es enteramente digital. El *dataway* es independiente del tipo de instrumentos que se le conectarán y de la computadora utilizada.

Las líneas del *dataway* caen en alguna de las siguientes cuatro categorías:

- 1) Líneas bus de todas las estaciones.- intercomunican todas las estaciones .
- 2) Líneas bus de las estaciones normales.- son las líneas de datos.
- 3) Líneas individuales.- son líneas individuales entre la estación de control y cada estación normal.

4) Contactos para conexiones irregulares.

Una operación típica del *dataway* involucra al menos 2 módulos, uno de los cuales actúa como controlador y el otro como módulo controlado; ambos reciben señales de algunas líneas del *dataway* y generan señales en otras de acuerdo con las especificaciones del estándar.

Las ESTACIONES son los 25 espacios en donde se insertan los módulos. Todas son idénticas excepto por la de la extrema derecha, la cual tiene el rol especial de estación de control; las líneas de datos en el *dataway* son accesibles al resto de las estaciones normales pero no para la estación de control.

Cada estación tiene guías superiores e inferiores para la entrada del módulo, así como un contacto al *dataway* (lo que hace que la comunicación de los instrumentos o módulos con el *dataway* sea enteramente digital).

LOS MODULOS.

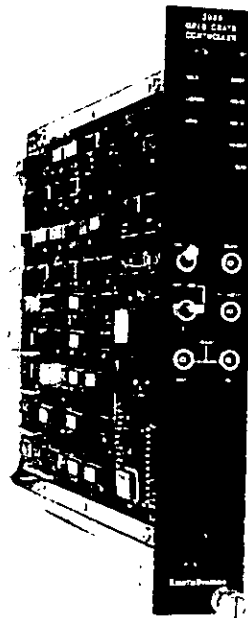
Son los instrumentos que se conectan al *crate*. Cada uno de ellos es tratado como una caja negra con la estandarización limitada a la configuración mecánica externa, los requerimientos de energía eléctrica y las entradas y salidas de señal, lo que hace posible la compatibilidad entre instrumentos y su intercambiabilidad (de una estación a otra) sin importar su diseño y funcionamiento interno. Es posible encontrar en el mercado una larga lista de instrumentos que siguen el estándar CAMAC, y más allá, es posible para los conocedores construir sus propios instrumentos CAMAC para cubrir necesidades específicas, únicamente apegándose al estándar (cuya utilización no requiere de licencia ni de ningún otro permiso)[2.2].

Los módulos consisten básicamente de un panel frontal con un tornillo fijador, correderas (o guías) superior e inferior que se deslizarán en las guías del *crate*, y un conector de 86 contactos que va a dar al *dataway* (fig. 2.3). Un módulo puede ocupar más de una estación y de ser así, tendrá más de un conjunto de guías y más de un conector.

En 23 estaciones del *crate* se conectarán instrumentos CAMAC y en las 2 de la extrema derecha se conectará el controlador.

Usualmente, un módulo está dividido en subsecciones independientes entre sí; con direcciones de acceso diferentes (*subaddress*). Así, un módulo con subsecciones puede verse

como un instrumento con varias entradas/salidas trabajando independientemente; por ejemplo, un módulo contador de pulsos puede tener 6 canales cada uno de los cuales atiende a una señal diferente.



Kernal Systems Model 3000
GPIB Crate Controller

Fig. 2.3. Ejemplo de módulo CAMAC.

El controlador es un módulo con características muy especiales; se encarga de manejar toda comunicación entre los instrumentos y el *dataway*, y también entre el *crate* y la computadora (u otro controlador externo). Ocupa 2 estaciones y son las de la extrema derecha en el *crate*. El controlador sirve también como ‘traductor’ del lenguaje del *dataway* al de la computadora.

El controlador es entonces la única unidad en el *crate* que puede habilitarse para comunicarse con la computadora. Esta comunicación puede hacerse de dos maneras:

Dedicada.- El controlador sirve de interfaz y llega a ser una extensión del bus I/O de la computadora específica.

No dedicada.- Permite versatilidad y puede ser conectada a cualquier tipo de computadora ya sea a través de una conexión serial o paralela.

Para cualquier operación en el *dataway* debe contarse con el módulo controlador y al menos otro módulo que será controlado. El controlador es el encargado de emitir las señales

necesarias en las líneas del *dataway* para la ejecución de comandos en un módulo específico y para realizar operaciones genéricas (en todos los módulos) tales como limpiar las líneas o inicializar los instrumentos. Las líneas del *dataway* transportarán datos (hasta 24 bits en paralelo) a través de sus líneas W *write* y de sus líneas R *read*; información de status será transportada en las líneas LAM *look at me*, B *busy*, Q *response*, y X *command accepted*; a saber:

Línea X *command accepted*.- El módulo genera X=1 en la línea X si reconoce que puede ejecutar el comando. X=0 indicará o bien un mal funcionamiento del módulo, o bien que no está equipado para realizar la operación.

Línea B *busy*.- El controlador genera B=1 en la línea B para indicar a toda unidad que está ejecutándose una operación.

Línea Q *response*.- El módulo genera una señal en esta línea para indicar el estado de alguna de sus características. La interpretación del estado de Q depende de la operación que esté ejecutándose y debe especificarse claramente en el manual de módulo.

Línea LAM *look at me*.- Según el estándar, cualquier módulo puede generar una señal en una línea individual llamada LAM que va directamente a la estación de control, para indicar que requiere atención; los módulos que ocupan más de una estación pueden indicar diferentes peticiones LAM.

Así, son dos tipos de operaciones las que se pueden realizar en el *dataway*: operaciones de comando y operaciones genéricas.

En las operaciones de comando, el controlador genera un comando que incluye el envío de señales en las líneas adecuadas con el objeto de especificar el módulo, la subsección del módulo (*subaddress*) y la función que se ejecutará.

Existe un código de función estándar compuesto por 32 funciones F0 a F31 y se subdivide funcionalmente en 3 grupos: operaciones de lectura *read*, operaciones de escritura *write* y operaciones de control.

Comandos *read*: F0 a F7 Involucran transferencias de datos y de información de status, desde un módulo al controlador a través de las líneas *read*, Q y X.

Comandos *write*: F16 a F23 Involucran transferencias de datos del controlador al módulo a través de las líneas *write*, y de información y status del módulo al controlador a través de las

líneas Q y X.

Comandos de control: F9 a F15 y F24 a F31 No hay transferencia de información en las líneas *read* o *write*; sin embargo, puede transmitirse información de status en la línea Q. Con estos comandos no se transfieren datos, en su lugar se consultan o se definen características del módulo.

A continuación se transcribe la tabla de códigos de función (*Function Codes*) que maneja el estándar CAMAC:

<u>Código</u>	<u>Función</u>
F0	Read Group 1 Register
F1	Read Group 2 Register
F2	Read and Clear Group 1 Register
F3	Read Complement of Group 1 Register
F4	Nonstandard
F5	Reserved
F6	Nonstandard
F7	Reserved
F8	Test Look-at-Me
F9	Clear Group 1 Register
F10	Clear Look-at-Me
F11	Clear Group 2 Register
F12	Nonstandard
F13	Reserved
F14	Nonstandard
F15	Reserved
F16	Overwrite Group 1 Register
F17	Overwrite Group 2 Register
F18	Selective Set Group 1 Register
F19	Selective Set Group 2 Register
F20	Nonstandard
F21	Selective Clear Group 1 Register
F22	Nonstandard
F23	Selective Clear Group 2 Register
F24	Disable
F25	Execute
F26	Enable
F27	Test Status
F28	Nonstandard
F29	Reserved
F30	Nonstandard
F31	Reserved

No todas las funciones que establece el estándar son aplicables a todos los módulos, para saber cuáles sí están habilitadas se debe consultar el manual del módulo.

Note que las operaciones *read* y *write* están definidas para ejecutarse en los “registros del

grupo 1" o en los "registros del grupo 2". Es necesario mencionar que un registro es una 'fuente o receptor de datos accesable' en un módulo, no necesariamente tiene la propiedad de almacenar datos. El registro sobre el que se desea escribir o desde el cual se desea leer en una operación debe especificarse en el comando y es conocido como *subaddres* o subsección en un módulo.

Los grupos 1 y 2 se refieren a dos conjuntos de hasta 16 registros cada uno. En los registros del grupo 1 se leen o escriben datos y en los registros del grupo 2 se puede leer o escribir información referente al estado u organización interna del módulo.

Ahora bien, son tres las operaciones genéricas que se pueden realizar, éstas se ejecutan en todos los módulos conectados al *dataway* y son:

Z Initialize. - Se usa al encender el sistema, debe causar la inicialización de los registros en todos los módulos.

I Inhibit. - Inhibe algunas características en todos los módulos.

C Clear. - Limpia algunos registros de todos los módulos.

La forma en que respondan los módulos ante estas señales es decisión del fabricante y debe estar bien documentada en el manual del módulo.

Ahora bien, la pregunta que surge inmediatamente es ¿cómo hacer para mandar ejecutar alguna operación? Para ello se necesita conectar el *crate* a una computadora. La conexión entre el *crate* y la computadora se conoce como *Highway*. Este *highway* va desde el controlador hasta la computadora y puede ser de tipo serial o paralelo (según el estándar). Los estándares para *highways* están definidos en términos del formato de mensaje y estándares de señal en los puertos de entrada y salida.

Highways paralelos.- Son capaces de transmitir datos a muy altas velocidades con un máximo de transparencia al *dataway*; es especialmente atractivo para transmitir en pequeñas distancias; de hecho es un *highway* corto.

Highways seriales.- Utilizan un número limitado de líneas y son particularmente atractivos en instalaciones donde son requeridos *highways* largos y donde el costo del cableado puede ser alto. Este sistema serial puede ser bit serial o byte serial con 8 bits por byte.

El protocolo para controlar un sistema CAMAC desde la computadora varía según el tipo de *highway* que se esté utilizando y debe estar muy bien documentado en el manual del

controlador.

Así, sea cual sea el *highway* que se utilice, el estándar distingue dos tipos de transferencias entre la computadora y el controlador: transferencias simples (*Single CAMAC Operation*) y transferencias de bloques (*Block transfer*).

Transferencia Simple.- Es la actividad que ocurre en respuesta de un comando CAMAC. Puede consistir en la transferencia de una palabra CAMAC (*Single Data Word*) entre un módulo y la computadora, en el cambio de estado de un módulo, o en la transferencia del valor de Q desde un módulo.

Transferencia de Bloques.- Secuencia específica de transferencias simples, iniciada por un comando de un nivel más alto que los de las operaciones simples. Este comando contiene toda la información requerida para especificar la secuencia deseada de transferencias simples. La información de control tal como la disponibilidad de la computadora para poder participar en transferencias de datos, el estado de la línea Q y/o el estado de ciertas LAM's y señales especiales de sincronía deben estar disponibles para que el controlador maneje la actividad en el *dataway*. Existen varios modos de transferencia de bloques y cuál o cuáles maneje el controlador son decisión de su diseñador y deben estar muy bien documentados en el manual.

Los modos no son otra cosa que algoritmos para la transferencia de bloques, pueden ser parte del *hardware* del controlador o pueden implementarse por *software*, y el usuario debe verificar que tanto los módulos, el controlador y en algunos caso el *software* que implemente en la computadora, sean capaces de llevar a cabo tal o cual algoritmo correctamente.

Los distintos métodos de transferencia de bloques pueden clasificarse por la naturaleza de 3 características fundamentales:

- La forma en que se determina la dirección (*CAMAC Address*) de la siguiente operación.
- El origen de la señal de sincronía de transferencias individuales.
- El método usado para finalizar la transferencia de bloques.

TIPOS DE CONTROLADORES [2.3]

El estándar CAMAC define que la comunicación con la computadora puede ser serial o paralela; así pues, es posible encontrar diversos tipos de controladores según la interfaz que

ofrezcan. Los hay con interfaz paralela (como SCSI) o serial (como rs232), algunos ofrecen mayor velocidad de transferencia que otros, pero todos están diseñados para cubrir necesidades de conexión con equipos CAMAC. Así por ejemplo, del fabricante KineticSystems , en su catálogo CAMAC 1996-1997, es posible destacar los siguientes módulos controladores:

Modelo 3922.- *Parallel Bus Crate Controller.*

Permite la conexión con la computadora a través de una interfaz paralela RS485 con transferencias desde 250Kbytes/s, es posible conectarlo con el puerto PCI de una PC, con el Microchannel de una IBM PS/2 e inclusive puede conectarse a cualquier *slot* de expansión libre de una IBM PC/AT o compatible con el uso de adaptadores. La longitud de cable máxima que soporta es de 90m.

Modelo 3929.- *SCSI Crate Controller.*

Permite la conexión con la computadora vía la interfaz SCSI (*Small Computer Systems Interface*). Las especificaciones para un bus SCSI lo han convertido en uno de los canales más populares en PC's y *Workstations*; con éste, es posible direccionar o acceder hasta 8 periféricos (impresoras, *diskdrives*, gabinetes multitarjetas) en el mismo bus. El cable de conexión con la computadora es capaz de transferir datos a 5Mbytes/s, lo cual permite aplicaciones en tiempo real. La longitud de cable máxima que soporta es de 6m.

Modelo 3972.- *Interconnect CAMAC Crate Controller.*

Permite la conexión con un bus GI de fibra óptica (*Grand Interconnect*). Este bus soporta hasta 126 nodos incluyendo equipos VXI o CAMAC. La distancia máxima entre dos nodos es de 2 Km. El controlador CAMAC se comportará como un dispositivo esclavo que recibe los comandos de un *Interconnect Highway Driver*. La transacción de datos se inicia ya sea manipulando registros en el controlador o enviando una lista de operaciones. Por supuesto, la computadora debe contar con un *Interconnect Highway Driver* que es una interfaz al puerto PCI alcanzando una velocidad de transferencia de hasta 20Mbytes/s.

Modelo 3988.- *GPIB Crate Controller.*

Permite la conexión con la computadora a través de la interfaz GPIB (*General Purpose Interface Bus*). Soporta transferencias de datos de hasta 600Kbytes/s.

Modelo 3989.- RS-232 Crate Controller.

Permite la conexión con una computadora o con un módem, a través del puerto serial RS-232. Este es el tipo de controlador que se usará en este trabajo de modo que sus características se presentarán con detalle más adelante.

2.1.3.- El CAMAC y las otras opciones

Como ya se mencionó, CAMAC fue uno de los primeros estándares de instrumentación modular. A lo largo de los años, nuevos estándares han sido desarrollados para obtener un desempeño óptimo en la instrumentación y para integrar las nuevas tecnologías.

Aún cuando CAMAC sobrevive aún y sigue vendiéndose, los nuevos estándares ofrecen claras ventajas.

El VXI por ejemplo [2.4], es un estándar de gabinete multitarjetas introducido en 1987 (IEEE 1155) con un bus (*dataway*) de 32 bits frente al de 24 bits de CAMAC. El gabinete VXI más común contiene 13 *slots* (aunque también los hay de 5, 6 o 20 *slots*) y requiere también de un controlador llamado *slot-0 controller*. Los *slots* restantes pueden albergar otro tipo de módulos tales como procesadores, memoria o instrumentación.

Una de las principales ventajas de VXI es la gran variedad de soluciones para la adquisición y/o control de señales analógicas; ofrece versatilidad y velocidad. Por ejemplo, un lector de entradas analógicas puede hacer 200 000 lecturas en cada uno de sus canales, por segundo (lo que no se logra con CAMAC). Además, ofrece instrumentos únicos capaces de adquirir y generar señales tanto analógicas como digitales en un mismo módulo.

Otra de las ventajas de VXI es que puede usarse para aplicaciones de alta frecuencia en el rango de los GigaHertz (CAMAC trabaja a frecuencias máximas del orden de los MegaHertz).

La computadora que controla al VXI puede ser una computadora embebida como parte del *slot-0 controller* o bien puede ser una computadora externa conectada a través de un bus de interconexión (*highway*) que puede ser paralelo, SCSI, Ethernet, serial, u otro bus de interconexión. Es posible obtener comunicación entre crates o con la computadora a velocidades desde 10Mbytes/s.

VXI permite autoconfiguración pues cada módulo proporciona información acerca de su localización lógica y física y es posible saber por *software* si un módulo se encuentra o no presente.

Los productos en el mercado, ofrecen secuenciadores de listas capaces de almacenar enormes cantidades de datos en forma local para ser más tarde leídos y/o procesados por la computadora (también se cuenta con módulos similares para equipos CAMAC; sin embargo su capacidad de almacenamiento es inferior).

Así pues, VXI es capaz de ofrecer un verdadero desempeño en tiempo real usando a la computadora como un simple “*master*” y realizando todas las acciones críticas (incluyendo toma de decisiones) por medio de *hardware*.

Una última ventaja: Actualmente se ha desarrollado una gran cantidad de *software* para diversas plataformas, capaz de crearle al usuario “instrumentos virtuales” con VXI.

2.2.- Funcionamiento del sistema CAMAC que se utiliza en este trabajo.

2.2.1.-EL CRATE

El *crate* CAMAC que se usa en este trabajo [2.5], es el modelo 1570 de KineticSystems. Cumple con la especificación CAMAC IEEE 583, de modo que es un gabinete multitarjetas de 25 estaciones que incluye un *dataway* y conectores de 86 contactos en cada estación (para la conexión con los módulos).

Según lo dicta la norma CAMAC, el *crate* cuenta con una única fuente de poder para alimentar a todos los instrumentos. Con el fin de disminuir el riesgo de falla y prolongar la vida de la fuente de poder, incluye también una unidad de ventilación.

El manual del *crate*, contiene información detallada acerca de sus especificaciones físicas, instalación, uso y mantenimiento. Ello sale del ámbito de este trabajo, es por ello que tales características no se describen.

2.2.2.- Funcionamiento del RS-232 *Crate Controller* [2.6]

El sistema CAMAC que se utilizará cuenta con un módulo controlador con puerto serial RS-232; es el Model 3989 de KineticSystems que ocupa 2 estaciones.

Sigue los lineamientos del estándar, de modo que el controlador tiene capacidad para manejar un *crate* CAMAC con 25 estaciones, 2 de las cuales serán ocupadas por el mismo controlador (la 24 y la 25).

Se tendrá disponible el código de funciones que describimos anteriormente, que consta de 32 funciones F0 a F31, El controlador generará las señales necesarias en el *dataway* para ejecutarlas.

Se podrán enviar las 3 señales genéricas *Initialize, Clear e Inhibit*, a través de comandos CAMAC.

Para ejecutar cualquier operación, deberá especificarse un número de función *function code* (F0 a F31), un número de estación *station number* (N0 a N23 y N30) y una subsección de módulo *subaddres* (A0 a A15) que es usada para acceder diferentes secciones de un mismo módulo.

Los módulos enviarán información de estado al controlador a través de las líneas LAM, Q y X.

Se pueden hacer transferencias de bloques en modo *Address Scan, Stop on transfer count* y *Q-scan* si los módulos involucrados soportan tales modos de transferencia. Por supuesto la operación de los tres modos está documentada en el manual del controlador.

El usuario de un sistema CAMAC podrá configurar al controlador según sus necesidades, las opciones incluyen:

- 1.- Medio de transmisión RS-232 o RS-422.
- 2.- Puede habilitar o deshabilitar algunas señales de control y estado del RS-232.
- 3.- Amplia selección del reloj que controlará la transmisión y recepción.
- 4.- Formato de datos ASCII decimal, octal o hexadecimal.
- 5.- Paridad par o impar.
- 6.- Habilitar o deshabilitar la detección de errores o CHECKSUM.

- 7.- Trabajar en modo NORMAL o en modo MONITOR.
- 8.-Selección de la P-2 *Signal hold*
- 9.-Selección de la Z *Signal 3989 reset*.
- 10.- 7 u 8 bits de datos por caracter.

CONFIGURACION POR SWITCHES

Puede configurarse para que trabaje en:

(la opción subrayada es el valor con el que viene configurado de fábrica)

Modo: Monitor o Normal.

Número de bits de parada: 1, 1.5 o 2.

Longitud de Caracter: 7 u 8 bits

Tipo de formato ASCII: decimal, hexadecimal u octal.

Paridad: par o impar.

Generación y chequeo de paridad: Habilitada o deshabilitada.

Generación y chequeo de errores CHECKSUM: habilitada o deshabilitada.

Baud rate: de 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200.

Crate Address (en operación normal): 0,1,2,...,127 excepto 17 y 19 que son reservadas.

Estándar de Transmisión: RS-232 o RS-422.

Línea RTS *request to send*: Habilitado o deshabilitado.

Línea DTR *data terminal ready*: Habilitada o deshabilitada.

Reloj: Suministrado por el controlador

Señal P-2 *hold*: habilitada o deshabilitada.

Señal *reset on Z*: habilitada o deshabilitada.

Configuración: *Multidrop* o *Stand Alone*

Así, el controlador del *Crate* puede ser usado en cuatro configuraciones básicas: como controlador principal, como controlador auxiliar, como sistema de múltiples controladores o como CAMAC *test system*.

CAMAC TEST SYSTEM.- El controlador debe configurarse en modo monitor, como principal o como auxiliar. Este modelo de controlador puede verse como 2 módulos separados. Uno, en modo normal, es utilizado conjuntamente con la computadora con el propósito de transmitir señales por el RS-232 ó por el RS-422. El otro módulo es un simple *CAMAC TEST CONTROLLER* (Controlador de monitoreo) si se usa con una terminal de video (*video terminal*). El usuario puede hacer una lista de comandos CAMAC para monitorear y evaluar el buen funcionamiento de los módulos. En el modo monitor no puede hacerse manipulación ni transferencia de datos, las características en este modo se resumen:

- No se tiene acceso a los registros internos del controlador y sólo se pueden hacer operaciones *dataway*.
- Sólo se permiten transferencias de datos simples (*no block transfer*).
- Los datos resultantes de una transferencia no pueden ser manipulados o procesados.
- No se pueden tomar decisiones de *software*.
- El controlador debe estar *ON- LINE*.

El modo monitor es sólo un medio rápido de evaluar el funcionamiento de los módulos CAMAC sin necesidad de una computadora. El controlador permite al usuario (en este modo) crear y editar una lista de comandos CAMAC a ser ejecutados, todo ello a través de un menú que aparece en la pantalla de la terminal.

CONTROLADOR PRINCIPAL.- Requiere estar conectado en los *slots* 24 y 25 del *Crate*, configurado en modo normal, requiere el uso de una computadora con puerto serial para comunicación serial.

CONTROLADOR AUXILIAR.- Para trabajar como controlador auxiliar requiere estar configurado en modo normal y conectado con un controlador principal al que estará supeditado. Por supuesto que necesita estar conectado a una computadora en forma serial.

MULTICRATE SYSTEM.- Para trabajar en un sistema *multicrate* requiere estar configurado en modo normal ya sea como controlador principal o auxiliar pero con la opción *multi-drop*. Requiere estar conectado a un bus RS-232 el cual llegará a la computadora y a los otros *Crates*.

Es importante hacer notar que toda vez que se configure en modo normal, se tendrán las siguientes reglas:

- 1.- La computadora puede acceder los registros internos del controlador aún cuando éste se encuentre *OFF LINE*, usando la dirección N=30; sin embargo para acceder al *dataway* y a los módulos debe estar *ON LINE*.
- 2.- Toda la transmisión de datos al y desde el controlador es en ASCII, no en binario.
- 3.- La salida del controlador puede controlarse enviando caracteres XON (para habilitar salida) y XOFF (para deshabilitar salida).
- 4.- Podrán conectarse hasta 125 *crates* en *multidrop*.
- 5.- El controlador puede producir una señal de BREAK en ciertas condiciones (el usuario las decide). Si la computadora es incapaz de detectar un BREAK, el estado del controlador puede ser monitoreado periódicamente.
- 6.- Asimismo, la computadora puede interrumpir al controlador en cualquier momento, enviando una señal de BREAK. Esto pondrá al controlador en un estado inicial.
- 7.- El número de bits de datos a transferir entre la computadora y el controlador puede seleccionarse por *software* entre 24, 16 u 8 bits lo que permite transferencias más rápidas.
- 8.- Se pueden seleccionar tres tipos de transferencia de bloques.
- 9.- También por *software* puede elegirse que el controlador envíe a la computadora un byte de estado que informe de las condiciones actuales del *crate*.

Todos los mensajes enviados y recibidos entre la computadora y el controlador deben seguir un formato; es decir, el controlador espera recibir los comandos, a través del puerto RS-232, como una secuencia ordenada de caracteres ASCII. Solo así los reconocerá como comandos válidos y se alistará a ejecutarlos. Asimismo, las respuestas que envía a la computadora son secuencias específicas de caracteres ASCII que deberán ser interpretadas.

A continuación se describen los formatos que se aplican según el tipo de transferencia deseado.

a) Transferencias simples de escritura.

Formato de comando: SOH C N, A, F STX DH, DM, DL ETX CHKSUM

Formato de respuesta: SOH C STX STATUS ETX CHKSUM

b) Transferencias simples de lectura

Formato de comando: SOH C N, A, F STX ETX CHKSUM

Formato de respuesta: SOH C STX DH, DM, DL, STATUS ETX CHKSUM

c) Transferencias simples de control

Formato de comando: SOH C N, A, F STX CHKSUM

Formato de respuesta: SOH C STX STATUS ETX CHKSUM

d) Transferencias de bloques de escritura en Address Scan

Formato de comando:

SOH C N, A, F STX DH, DM, DL, CHKSUM DH, DM, DL, CHKSUM ... DH, DM, DL, CHKSUM ETX CHKSUM

e) Transferencias de bloques de lectura en Address Scan

Formato de comando: SOH C N, A, F STX ETX CHKSUM

f) Transferencias de bloques de escritura en Q-stop

Formato de comando:

SOH C N, A, F STX DH, DM, DL, DH, DM, DL, ... DH, DM, DL, ETX CHKSUM

g) Transferencias de bloques de lectura en Q-stop

Formato de comando: SOH C N, A, F STX ETX CHKSUM

donde SOH=ASCII 1, C=Caracter ASCII correspondiente a la dirección del *Crate*, N=Dirección del módulo, A=*subaddress*, F=*function code*, STX=ASCII 2 y ETX=ASCII 3. Por otro lado, DH, DM y DL son los datos a escribir o recibir, están codificados según los requerimientos del módulo referido por N. Son siempre cadenas de caracteres representado cada una números entre 0 y 255.

Así, por ejemplo, si se desea hacer una transferencia simple de escritura (con *function code* 17) dentro del *crate* cuya dirección es 65 (C=65), hacia el módulo ubicado en la estación 5 (N=5), refiriéndose al canal 3 del módulo (A=3), entonces deberá transmitirse por RS-232 la cadena de caracteres equivalente a:

$$\text{ASCII 1} + \text{"A5,3,17"} + \text{ASCII 2} + \text{"255,0,30"} + \text{ASCII 3}$$

de modo que DH=255, DM=0 y DL=30 (el signo '+' indica una operación de concatenación).

Es posible controlar por *software* si serán 8, 16 o 24 bits de datos los que se transferirán por las líneas del *dataway* en comandos de escritura. Ello se logra enviando únicamente DH para 8bits, DH y DM para 16bits y DH,DM y DL para 24bits.

En cuanto a transferencias de bloques, es importante hacer notar que son necesarias algunas previsiones adicionales antes de realizarlas; en este trabajo no se manejará este tipo de

transferencia, sin embargo está bien documentada en el manual correspondiente.

El CHKSUM es un dato opcional, si la generación/detección de CHECKSUM está habilitada, el usuario deberá calcularla tal como se indica en el manual y enviarla en el formato descrito. El byte de STATUS es también opcional, el usuario deberá indicar al controlador si debe generarlo.

El controlador cuenta con siete registros internos que pueden ser leídos y/o escritos según sea el caso:

- 1) *Control Status Register (CSR)*.- Este registro permite leer o definir las características generales de funcionamiento del controlador y de las señales que es capaz de generar o detectar, tales como I,Z,C,Q y X.
- 2) *Transfer Count Register (TCR)*.-Este registro permite leer o definir el número de “palabras” involucradas en una transferencia de bloques.
- 3) *LAM Request Register (LRR)*.- Este registro permite detectar la dirección de los módulos que están enviando una petición LAM.
- 4) *Disable LAM Mask (DLMR)*.- Este registro permite decidir las direcciones de los módulos que serán capaces de generar una petición LAM.
- 5) *Break Mask Register (BMR)*.- Este registro permite definir las condiciones que deben cumplirse para que el controlador pueda generar una señal de BREAK.
- 6) *Interrupt Request Register (IRR)*.- Este registro permite detectar las condiciones que indujeron al controlador a generar el último BREAK.
- 7) *Extended Status Register (EXS)*.- Este registro permite leer las características generales de estado del controlador después de la última transferencia (como por ejemplo el estado de las líneas Q,X y LAM).

El acceso los registros internos del controlador se hace de la misma manera que las transferencias simples, sólo que aquí siempre deben incluirse 24 bits de datos y un N=30.

2.3.- El CAMAC y el interferómetro.

2.3.1.- Configuración actual del controlador del *CRATE*.

La configuración actual del controlador del *Crate*, responde a las necesidades de esta aplicación y es como sigue:

P-2 hold: habilitada

CRATE Address: 65 ó ASCII 'A'

MODE/PARAMETER: 2 bits de parada, 8 bits por caracter , operación NORMAL.

FORMAT: ASCII decimal, paridad impar habilitada, CHECKSUM deshabilitado.

RESET ON Z: habilitada.

BAUD: 19 200 bauds

MULTIDROP: *stand alone*

CLOCK SELECT: Reloj asincrono.

RS-232: RTS y DTR deshabilitadas; trabajando bajo RS-232.

2.3.2. Módulos indispensables

Dados los requerimientos para el desarrollo del presente trabajo, dos módulos CAMAC resultaron indispensables: el modelo 3982-Z3B *List Sequencing Crate Controller* y el modelo 3664-L1A *Time Interval Counter*. Estos módulos hacen posible la conexión del sistema interferométrico con la computadora. A continuación se describen sus características generales (la participación que tendrá cada uno de los módulos en esta aplicación, se describirá más adelante en el capítulo 4).

Modelo 3982-Z3B *List Sequencing Crate Controller* [2.7].

Este módulo es un secuenciador de listas y al mismo tiempo un controlador auxiliar autónomo. Es capaz de almacenar una lista de 8192 comandos (ya sean de escritura, lectura o control), para ser ejecutados en forma diferida. La ejecución de la lista puede ser iniciada a través de un comando enviado desde la computadora, a través de la recepción de una señal eléctrica

externa en un canal del módulo dispuesto para ese fin, o bien puede ser iniciada una vez que sea detectada una señal LAM generada por otro módulo específico.

Es posible elegir por *software* si la lista debe ser ejecutada periódicamente y en ese caso definir la frecuencia de ejecución (misma que puede ir desde 2Hz hasta 500Hz según lo defina el usuario). Por otro lado, la frecuencia con que ha de ejecutarse cada comando también puede determinarse por *software* y puede ir desde 5KHz hasta casi 1MHz.

Este módulo cuenta con un medio de almacenamiento capaz de albergar hasta 4096 respuestas CAMAC y es capaz de generar una señal LAM bajo circunstancias predefinidas. También cuenta con una serie de registros internos que permiten definir sus características de operación.

Funciona como controlador auxiliar autónomo; es decir, debe conectarse a un controlador principal que en este caso es el modelo 3989 RS-232 *Crate Controller*.

Modelo 3664-L1A *Time Interval Counter* [2.8].

Es un módulo capaz de medir intervalos de tiempo en 6 canales diferentes. Cuenta con un reloj interno programable para generar TICKS a frecuencias desde 1Hz hasta 10MHz. La cuenta de tiempo representa el número de TICKS generados desde el arribo de una señal de inicio (que bien puede ser una señal eléctrica externa -pulso- común a los 6 canales o un comando específico) y hasta la recepción de un número predeterminado de pulsos en un canal.

Las frecuencias aceptables en los pulsos de entrada son de 1Hz a 5MHz, el número máximo de pulsos a esperar por canal es de 65535, y el intervalo máximo de tiempo que puede almacenar es de 16777215 TIKS de reloj.

Este módulo está capacitado para generar una señal LAM tanto en el momento en que se ha detectado el arribo del número de pulsos predeterminado, como en el momento en que el intervalo de tiempo ha rebasado el máximo tiempo posible.

El módulo también cuenta con una serie de registros internos que permiten definir sus características de operación, se hará referencia a ellos más tarde en el capítulo 4.

CAPITULO 3. - CONEXION DE LA ESTACION DE TRABAJO UNIX CON EL SISTEMA DE ADQUISICION DE DATOS.

3.1.- UNIX y el lenguaje de programación C.

UNIX es un sistema operativo que ha alcanzado una posición de gran importancia. Se ha instalado en todo tipo de máquinas, desde microcomputadoras hasta supercomputadoras.

Entre sus principales características pueden distinguirse su naturaleza como sistema abierto, el soportar múltiples usuarios al mismo tiempo, la multiprogramación (en la cual muchos procesos residen en memoria y compiten por los recursos de la máquina) y su amplia documentación.

Sin embargo, UNIX no es un ente único; diferentes proveedores ofrecen sus sistemas con características especiales (UNIX SystemV, UNIX de Berkeley, LINUX, HP-UX, etc.) todos ellos conformando la gran familia de sistemas UNIX.

UNIX nace de la mano del lenguaje de programación C a principios de los años 70. C es considerado como un lenguaje de alto nivel que conserva la habilidad de trabajar en bajo nivel, es un lenguaje pequeño que contiene sólo las operaciones básicas; de modo que las entradas, las salidas y el multiprocesamiento son ejemplo de operaciones que son realizadas a través de llamadas a funciones.

Combinar las potencialidades de UNIX con la versatilidad del lenguaje de programación C, resultó ser una herramienta vital para la comunicación que se buscaba entre el sistema de adquisición de datos y la computadora.

Aquí se trabajó con el sistema de Hewlett Packard HP-UX al que en forma genérica se le ha llamado UNIX en el desarrollo del presente trabajo, y los programas necesarios fueron escritos en C.

3.2.- El manejo de dispositivos en UNIX (HP-UX)

En UNIX, para poder comunicarse con un dispositivo cualquiera, ya sea una impresora, un modem o incluso el disco duro, se necesita:

- Un medio para conectar físicamente el dispositivo: un puerto de entrada y/o salida, un manejador del disco duro o de *floppy* según sea el caso, y por supuesto el cableado necesario.
- Un programa que maneje el dispositivo (*driver*) **que sea parte del *kernel*** de UNIX.
- Al menos un archivo de dispositivo asociado con el dispositivo en cuestión.

Cada fabricante, dota a sus computadoras de diversos medios para comunicarse con dispositivos; (los dispositivos pueden ser terminales, impresoras, discos duros, cintas, modems, y las tarjetas de interfaz a las que están conectados). De hecho existen algunos puertos obligados (hablando de computadoras personales y estaciones de trabajo), como el puerto para conectar el monitor y el puerto para conectar el teclado. En UNIX, aún estos puertos, deben tener un *driver* que los maneje y al menos un archivo de dispositivo asociado.

Para entender estos requerimientos, es necesario conocer tanto la función del *kernel* de UNIX como la función de los *drivers* en el manejo de dispositivos.

El *kernel* o núcleo del sistema operativo UNIX, entre otras cosas, es el encargado de controlar el acceso a la computadora y a sus archivos, supervisa la transmisión de la información entre la memoria principal y los periféricos. La única vía de acceso directo al *kernel* son las llamadas al sistema (*system calls*); de modo que todo, incluso los comandos, está hecho con *system calls*. Estas son la interfase entre el *kernel* y los programas de usuario que corren en él. Es la única manera de acceder al sistema de archivos, a los mecanismos multitarea, a la comunicación entre procesos, etc. El mismo *shell* está hecho de llamadas al sistema.

Las *system calls* que soporta cada sistema, y la función que realizan puede consultarse en la sección (2) de los manuales de referencia del sistema, y aunque pueden variar de uno a otro, existen algunas que son básicas como por ejemplo: *open()*, *read()*, *write()*, *close()* que son las encargadas de 'abrir', 'leer', 'escribir' o 'cerrar' archivos respectivamente. La información en esa sección del manual, supone la utilización del lenguaje C en los programas de usuario pero pueden

hacerse llamadas al sistema desde otros lenguajes de programación.

Por otro lado, se sabe que todo manejo de datos en UNIX se hace a través de archivos. En el sistema de archivos se definen tres tipos:

- Archivos normales (u ordinarios).- Contienen datos y pueden estar en formato binario o en ASCII.
- Directorios.- No contienen datos en sí, pero contienen ligas hacia otros archivos.
- **Archivos especiales.**-Tampoco contienen datos pero representan entidades tales como dispositivos (archivos de dispositivo) o *pipes* (archivos FIFO).

Para el *kernel*, ningún archivo tiene un nombre asociado directamente, en vez de eso, tiene asociado un número *i-number* que es un índice a un arreglo de i-nodos. Cada i-nodo contiene información tal como el tipo de archivo (ordinario, directorio o especial), número de ligas, identificador ID del propietario del grupo, los permisos de lectura, escritura y ejecución del propietario, del grupo y de los otros, el tamaño en bytes (en caso de archivos ordinarios), las fechas de último acceso, última modificación y cambio de status, así como el puntero a los bloques de disco que es donde están guardados los datos que contiene el archivo.

El acceso a los archivos se hace entonces a través de directorios. Un directorio consiste de una tabla que contiene el nombre del archivo y su *i-number* correspondiente. Así, cuando un usuario hace referencia a un nombre de archivo, el sistema operativo busca en el directorio el *i-number* correspondiente, con ese *i-number* tiene acceso a la información contenida en el i-nodo asociado, éste puede llevarnos a los datos en caso de que se requiera. El par nombre-de-archivo/i-nodo es llamado 'liga'.

En el caso de los archivos especiales, los i-nodos no contienen el tamaño de datos, en vez de eso, contienen un número de dispositivo *device number*, el cual es un índice a una tabla usada por el *kernel* para encontrar una colección específica de rutinas que serán las encargadas de comunicarse con el dispositivo llamadas *device driver*. Los permisos de lectura y escritura sobre estos archivos especiales significan la habilidad para ejecutar las llamadas al sistema *read()* y *write()*; el permiso de ejecución no tiene sentido.

Un *driver* es entonces *software* cuya finalidad es enlazar al sistema operativo con los dispositivos de entrada y salida; las rutinas que lo conforman se denominan *entry points*. Cuando

el *kernel* detecta que una aplicación está haciendo una llamada al sistema *system call* para realizar una operación sobre un archivo especial, delega al *driver* la tarea de entenderse con el dispositivo, ejecutando la rutina o *entry point* adecuada (para una llamada *open* corresponderá un *open entry point*). El *driver* actúa como traductor, convirtiendo las peticiones del sistema operativo en comandos que los dispositivos (o mejor dicho, los controladores de los dispositivos) puedan entender. La existencia de *drivers* simplifica al sistema operativo, sin ellos, éste sería el responsable de tratar directamente con el *hardware*. Esto significaría que los diseñadores de sistemas operativos tendrían que incluir el soporte para absolutamente todos los dispositivos conectables a la computadora, también significaría que la inclusión de un nuevo dispositivo requeriría la modificación del código del sistema operativo en sí (ver fig. 3.1).

Así, el sistema operativo delega a los *drivers* la función de atender los detalles de *hardware* de los dispositivos, de modo que el diseñador del sistema operativo no debe preocuparse por los detalles sino por los aspectos generales; y el diseñador de *drivers* sólo debe preocuparse por los detalles.

Algunos autores manejan cuatro tipos de *drivers* (clasificados por la forma en que se comunican con el sistema operativo):

1. De Bloques.- Se comunican a través de bloques de datos de tamaño fijo, se utilizan para manejar dispositivos como diskets y discos duros. La transferencia es realizada hasta el momento en que se tiene un bloque completo.
2. De Caracter.- Manejan transferencias de datos de tamaño variable y la transferencia es directa, un carácter a la vez. Se usan para el manejo de terminales, modems, impresoras, etc.
3. De Terminal.- Son los *drivers* de carácter especializados en conectar terminales.
4. *STREAMS*.- Se usan para comunicación a altas velocidades y bajo algún protocolo (principalmente en el manejo de redes).

El tipo de *driver* definirá los dispositivos que puede soportar, el sistema operativo espera encontrar un conjunto específico de *entry points* según el tipo de *driver* que se esté manejando.

En UNIX, generalmente un *driver* está escrito en lenguaje C (pero restringido a las instrucciones y funciones que son soportadas por el *kernel*). El código objeto de los *drivers* debe ser ligado con los objetos del *kernel* (se reconfigura el *kernel*) y el resultado es un nuevo archivo

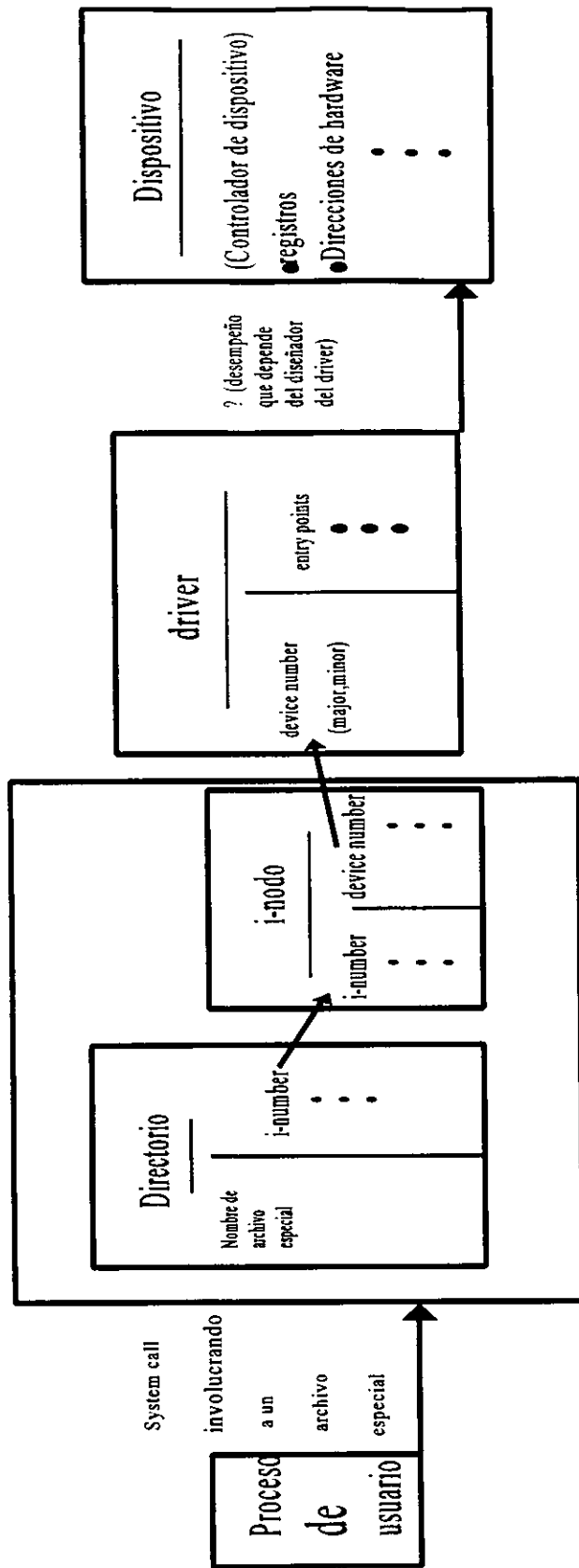


Fig. 3.1 Interrelación entre sistema operativo, driver, dispositivo y proceso.

de arranque del sistema operativo, que incluye los *drivers*. Así se logra que el sistema operativo vea a los dispositivos como si fueran archivos. (Los *drivers* en el sistema se localizan convencionalmente en la biblioteca `/etc/conf/lib`).

A diferencia de los programas de aplicación, los *drivers* no constituyen procesos normales sino que son parte del *kernel* y de esa manera pueden hacer cosas que las aplicaciones no pueden; como acceder y tal vez modificar cualquier parte del *kernel* y de los registros o memoria de la computadora, deshabilitar y/o recibir interrupciones de *hardware*, acceder directamente el *hardware*.

Lo que un *driver* haga es decisión absoluta de su diseñador, de modo que el usuario tan sólo lo puede ver como una caja negra con un comportamiento que debe estar perfectamente documentado. Los *drivers* que suministran los fabricantes se suponen estar bien documentados a modo de expresar lo que se espera que hagan ante una determinada llamada al sistema. También se suponen ser suficientemente versátiles como para satisfacer los requerimientos de los usuarios en el manejo de dispositivos; sin embargo si por alguna razón esto no se cumple, es posible que el usuario escriba e implemente sus propios *drivers*.

El escribir un *driver* no es una tarea fácil. Pueden ser escritos en C pero únicamente con las funciones soportadas por el *kernel* y generalmente ésta no es información de fácil acceso y puede variar de una versión a otra de UNIX lo que dificulta la portabilidad. El *driver* debe resolver consistentemente tres aspectos: el intercambio de información con el sistema operativo, el intercambio de información con el dispositivo y el funcionamiento propio del *driver* en el sentido de cómo interpreta los comandos del sistema operativo, cómo se administra en tiempo para dar servicio a varias peticiones, cómo procesa las interrupciones de *hardware*, etc. Para todo ello se debe contar con información precisa del *hardware* que se usará tal como registros y direcciones a las que el dispositivo responderá y la forma en que lo hará (datos técnicos), así como la información correspondiente a la computadora y sistema operativo que se usará.

El diseño del *driver* debe estar muy bien hecho y el código optimizado al máximo, para evitar tener sorpresas a la hora de hacer las pruebas y de implementarlo, pues el *driver* formará parte del *kernel* (o sea del sistema operativo en sí) y algún mal manejo puede ser fatal. También debe probarse exhaustivamente porque si las fallas resultan a través del tiempo puede resultar en

información falseada o perdida y eso puede ser aún más costoso.

Se puede decir que para decidirse a escribir un *driver* es necesario evaluar si la funcionalidad y el desempeño del dispositivo mejorará en forma importante frente a las dificultades, costos y riesgos en que se incurrirá.

De cualquier manera, cuando se desea manejar un dispositivo en UNIX, se debe estar seguro de que se cuenta con los tres requisitos mencionados, y de no ser así, emprender las acciones necesarias .

Cuando se adquiere una máquina computadora, el fabricante debe proporcionar las especificaciones del producto que se está adquiriendo tales como versión del sistema operativo, capacidad de memoria y velocidad de procesamiento, puertos de entrada/salida, estándar de red, los *slots* que tienen para expansión, etc.

Si la máquina no cuenta con el medio físico que se necesita, será necesario consultar con el fabricante o distribuidor, para saber si puede adquirirse (un puerto o un *drive*) y de ser así hacerlo.

Como se mencionó, el fabricante proporciona los *drivers* que manejarán los dispositivos que la máquina puede soportar (o el usuario pudo haber escrito los suyos propios). En el caso de HP-UX, se cuenta con un manual que indica cómo instalar los dispositivos más comunes [3.1], tales como impresoras, módems, terminales, etc. Básicamente lo que debe hacerse es conectar el dispositivo a la computadora y entonces configurar el *kernel* (lo que incluye modificar sus archivos y tablas de configuración), ligar el *driver* con los archivos objeto del *kernel* para producir uno nuevo, crear las entradas necesarias en el directorio */dev* (archivos especiales) y reinicializar el sistema con el nuevo *kernel*. Como los detalles para efectuar cada paso dependen de la versión de UNIX y/o del modelo de máquina que se use, debe consultarse el manual correspondiente.

Es necesario reconfigurar el *kernel* en las siguientes situaciones:

- Cuando se añade un nuevo periférico al sistema, que requiere un *driver* que no está configurado en el *kernel* (hay que añadir el *driver*).
- Cuando se desea quitar un *driver* del *kernel* (en el caso de que no sea usado por ningún periférico). Es deseable retirar *drivers* en ocasiones ya que un *kernel* pequeño es más eficiente.

- Cuando se cambian los parámetros del *kernel*, con el objeto de obtener un óptimo desempeño.
- Cuando se añade cierto tipo de *software* para red (LAN o *Network services*).
- Cuando se cambia la ubicación de ciertos ‘dispositivos’ del *kernel* como son: el *root file system*, el *system console*, etc.

Es necesario hacer notar que el reconfigurar el *kernel* es una actividad que sólo puede ser realizada por un superusuario (o root).

El HP-UX cuenta con una herramienta llamada *System Administration Manager* (SAM) que puede ayudar en la reconfiguración del *kernel* y la generación de los archivos especiales adecuados.

Por supuesto que esta actividad también puede hacerse a través de comandos UNIX.

3.3.- Conexión física entre la computadora y el sistema de adquisición de datos

Para lograr la conexión física entre una computadora y el CAMAC, se necesita que la computadora tenga disponible un puerto de entrada/salida compatible con el del controlador del CAMAC; en este caso es un puerto RS 232. También se necesita tender un cableado adecuado.

Antes de describir dicha conexión, será conveniente presentar al equipo con el que se cuenta. Se está trabajando con una máquina HP 9000 Modelo 712/60 con las siguientes características:

- Sistema operativo HP-UX ver. 9.03
- Procesador PA7100LC a 60 MHz
- Memoria 16MB.
- Interfaz de usuario gráfica HP VUE 3.0
- Disco duro 8.5' SCSI de 400MB
- *Floppy drive* 3.5' HD
- Estándares de red soportables: Ethernet IEEE 802.3, LAN AUI y TP (*twisted pair*)
- Puerto de red opcional: puerto IBM token ring 802.5
- Puertos I/O estándar: Interfaz serial EIA RS232C , CCITT V.24/V.28 con conector macho

de 9 pines DTC (PC estándar). Interfaz paralela bidireccional de 24 pines Centronics. Dos puertos IBM PS/2 (PC 101/102 estándar) para teclado y mouse. Entrada y salida de audio. Puerto para monitor. Interfaz SCSI-2 *single ended* de 8 bits

- Puertos I/O opcionales: Otro puerto RS232C de 9 pines, ó puerto LAN AUI y RS232C, ó X.25 *data link* HDLC y RS232C, ó puerto 802.5 IBM token ring.

Actualmente se está conectado en red por Ethernet.

Muy importante será la arquitectura del bus en el manejo de dispositivos [3.2].

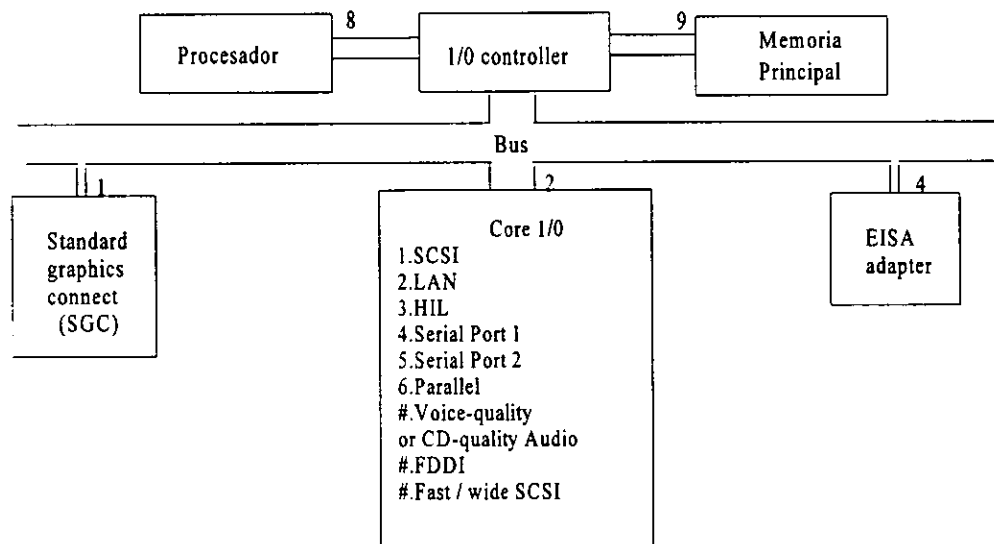


Fig. 3.2. Arquitectura de los sistemas 700.

En la figura 3.2, los números junto a los recuadros representan el *System Bus Module Number*. Cada interfaz del Core I/O cuenta con un número de función (*function number*). Esos números son usados en archivos de dispositivo como parte del *device number*, también constituyen el *hardware path* para cada elemento.

El Core I/O es una tarjeta multifuncional que incluye HP-HIL, audio, dos puertos asíncronos RS232 C , un puerto IEEE 802.3 Ethernet LAN, un puerto paralelo Centronics y un puerto SCSI-2 externo. Cada función se identifica por un *function number* separado.

El puerto que se usará es RS232-C llamado SIO (*Serial Input/Output*). Pueden conectarse a este puerto una gran variedad de dispositivos como impresoras, plotters, módems o scanners. Es un puerto programable; de modo que el usuario puede definir parámetros tales como baud rate, longitud de caracter, paridad, etc. (Más adelante se explicará cómo). El SIO es usado como interfaz para dispositivos seriales (que transmiten caracteres en serie) asíncronos; opera hasta 19.2K de baud rate.

En la serie 700 de las computadoras HP9000, los puertos seriales pueden soportar las siguientes configuraciones de baud rate adicionales: 57600, 115200, 230400 y 460800 bauds [3.3]. Puede requerirse un convertidor de RS232 a RS422 para lograr una longitud de cable funcional en esos baud rates (ya que RS232 sólo especifica hasta 19200 bauds).

También puede definirse control de flujo por las líneas RTS y CTS que puede ser habilitado a través de un bit en el *device number* o por *software*.

Estas características son válidas sólo para la serie 700, los datos referentes a características muy especiales de cada arquitectura o modelo deben consultarse en su manual correspondiente.

El conector serial con que cuenta la computadora es un DB-9 y la asignación de los pins es:

1	DCD	Data Carrier Detect
2	RxD	Receive Data
3	TxD	Transmit Data
4	DTR	Data Terminal Ready
5	GND	Ground
6	DSR	Data Set Ready
7	RTS	Request to Send
8	CTS	Clear to Send
9	RI	Ring Indicator

La comunicación *full-duplex*, característica de la norma RS-232, permite transmitir y recibir caracteres en ambas direcciones al mismo tiempo. Se dice que la conexión es con control de módem cuando se usan las líneas de control durante la comunicación; éstas verifican que el dispositivo y la computadora estén conectados antes de establecer la comunicación.

Se dice que la conexión es sin control de módem cuando no se usan esas líneas de control y sólo se utilizan las líneas básicas: Recepción (RxD), transmisión (TxD) y tierra (GND). También es conocida como módem nulo (*null modem*) o conexión directa. En este caso se parte del supuesto de que la computadora y el dispositivo están conectados en todo momento.

También puede decirse que un periférico bajo interfaz RS232 puede establecer dos tipos de comunicación:

- DCE *Data Communication Equipment* (transmite por el pin3 y recibe por el pin 2)
- DTE *Data Terminal Equipment* (transmite por el pin 2 y recibe por el pin 3)

El CAMAC utiliza el tipo DTE con conector DB25 cuya asignación de pins es:

2	TxD	Transmit Data
3	RxD	Receive Data
4	RTS	Request to Send
5	CTS	Clear to Send
6	DSR	Data Set Ready
7	GND	Ground
8	DCD	Data Carrier Detect
20	DTR	Data Terminal Ready
22	RI	Ring Indicator

Por las características del CAMAC que se describieron en el capítulo anterior, se puede considerar como si fuera un módem nulo (recuerde que aún cuando cuenta con otras ciertas líneas de control, no son necesarias y no las usa el CAMAC).

Para efecto de comunicación con la computadora, existen 3 tipos de acceso al puerto RS232 : *call-in*, *call-out* y *direct*.

- *call-in*.-Se usa cuando se espera que la conexión se establezca por una llamada entrante (de afuera hacia adentro)
- *call-out*.- Se usa para establecer la conexión a través de una llamada saliente (hacia el exterior).
- *direct*.- Se usa cuando no se desea usar las líneas de control. Se utiliza para terminales

conectadas directamente u otros dispositivos cuya conexión implica sólo los tres cables básicos.

Por otro lado, en el caso de desear trabajar con las líneas de control de módem, existen dos maneras: el modo CCITT y el modo simple. En el primero, las líneas de control son: DTR y RTS, las de status son: DSR, DCD, y CTS, también la señal RI indica la presencia de una llamada entrante; el usuario no puede cambiar la configuración ni el desempeño de las líneas de control. En el modo simple la línea de control es DTR y la de status es DCD; el usuario puede cambiar la configuración de las líneas de control (por *software*).

Los modos de control y el tipo de acceso al puerto que se deseen establecer, deben definirse al crear el archivo especial asociado. Esto entre otras cosas, será codificado en los números *minor* y *major* que serán explicados posteriormente con más detalle; estos números conforman el *device number*, son absolutamente dependientes de la implementación, varían de un equipo a otro y están bien documentados en el manual correspondiente.

Si más de un archivo trata de acceder un puerto en un mismo momento, para saber quien lo logrará con éxito, comienza a funcionar un mecanismo de interbloqueo de modo que se obedecerá a las siguientes prioridades: *direct*, *call-in*, *call-out*.

Hasta este momento se ha descrito en forma general las características de la computadora con la que se trabajará. Se sabe que cuenta con un puerto RS232 compatible con el del CAMAC y la conexión será directa. Así las cosas, para lograr la conexión física lo único que hace falta es construir un cable que puede ser de par trenzado blindado, con un conector macho DB25 en un extremo y un conector hembra DB9 en el otro extremo; con los pins conectados como se muestra en la figura 3.3.

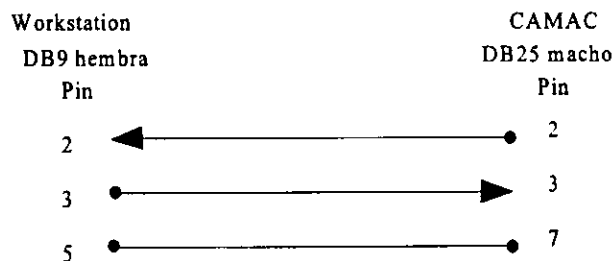


Fig. 3.3. Configuración del cable.

Si fuera necesario, o de alguna utilidad, podrían habilitarse las líneas RTS *request to send* y DTR *data terminal ready* en el CAMAC.

3.4.- Conexión lógica entre la computadora y el controlador del sistema de adquisición de datos

3.4.1 El *driver* y el archivo especial.

Una vez que se tiene resuelta la conexión física, es necesario ocuparse de los otros dos requisitos que se plantearon: el *driver* y el archivo especial asociados al puerto.

¿Cómo saber cuál es el *driver* que requiere el puerto RS232 con el que se trabajará? Una forma de saberlo, es consultando el manual '*Installing Peripherals*' correspondiente estrictamente al modelo de máquina que se tiene. El fabricante tiene la responsabilidad de hablar en él sobre los *drivers* correspondientes a cada interfaz. Aquí interesa la parte en donde se hable de conectar modems (ya que se conectará un modem nulo). En ese manual deberá venir la información explícita de cuáles son los archivos de configuración que deben modificarse (que deben ser por lo menos dos, uno que contiene todos los *drivers* disponibles en el sistema, y otro que contiene la selección de *drivers* que serán incorporados al *kernel*); así como los pasos a seguir para la reconfiguración del *kernel*. Otros documentos que pueden ser de utilidad son los manuales de administración del sistema y la sección (1M) del manual de Referencia, ya que incluyen información acerca del uso de periféricos.

El sistema HP-UX cuenta por default con un archivo llamado */etc/master* que contiene toda la información relacionada con los dispositivos que el sistema puede soportar (en cuanto a subsistemas, *file systems*, *I/O drivers*, *pseudo drivers*, parámetros ajustables y asignación del *major number*).

Este archivo es suministrado como parte integrante del sistema operativo HP-UX y es sólo un archivo de referencia. Tal archivo, es también conocido como tabla maestra de información sobre dispositivos *-master device information table-* [3.4].

Este es el archivo básico para manejo de dispositivos; sin embargo, para la reconfiguración del *kernel* es necesario otro archivo, el que contendrá la selección de características que serán

válidas en el sistema y el subconjunto de *drivers* que serán parte del *kernel*. El nombre por omisión de este archivo es `/etc/conf/dfile`. El superusuario deberá generarlo según las necesidades que se tendrán. La generación de este archivo y los demás pasos para la reconfiguración del *kernel* pueden llevarse a cabo de dos maneras: por comandos HP-UX o por SAM.

Los comandos involucrados en la reconfiguración del *kernel* son `/etc/config` y `make`. La información detallada de cómo se usan estos comandos viene en el manual '*Installing Peripherals*' de la serie 700. En la fig. 3.4 se muestra un diagrama de bloques de la funcionalidad de tales comandos.

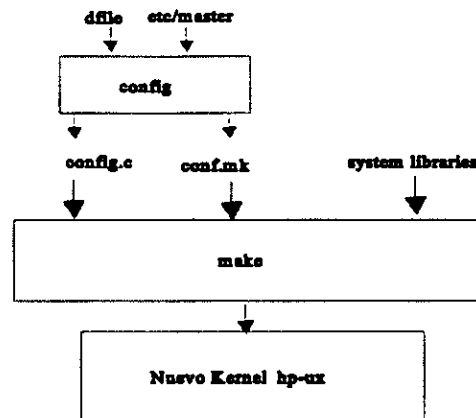


Fig. 3.4. Comandos para la configuración del *kernel*.

El SAM es una herramienta muy amigable para ayudar al superusuario en la reconfiguración del *kernel*, ya que cuenta con una serie de menús que se explican por sí mismos. De cualquier manera, puede utilizarse el manual correspondiente en caso de duda.

Para ver la lista de *drivers* configurados en el *kernel* y sus *major numbers* puede ejecutarse el comando `lsdev (/etc/lsdev)`; éste lista los *drivers* en el sistema si es que están disponibles para invocación a través de archivos especiales.

Algunos *drivers* disponibles en el sistema sólo pueden ser usados a través de otros *drivers* y tratar de usarlos directamente desde un archivo especial puede producir resultados inesperados.

Así pues, revisando el archivo `dfile` se encontró que el `SERIAL DEVICE DRIVER= asio0`. Ese es el *driver* de interés. El siguiente paso una vez que se ha verificado que el *driver* está configurado en el *kernel*, es crear el o los archivos de dispositivo necesarios.

La importancia de los archivos de dispositivo es que, por un lado, apuntan a la dirección

lógica de los periféricos o dispositivos y por otro lado, al *driver* que los manejará. Esta información estará codificada en el *device number* compuesto por dos números: el *major number* que apunta a la rutina del *driver* en una petición I/O y el *minor number* que apunta a la dirección de *hardware* y es usado por el *driver* para obtener información de cómo manejar los datos y qué dispositivos atender.

HP-UX en series 700 sólo maneja dos tipos de archivo de dispositivo: de bloque y de caracter, de modo que todos los dispositivos de I/O en el sistema caen en uno de los dos tipos.

Para la aplicación en este trabajo, el archivo especial deberá ser de caracter ya que como se vió en el capítulo 2, se necesita transferir bloques de datos de tamaño variable.

La creación de archivos especiales se puede hacer también ya sea por SAM o por comandos HP-UX.

SAM configura la mayoría de los periféricos soportados por HP-UX, para esos dispositivos que SAM soporta, crea los archivos de dispositivo y añade los *drivers* al *kernel* si es necesario. Los periféricos que no son soportados por SAM son:

- Tarjetas gráficas de interfaz y *graphics displays*
- Impresoras HP C2753A Modelo F100
- Manejadores de discos flexibles
- Scanjets y digitalizadores
- **Instrumentos.**

Para ellos, SAM no crea los archivos de dispositivo, pero puede usarse para añadir los *drivers* necesarios.

El comando HP-UX para la generación de archivos especiales es `mknod` [3.5] (`/etc/conf/mknod`). Son tres los tipos de archivos que pueden crearse con este comando:

- Archivos de dispositivo de caracter.
- Archivos de dispositivo de bloque.
- Archivos FIFO también conocidos como pipe.

Para crear un archivo especial de caracter, se deben dar los siguientes parámetros: nombre.- es el *pathname* donde se creará el archivo. El nuevo archivo tiene por default los permisos de escritura y lectura para todos los usuarios (066).

Tipo.- es el tipo de archivo (c para caracter).

Major.- el *major number* del *driver* que manejará el dispositivo.

Minor.- el *minor number* que especifica la dirección de *hardware*. El *minor number* es absolutamente dependiente del sistema y debe contarse con el manual *Installing Peripherals* del sistema para conocerlo.

Sólo los usuarios con privilegios apropiados pueden usar *mknod* para crear archivos de dispositivo. El *script* *mkdev.sys* que se usa opcionalmente en la reconfiguración del *kernel* consta de sentencias *mknod* para la generación de entradas en el directorio */dev*.

En los sistemas 700 los *minor numbers* son código hexadecimal de seis dígitos (24 bits) de la forma OxS_1S_2FDDD

Ox indica que es formato hexadecimal.

S_1 es el número de *System Bus Module* y puede ser:

1: Std. Graphics

2: Core I/O

4: EISA Adapter

8: Processor

9: Memory

S_2 es el número de *slot* de la tarjeta EISA -*EISA slot number*- (es 0 si el *system bus module* es diferente de 4).

F es el *Function number* para core I/O , especifica una función en particular de la tarjeta de interfaz multifuncional core. (es 0 si el *system bus module* es diferente de dos).

1: SCSI

2: LAN

3: HIL

4: Puerto serial 1 (etiquetado como A)

5: Puerto serial 2 (etiquetado como B)

6: Puerto paralelo

DDD Son 12 bits específicos del *driver*. Su significado especial está incluido en el manual *Installing Peripherals*.

Se puede usar el comando `ioscan [3.6] (/etc/ioscan)` para inspeccionar el *hardware* con el que cuenta el sistema en cuanto a tarjetas de interfaz, discos y cintas. `ioscan -f` produce la siguiente salida:

Class	H/W Path	Driver	H/W Status	S/W Status
graphics	1.0.0	graph3	ok(0x785)	ok
scsi	2.0.1	c700	ok(0x7082)	ok
disk	2.0.1.6.0	scsi	ok(0x202)	ok
lan	2.0.2	lan01	ok(0x708a)	ok
serial	2.0.4	asio0	ok(0x708c)	ok
parallel	2.0.6	parallel	ok(0x7074)	ok
audio	2.0.8	audio	ok(0x707b)	ok
floppy	2.0.10	pcfdc	ok(0x7083)	ok
disk	2.0.10.1.0	pcfloppy	ok(0x0)	ok
ps2	2.0.11	ps2	ok(0x7084)	ok
ps2	2.0.12	ps2	ok(0x7084)	ok

Su formato de salida es como sigue:

- Class Es la clase de tarjeta de interfaz a la que el dispositivo está conectado.
- Hw path Especifica la dirección de los componentes de *hardware*. Consiste de una cadena de dígitos en formato hexadecimal separado por periodos. Coinciden los primeros tres dígitos con los del *minor number*.
- Driver Especifica el *driver* controlador de esa dirección de *hardware*.
- Hw status Identifica a los componentes de *hardware*, es un valor hexadecimal que mostrará qué clase de *hardware* está conectado. Ese identificador es tomado del *driver*.
- Sw status Se refiere al estado del *driver* controlador del componente de *hardware*.

Para lo que interesa, los últimos 3 dígitos del *minor number* son de la forma:

DDD : 0000 abcd efgh

donde

a y b definen el *receive FIFO trigger level*, mientras que c y d definen el *transmit FIFO limit* de la siguiente manera:

a	b	NIVEL	c	d	LIMITE
0	0	1	0	0	1
0	1	4	0	1	4
1	0	8	1	0	8
1	1	14	1	1	12

Es importante hacer notar que el *receive FIFO trigger level* se usa para definir el nivel en el cual una interrupción es detectada en el sistema. Un valor bajo permite reaccionar más rápidamente a la recepción de caracteres. Por otro lado, el *transmit FIFO limit* define el número de caracteres que son aceptados exitosamente en el *transmit FIFO*. Un valor pequeño permite mayor sensibilidad en el control de flujo (XON, XOFF). Tales bits deben definirse apropiadamente dependiendo del uso que se dará al puerto serial, de las necesidades de respuesta y de la tolerancia del dispositivo a la recepción de datos después del control de flujo. Por otro lado:

e=0 define NO usar control de flujo por las líneas RTS y CTS.

e=1 define SI usar control de flujo por las líneas RTS y CTS.

f=0 define una conexión utilizando las líneas de control de modem.

f=1 define una conexión que no usa las líneas de control de modem *direct connect*

g=0 define modo de control de modem según *US modems*

g=1 define modo de control de modem CCITT

h=0 define el acceso al puerto como *call-in*

h=1 define el acceso al puerto como *call-out*

Entonces, el *minor number* asociado al puerto RS232 en cuestión es 0x204004 y el *major* es 1 ya que es el puerto serial 1 que forma parte del *system bus module CORE I/O*, la conexión será directa y como si fuera un US modem.

En conclusión, *asio0* es el *driver* que maneja al puerto serial RS232, está ya configurado en el sistema, con su *minor number* se puede crear un nuevo archivo especial y/o verificar si se

tiene alguno en existencia asociado con ese puerto.

Si se lista el directorio /dev, se encuentra que el primer caracter del primer campo constituye el tipo de archivo (bloque, caracter o directorio), lo siguiente son los permisos. El segundo campo es la liga, el tercer campo es el propietario, el cuarto campo es el grupo al que pertenece, el quinto campo es el *major number*, lo que sigue es la fecha y hora de ultimo acceso, y el último campo es el nombre del archivo. Una entrada ejemplo del directorio /dev sería:

```
crw-r--r--    1    lp    bin    11    0x206002    Apr  22    1994    plt_parallel
```

En el presente sistema, se tienen 3 archivos especiales de caracter asociados con el puerto:

```
/dev/tty00      pensado para terminal  
/dev/plt_rs232_a  pensado para ploter serial  
/dev/ptr_rs232_a  pensado para impresora serial
```

se podría usar cualquiera de ellos para entablar comunicación con el CAMAC.

Así pues, ya se tienen todos los elementos físicos y lógicos necesarios para establecer comunicación con el CAMAC. En este punto es necesario recordar que el *driver* se considerará como una caja negra sensible a ciertas llamadas al sistema. Lo que sigue es saber cómo reacciona ante esas llamadas al sistema y qué es lo que puede esperarse de su desempeño.

3.4.2.- *Entry points* y llamadas al sistema.

Como se vió, la única herramienta para hacer peticiones al *kernel* son las llamadas al sistema. Cuando una *system call* se ejecuta para realizar una operación en un archivo especial, la subrutina apropiada del *driver* es llamada, el sistema operativo selecciona el *entry point* en una tabla que corresponde al *driver* y al *entry point*. Así por ejemplo, en C no existe diferencia entre una *system call* y cualquier otra llamada a función

```
var=read(fd,buf,num);
```

también se pueden hacer llamadas al sistema desde otros lenguajes de programación.

Tras una llamada al sistema, el sistema operativo llama a un *entry point*, pasando así el

control al *driver*. El *driver* examina los parámetros que recibe y las estructuras de datos del *kernel* para saber qué hacer. Tales *entry points* son definidos por el diseñador del *driver*, algunos de ellos son indispensables en todo *driver* por ejemplo:

- *open entry point*.- Es la rutina que será ejecutada tras una llamada al sistema `open()` sobre el archivo especial. Su labor deberá ser validar el *device number* interpretando el *minor number* y hacer la inicialización previa a un `read()` o `write()` de acuerdo con la información codificada en el *minor*. También definirá algunos parámetros de *hardware* necesarios para la comunicación y deberá resolver *opens* múltiples.
- *read entry point*.- Es responsable de obtener del dispositivo los datos requeridos y transferirlos al proceso del usuario tras una llamada al sistema `read()`.
- *write entry point*.- Es responsable de obtener del proceso del usuario la información que será escrita, ordenar al dispositivo que acepte los datos y administrar o manejar la transferencia tras una llamada al sistema `write()`.
- *close entry point*.- Esta rutina deberá encargarse de atender las llamadas al sistema `close()` y en su momento, esperar a que todas las salidas terminen y hasta entonces liberar los recursos, permitir inicializar el dispositivo, etc.

Y así pueden definirse muchos más *entry points* por ejemplo:

- *intr entry point*.- Para manejo de interrupciones de hardware.
- *ioctl entry point*.- Es la rutina que se ejecutará tras una llamada al sistema `ioctl()`. Este tipo de llamadas se usan para hacer peticiones especiales al *driver* o para obtener información sobre la configuración y status del dispositivo y del *driver*. El *driver* debe aceptar los detalles de la petición `ioctl` y realizar las operaciones necesarias (o en su caso rechazarlas con un mensaje de error).

Hasta este momento se ha dicho que el *driver* es una caja negra que configura la comunicación con el dispositivo en cierto estado por omisión; la forma en que responde a cada *system call* es decisión del diseñador; sin embargo es posible cambiar esos parámetros por omisión para cubrir hasta cierta medida las necesidades de los usuarios. Esa es la función de la llamada al sistema `ioctl()`. La forma general de una *ioctl system call* es :

```
retval=ioctl(fildes, request, arg);
```

donde *fd* es el número correspondiente al *file descriptor* o identificador del archivo, *request* es el tipo de petición *ioctl()* y *arg* contiene información adicional que puede ser un entero o un puntero a un buffer (para pasar información en alguna dirección).

Se pueden definir diversas peticiones *ioctl* para cada dispositivo.

El *ioctl entry point* es un gran switch que bifurca dependiendo de la petición *ioctl* que reciba. Así por ejemplo pueden definirse *ioctl read* y *write*, de modo que ignorando los tradicionales *read* y *write entry points*, uno puede hacer lo que quiera (excepto *open* o *close*).

Regresando al punto de interés, el problema es saber cuáles son los parámetros por default con que trabaja *asio0*, si esos satisfacen nuestras necesidades y si no, cómo pueden cambiarse.

En todos los sistemas UNIX puede encontrarse un archivo llamado *termio.h* que define una estructura de datos que contiene alrededor de 50 *flag bits* diseñados especialmente para definir los parámetros a usar en transferencias asíncronas tales como *baud rate*, tamaño de palabra, etc., así como la forma en que se procesarán los caracteres de entrada y de salida. El *driver* trabaja con ciertos parámetros por omisión y cambiarlos se hace a través de llamadas *ioctl* y en el caso de transferencias asíncronas se usa la estructura *termio* (*general terminal interface*). Todos los puertos de comunicación asíncrona usan la misma interfaz general, sin importar el *hardware* que esté involucrado (modems, terminales, etc).

Más adelante se verá que *termio* es suficientemente versátil como para satisfacer las necesidades de este trabajo con el CAMAC. Básicamente esas necesidades se refieren a hacer compatibles los parámetros de transmisión.

3.4.3.- La configuración del puerto y *termio.h*.

En la página de manual *termio* (ver referencia [3.3]) se puede encontrar una descripción detallada de las facilidades que se obtienen usando la estructura *termio*. Aquí se intenta dar un apunte. *termio* configurará dispositivos de comunicación asíncrona tales como terminales, modems y dispositivos conectados en serie por RS232. Dicha configuración se dará a través del archivo especial asociado al dispositivo (también llamado *terminal file*).

La estructura *termio* esta definida en el archivo *termio.h* y tiene la forma:

```

#define NCC 8
struct termio{
    unsigned short c_iflag;    /** input modes **/
    unsigned short c_oflag;    /** output modes **/
    unsigned short c_cflag;    /** control modes **/
    unsigned short c_lflag;    /** local modes **/
    char c_line;              /** line discipline **/
    unsigned char c_cc[NCC];   /** control characters **/
};

```

cada uno de los componentes de la estructura tiene asociados los parámetros específicos del puerto, los cuales serán codificados en bits (llamados 'flag bits').

Así, `c_iflag` describe la forma en que serán procesados los caracteres de entrada, `c_oflag` describe la forma en que serán procesados los caracteres de salida, `c_cflag` describe los parámetros de *hardware*, `c_lflag` describe ciertas funciones especiales, `c_line` es la conexión con los *entry points* del *driver* (**no es configurable**), `c_cc` define los caracteres de control que se usarán en una transmisión cualquiera. Los bits son:

Para <code>c_iflag</code>	<code>IGNBRK</code>	ignorar condición de BREAK
	<code>BRKINT</code>	enviar señal de interrupción en BREAK
	<code>IGNPAR</code>	ignorar caracteres con errores de paridad
	<code>PARMRK</code>	marcar errores de paridad
	<code>INPCK</code>	chechar paridad en la entrada
	<code>ISTRIP</code>	cambiar caracter a 7 bits en la entrada
	<code>INLCR</code>	cambiar NL a CR en la entrada
	<code>IGNCR</code>	ignorar CR en la entrada
	<code>ICRNL</code>	cambiar CR a NL en la entrada
	<code>IUCLC</code>	cambiar mayúsculas a minúsculas en la entrada
	<code>IXON</code>	reconocer caracteres START y STOP para control de flujo
	<code>IXANY</code>	reanudar flujo al recibir cualquier caracter
	<code>IXOFF</code>	enviar caracteres START y STOP para control de flujo

el default es todos los bits de `c_iflag` apagados.

Para <code>c_oflag</code>	OPOST	postprocesar salida
	OLCUC	cambiar minúsculas a mayúsculas en la salida
	ONLCR	cambiar NL a CR en la salida
	OCRNL	cambiar CR a NL en la salida
	ONOCR	no enviar CR si está en la columna 0
	ONLRET	cambiar NL a RETURN en la salida
	OFILL	usar caracteres de llenado para retrasar salida
	OFDEL	definir caracter de llenado
	NLDLY	selecciona tipo de retraso en caso de NL <i>new line</i>
	NL0	ningún retraso
	NL1	retraso tipo 1 en NL
	CRDLY	selecciona tipo de retraso en caso de CR <i>carriage return</i>
	CR0	ningún retraso
	CR1	retraso tipo 1 en CR
	CR2	retraso tipo 2 en CR
	CR3	retraso tipo 3 en CR
	TABDLY	selecciona tipo de retraso en TAB tabulador horizontal
	TAB0	ningún retraso
	TAB1	retraso tipo 1 en TAB
	TAB2	retraso tipo 2 en TAB
	TAB3	convertir TAB en espacios
	BSDLY	selecciona tipo de retraso en caso de BS <i>backspace</i>
	BS0	ningún retraso
	BS1	retraso tipo 1 en BS
	VTDLY	selecciona tipo de retraso en VT tabulador vertical
	VT0	ningún retraso
	VT1	retraso tipo 1 en VT
	FFDLY	selecciona tipo de retraso en caso de FF <i>form feed</i>

FF0	ningún retraso
FF1	retraso tipo 1 en FF

el default es todos los bits de `c_oflag` apagados.

Para <code>c_cflag</code>	<code>Bnn</code>	selecciona <i>baud rate</i> ; <i>nn</i> puede variar de 50 a 38400
	<code>CSss</code>	selecciona tamaño de caracter, <i>ss</i> puede variar de 5 a 8
	<code>CSTOPB</code>	selecciona bits de parada
	<code>CREAD</code>	permitir entrada de caracteres
	<code>PARENB</code>	generar bit de paridad en la salida
	<code>PARODD</code>	seleccionar tipo de paridad
	<code>HUPCL</code>	colgar en último <i>close()</i>
	<code>CLOCAL</code>	seleccionar línea local o <i>dial-up</i>
	<code>LOBLK</code>	reservado

el default es `B300`, `CS8`, `CREAD` y `HUPCL` prendidos, y los demás bits de `c_cflag` apagados.

Para <code>c_lflag</code>	<code>ISIG</code>	reconocer señales en la entrada
	<code>ICANON</code>	procesar en forma canónica
	<code>XCASE</code>	reconocer mayúsculas como la combinación de <i>\minuscúla</i>
	<code>ECHO</code>	enviar echo
	<code>ECHOE</code>	enviar echo del caracter ERASE
	<code>ECHOK</code>	enviar echo del caracter KILL
	<code>ECHONL</code>	enviar echo del caracter NL
	<code>NOFLSH</code>	no limpiar bufer en caso de INT, QUIT o SUSP
	<code>TOSTOP</code>	negar la petición <code>write()</code> en caso de no venir del foreground process group.
	<code>IEXTEN</code>	habilitar extended functions

el default es todos los bits de `c_lflag` apagados.

Para efectos de comunicación con el CAMAC se han agrupado para reconocer los bits que tengan que ver con :

definir bits de parada (o bits de stop)

definir número de bits por caracter

generación y recepción de paridad

habilitar chequeo de paridad

definir tipo de paridad

definir baud rate

manejo de las líneas del RS232

manejo del BREAK

control de flujo

procesamiento de las entradas

procesamiento de las salidas

otras funciones especiales

El siguiente cuadro muestra cada uno de los bits configurables de termio según los grupos mencionados arriba, la función que realizan si están prendidos o apagados y sus correspondencias con la configuración CAMAC.

GRUPO	MODE	BIT	PRENDIDO	APAGADO	OPC. CAMAC	NOTAS
Bits de stop	c_cflag	CSTOP	2 bits de stop	1 bit de stop	1, 1.5 ó 2 bits de stop	Debe elegirse igual en ambos. En el CAMAC configurar ASCII decimal
Bits p/caracter (sin incluir el de paridad)	c_cflag	CS5	5 bits p/caracter	-	7 u 8 bits p/caracter	Debe elegirse igual en ambos
		CS6	6 bits p/caracter	-		
		CS7	7 bits p/caracter	-		
		CS8	8 bits p/caracter	-		
Generación y recepción de paridad	c_cflag	PARENB	generar bit de paridad y considerar un bit de paridad en los caracteres de enteeda.	no generar paridad	habilitar o deshabilitar paridad	atender los bits PARENB, INPCK, IGNPAR y PARMRK en lo referente a generación y chequeo de paridad: para deshabilitarla, apagar PARENB.
Chequeo de paridad	c_iflag	INPCK	Checar paridad en los caracteres de entrada.	no chequear paridad		
	c_iflag	IGNPAR	Ignorar caracteres con error de paridad.	leer caracteres con error de paridad		
Tipo de paridad Baudrate	c_iflag	PARMRK	Marcar errores de paridad	No marcar		
	c_cflag	PARODD	Paridad impar	Paridad par		
	c_cflag	B0	Colgar (es ignorado en direct connect)	-	paridad par o impar	Debe elegirse igual en ambos.
		B50	50 bauds	-	A elegir entre 50, 75, 110, 134.5, 150, 300, 600,	
		B75	75 bauds	-	1200, 1800, 2000, 2400,	
		B110	110 bauds	-	3600, 4800, 7200, 9600 y 19200 bauds.	
		B134	134 bauds	-		
		B150	150 bauds	-		
		B200	200 bauds	-		
		B300	300 bauds	-		
	B600	600 bauds	-			
	B900	900 bauds	-			
	B1200	1200 bauds	-			
	B1800	1800 bauds	-			
	B2400	2400 bauds	-			
	B3600	3600 bauds	-			
	B4800	4800 bauds	-			
	B7200	7200 bauds	-			

I Bits de terminio y configuración CAMAC.

GRUPO	MODE	BIT	PRENDIDO	APAGADO	OPC. CAMAC	NOTAS
Baudrate	c_cflag	B9600 B19200 B38400 EXTA EXTB	9600 bauds 19200 bauds 38400 bauds - -	- - - - -	A elegir entre 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600 y 19200 bauds.	Debe elegirse igual en ambos. Adicionalmente, el puerto serial de la HP ofrece los siguientes valores de baudrate: 57600, 115200, 250400 y 460800; para los cuales necesita un convertidor a RS422.
Lineas del RS-232	c_cflag	CLOCAL	la conexión no depende de las líneas de estado del modem. No tiene efecto en archivos especiales direct connect.	las líneas de estado del modem son monitoreadas	Usar estandar RS-232 o RS-422; también se pueden habilitar o deshabilitar las líneas RTS y DTR.	Debe elegirse el estandar RS-232 y deshabilitar las líneas RTS y DTR en CAMAC.
	c_cflag	HUPCL	Cerrar la conexión en el último close(). No tiene efecto en archivos especiales direct connect.	No cerrar la conexión.		
Manejo del break	c_iflag	IGNBRK	Ignorar BREAK	leer BREAK		Elegir IGNBRK Y BRKINT según las necesidades de la aplicación.
Control de flujo	c_iflag c_iflag	BRKINT IXON	Enviar SIGINT en BREAK Reconocer caracteres START y STOP para parar y/o reanudar la salida.	leer BREAK como NUL leer START y STOP como caracteres normales	Automáticamente envía y reconoce los caracteres XON y XOFF para control de flujo.	Prender IXON
	c_iflag c_iflag c_iflag	IXANY y IEXTEN IXOFF	Reanudar salida al recibir cualquier caracter Enviar caracteres STOP y START para controlar flujo	No reanudar salida al recibir cualquier caracter No enviar caracteres STOP y START para controlar flujo		No prender IXANY pues se busca control de flujo solo a través de XON y XOFF Prender IXOFF
Procesamiento de las entradas	c_cflag	CREAD	Permite recibir caracteres	No permite recibir caracteres	-	Siempre prender CREAD
	c_iflag	ISTRIP	Cambia caracteres válidos de 8bits a 7bits en la entrada	No cambiarlos	-	Apagar ISTRIP

2 Bits de terminio y configuración CAMAC (continuación)

GRUPO	MODE	BIT	PRENDIDO	APAGADO	OPC. CAMAC	NOTAS
Procesamiento de las entradas	c_iflag	INLCR	Convierte NL en CR en la entrada	No lo convierte	-	Apagar INLCR, IGNCR, ICRNL, IUCLC ya que deben leerse los caracteres ASCII tal cual.
	c_iflag	IGNCR	Ignora CR en la entrada	Leerlo	-	
	c_iflag	ICRNL	Convierte CR en NL	No lo convierte	-	
	c_iflag	IUCLC	Convierte mayúsculas en minúsculas.	Toma las mayúsculas tal cual	-	
	c_iflag	IEXTEN	Reconocer caracteres especiales INT, QUIT, SUSP.	Leer dichos caracteres como caracteres normales	-	Prender ISIG como una opción para manejo del BREAK, según las necesidades de la aplicación; del mismo modo considerar NOFLSH.
	c_iflag	NOFLSH	No limpiar las colas de entrada y salida en caso INT	Limpiarlas	-	Apagar XCASE.
Procesamiento en las salidas	c_lflag	XCASE	Reconocer mayúsculas como la secuencia de \minusc	Solo reconocer mayúsculas de tal cual	-	
	c_lflag	IEXTEN	Procesamiento canónico Postprocesar salida	Procesamiento no canónico No postprocesar salida	-	Prender ICANON según necesidades de la aplicación. Siempre apagar OPOST pues se necesita enviar los caracteres ASCII tal cual.
	c_oflag	ICANON	Cambiar minúsculas a mayúsculas	Enviar minúsculas tal cual	-	OPOST apagado invalida los siguientes bits de postprocesamiento.
		ONLCR	Cambiar NL a CR	Enviar NL tal cual	-	Los retrasos fueron pensados para dar tiempo a que el dispositivo realice algún movimiento mecánico en caso de RETURN, TAB, etc. Si se apagara OFILL, se usarían retrasos de tiempo pero en la serie 700 no están permitidos.
		OCRNL	Cambiar CR a NL	Enviar CR tal cual	-	Apagar OFILL y demás bits de retraso ya que el CAMAC no los necesita.
		ONOCR	No enviar CR en columna 0	Enviar CR aun en col 0	-	
		ONLRET	Cambiar NL a RETURN	Enviar NL tal cual	-	
		OFILL	Usar caracteres de llenado	No usar caracter de llenado	-	
		OFDEL	Caracter de llenado: DEL	Caracter de llenado: NUL	-	
		NL0	NL sin retraso	-	-	
		NL1	2 caract. de llenado en NL	-	-	
		CR0	CR sin retraso	-	-	
		*CR1	2 caract de llenado en CR	-	-	
		CR2	4 caract de llenado en CR	-	-	
		CR3	-	-	-	
		TAB0	TAB sin retraso	-	-	
		TAB1	2 caract de retraso en TAB	-	-	
	TAB2	2 caract de retraso en TAB	-	-		
	TAB3	Cambiar TAB por espacios	-	-		
	BS0	BS sin retraso	-	-		
	BS1	1 caracter de llenado en BS	-	-		

3 Bits de terminio y configuración CAMAC (continuación)

GRUPO	MODE	BIT	PRENDIDO	APAGADO	OPC. CAMAC	NOTAS
Procesamiento en las salidas	c_oflag	VT0	Tabulador vertical sin retraso	-	-	Apagar OFILL y demas bits de retraso ya que el CAMAC no los necesita.
	c_oflag	VT1	Retraso tipo l en VT	-		
	c_oflag	FF0	FF form feed sin retraso	-		
	c_oflag	FF1	Retraso tipo l en FF	-		
	c_lflag	XCASE y IEXTEN	Preceder mayúsculas por el caracter \	Enviar mayúsculas tal cual		Apagar XCASE ya que se requiere enviar los caracteres ASCII tal cual. Prender TOSTOP si así lo requiere la aplicación.
Funciones especiales	c_lflag	TOSTOP	Denegar el write() a procesos que no estén en el foreground process group.	No denegar el write()	-	
	c_lflag	ECHO	Enviar un echo al dispositivo de los caracteres recibidos	No enviar echo		Apagar ECHO, ECHOE, ECHOK y ECHONL ya que para efectos del CAMAC no es necesario enviar echos. En caso de dispositivos que sí lo requieran es necesario puntualizar que los bits ECHOE, ECHOK y ECHONL dependen del estado del bit ICANON en c_lflag.
	c_lflag	ECHOE ECHOK ECHONL	Enviar echo del ERASE Enviar echo del KILL Enviar echo del NL	no enviar echo de ERASE no enviar echo de KILL no enviar echo de NL	-	
	c_cflag	LOBLK	Bit reservado	-		El bit LOBLK está reservado para usar por shl()

4 Bits de terminio y configuración CAMAC (continuación)

Esos son los parámetros configurables, se asignan en la estructura termio de la siguiente manera según el lenguaje C (este es sólo un ejemplo):

```
#include </usr/include/sys/termio.h>
struct termio tbuf;
:
tbuf.c_iflag=IXON|IXOFF;
tbuf.c_oflag=0;
tbuf.c_cflag=B19200|CS8|CSTOPB|CREAD|PARODD;
tbuf.c_lflag=0;
tbuf.c_cc[4]=1;
:
:
```

Se debe hacer hincapié en el funcionamiento e interrelación de los siguientes bits:

REFERENTES A PARIDAD

PARODD.- Es un bit independiente, define el tipo de paridad que se generará y/o checará.

PARENB	INPCK	IGNPAR	PARMRK	RESULTADO
prendido	-	-	-	Un bit de paridad es añadido a cada caracter de salida y en los caracteres de entrada se reconoce un bit de paridad.
prendido	prendido	-	-	Se genera bit de paridad y se checa la paridad en la entrada.
prendido	apagado	-	-	Se genera paridad y se acepta un bit de paridad, sin embargo no se checa la paridad, se supone que no existen errores.
prendido	prendido	prendido	-	Se genera y recibe paridad, se checa y los caracteres con error son ignorados (excepto BREAK).
prendido	prendido	apagado	prendido	Los caracteres X con errores de paridad son leídos como la secuencia \377 \0 X donde X

				es el caracter con error (un caracter válido \377 es \377 \377).
prendido	prendido	apagado	apagado	Caracteres con error de paridad son leídos como \0.
apagado	-	-	-	No genera bit de paridad ni puede checar paridad en la entrada.

Nota: los valores arriba mencionados están en octal.

REFERENTES AL BREAK

Si la HP recibe un BREAK (ASCII NUL), limpia tanto la cola de entrada como la cola de salida y genera una señal SIGINT al *foreground process group* si es la *controlling terminal*.

IGNBRK	BRKINT	RESULTADO
prendido	-	Ignorar BREAK (caracter NUL).
apagado	prendido	Limpia las colas de entrada y salida y envía SIGINT al proceso si el dispositivo es la <i>controlling terminal</i> .
apagado	apagado	El BREAK es procesado como un caracter cualquiera (\0)

REFERENTE A CONTROL DE PROCESOS

TOSTOP.- Este es un bit independiente , se usa para evitar operaciones write() en la *controlling terminal* si el proceso no pertenece al *foreground process group*. (Restringe el acceso al dispositivo).

Es muy importante mencionar que todo archivo asociado a un puerto serial tiene asociada una cola de entrada *input queue* que puede guardar un número máximo de caracteres igual al valor del sistema MAX_INPUT, antes de la lectura.

El procesamiento de los datos de entrada se puede hacer de dos maneras: en modo canónico y en modo no canónico (definido en el bit ICANON).

Modo Canónico.- las entradas se procesan en unidades de líneas donde una línea se delimita por el caracter NL *new line*, EOF *end of file* ó EOT *end of transmission* , así que una

lectura `read()` no regresa hasta que una línea completa ha sido recibida. El valor del sistema `MAX_CANON` es el máximo número de caracteres en una línea. En este modo, cumplen con su función los caracteres `ERASE` y `KILL` que se definen en `c_cc`. Este modo es de principal utilidad en el manejo de terminales.

Modo No Canónico.- Las entradas no se procesan en líneas y los caracteres `ERASE` y `KILL` no tienen efecto, en cambio, las peticiones `read()` se satisfacen directamente de la *input queue* y se requiere la definición de dos valores: `MIN` y `TIME`; los cuales dictan la forma en que se dispondrá de los caracteres de entrada.

`MIN`.- Es el número mínimo de caracteres recibidos antes de que `read()` regrese con éxito. Se define en `c_cc[4]` y debe ser un valor menor o igual a `MAX_INPUT`.

`TIME`.-Es un contador de tiempo de 0.10 seg de granularidad.

CASO A. `MIN>0, TIME>0`

`TIME` se activa al recibir el primer caracter y se inicializa al recibir el siguiente. Si en la cola de entrada se tienen disponibles al menos `MIN` caracteres antes de que `TIME` expire, el `read` se satisface. Si expira, `read()` regresa con los caracteres recibidos. Note que si `TIME` expira, al menos un caracter será regresado.

CASO B. `MIN>0, TIME=0`

El `read()` sólo se satisface si se tienen disponibles en la *input queue* al menos `MIN` caracteres.

CASO C. `MIN=0, TIME>0`

`TIME` se activa al momento de ejecutarse el `read()`, éste se satisface tan pronto como se reciba un caracter o `TIME` expire.

CASO D. `MIN=0, TIME=0`

El `read()` obtiene lo que sea menor: o el número de caracteres pedidos en el `read()` o el número de caracteres disponibles en la *input queue* (no espera por más caracteres). Si no existen caracteres disponibles, el `read()` regresa con una cuenta de 0 caracteres.

Note que los casos A y B se usan para grandes transferencias de datos y los casos C y D se usan para transferencias de un sólo caracter y/o para verificar si se han recibido caracteres.

Este modo No Canónico se usa para transferencias con modems, scanners, ploters y otros dispositivos.

Los caracteres especiales que se definen en el vector `c_cc` de la estructura `termio` tienen ciertas funciones específicas ya sea que sean recibidos o enviados. Todos, excepto `NL`, `CR`, `START`, y `STOP` pueden ser cambiados o deshabilitados. Para deshabilitarlos debe ponerse su valor en `_POSIX_VDISABLE`. Sus funciones se describen a continuación.

NOMBRE	DEFAULT	ENTRADA/ SALIDA	FUNCION
<code>INTR</code>	<code>DEL</code>	ENTRADA	Es reconocido si <code>ISIG</code> está prendido; genera una señal <code>SIGINT</code> .
<code>QUIT</code>	<code>FS</code>	ENTRADA	Es reconocido si <code>ISIG</code> está prendido; genera una señal <code>SIGQUIT</code> .
<code>SWTCH</code>	<code>NULL</code>	ENTRADA	NINGUNA FUNCION ESPECIAL.
<code>ERASE</code>	<code>'#'</code>	ENTRADA	Es reconocido si <code>ICANON</code> está prendido, borra el caracter precedente.
<code>KILL</code>	<code>'@'</code>	ENTRADA	Es reconocido si <code>ICANON</code> está prendido, borra la línea entera.
<code>EOF</code>	<code>EOT</code>	ENTRADA	Es reconocido si <code>ICANON</code> está prendido, todos los caracteres son pasados al proceso. Es el caracter de fin de transmisión. Si <code>ICANON</code> está apagado, representa a <code>MIN</code> .
<code>NL</code>	<code>LF</code>	ENTRADA	Es reconocido si <code>ICANON</code> está prendido, actúa como delimitador de línea.
<code>EOL</code>	<code>NULL</code>	ENTRADA	Es reconocido si <code>ICANON</code> está prendido, actúa también como delimitador de línea. Si <code>ICANON</code> está apagado, representa a <code>TIME</code> .
<code>SUSP</code>	<code>^Z</code>	ENTRADA	Es reconocido si <code>ISIG</code> está prendido, genera una señal <code>SIGTSTP</code> .
<code>STOP</code>	<code>DC3</code>	ENT y SAL	Es reconocido si <code>IXON</code> está prendido, suspende la salida temporalmente para control de flujo. También es reconocido si <code>IXOFF</code> está prendido, en este caso

START	DC1	ENT y SAL	envía un caracter STOP cuando se está alcanzando un límite en las entradas que puede rebasar el bufer. Es reconocido si IXON está prendido, reanuda una salida suspendida. También es reconocido si IXOFF está prendido, en este caso envía un caracter START en cuanto la cola de entrada se drene hasta cierto nivel y esté lista para seguir recibiendo caracteres.
CR	CR	ENTRADA	Es reconocido si ICANON, se efectúa un retorno de carro <i>carriage return</i> .

HP-UX define otra estructura llamada `termios`, en `<termios.h>`, esta estructura aparece como más novedosa que `termio` y cuenta con otras funciones para definir los parámetros (que básicamente son los mismos).

```
#define NCCS 16
struct termios{
    tcflag_t c_iflag;    /**input modes**/
    tcflag_t c_oflag;    /**output modes**/
    tcflag_t c_cflag;    /**control modes**/
    tcflag_t c_lflag;    /**local modes**/
    tcflag_t c_reserved; /**reservado para uso futuro**/
    cc_t c_cc[NCCS];    /**caracteres de control**/
};
```

Bueno, hasta ahora se han mencionado las estructuras que permitirán configurar el puerto, sin embargo aún no se menciona cómo aplicar los cambios. Esto con `termio` se hace a través de llamadas `ioctl()` y con `termios` se hace a través de llamadas especiales de HP (equivalentes).

No existen muchas diferencias entre usar una y otra estructura; HP recomienda el uso de `termios` aunque `termio` es estándar en la mayoría de los sistemas. En el caso de `termio`, el bit `IEXTEN` no puede cambiarse y siempre se considera prendido; con `termios` sí puede prenderse o

apagarse a gusto del usuario. Por otro lado, con `termio` sólo se manejan los primeros 8 caracteres especiales antes mencionados y con `termios` se manejan todos.

3.4.4.-Llamadas al sistema y funciones especiales para el manejo de dispositivos.

En esta sección se da un vistazo a las llamadas al sistema y a las funciones especiales que se usarán para el manejo del CAMAC, por supuesto aquí se incluye la forma en que debe usarse `ioctl()` para aplicar los cambios que se definen en `termio`.

open().- Es la primera llamada al sistema que debe hacerse al manipular archivos, abre el archivo especial y le asigna un *file descriptor* único, el cual será el identificador del archivo, este número será usado para referirse al archivo en subsecuentes llamadas al sistema.

Sintaxis:

```
#include </usr/include/sys/fcntl.h>
```

```
int open(const char *path, int flags, int perms)
```

```
/** regresa -1 si error o un file descriptor si éxito **/
```

path.- Es el nombre del archivo, incluyendo su ruta; es un apuntador a una cadena de caracteres.

flags.- Son constantes simbólicas (o bits) definidas en `/usr/include/sys/fcntl.h`, definen ciertas características de acceso al archivo:

BIT	FUNCION
<code>O_RDONLY</code>	Se abre archivo para sólo lectura.
<code>O_WRONLY</code>	Se abre archivo para sólo escritura.
<code>O_RDWR</code>	Se abre archivo para lectura y escritura.
<code>O_NDELAY</code>	Sólo se usa con archivos especiales, el <code>open()</code> no espera que otro archivo que atiende el mismo puerto sea cerrado <code>close()</code> para regresar, si el puerto está ocupado envía un mensaje de error <code>EBUSY</code> .
<code>O_NONBLOCK</code>	En forma semejante que <code>O_NDELAY</code> , sólo se usa con archivos especiales. Si se prende, tampoco espera que otro

archivo que maneja el mismo puerto cierre `close()` , regresa con éxito.

NOTA.- Sólo uno de los dos bits anteriores puede ser prendido, ya sea `O_NONBLOCK` o `O_NDELAY`. Dependiendo del tipo de conexión (*call-in, call-out* o *direct connect*) tienen efecto estas dos *flags*. En archivos definidos *direct connect*, no importa el estado de las *flags*, el `open()` no espera por un estado particular de las líneas del RS232. Si `O_NDELAY` está apagada, un intento de `open()` mientras se está procesando otro `open()` será bloqueado hasta que el anterior sea cerrado `close()`. Ahora bien, si las 2 están apagadas, subsecuentes peticiones `read()` son bloqueadas hasta que existan datos disponibles o sea recibida alguna señal. Si cualquiera de las 2 es prendida, subsecuentes `read()` son completadas inmediatamente en alguna de las siguientes maneras:

- a) Si existen suficientes datos disponibles para satisfacer la petición `read()`, ésta completa exitosamente con los datos requeridos.
- b) Si no existen suficientes datos disponibles para satisfacer la petición `read()` , ésta completa exitosamente con los datos que estén disponibles.
- c) Si no existe ningún dato disponible, `O_NONBLOCK` regresa con error `EAGAIN` pero `O_NDELAY` regresa exitosamente , con ningún dato.

`O_APPEND`

En archivos ordinarios, asegura que todo lo que se escriba se agregará al final del archivo.

`O_CREAT`

En archivos ordinarios, si el archivo no existe, lo crea con los permisos definidos en el argumento `perms`.

O_TRUNC	En archivos ordinarios, y si el archivo existe, truncar su longitud a 0.
O_EXCL	Se usa en archivos ordinarios para definirlos como semáforo (para evitar que dos procesos accedan al mismo recurso al mismo tiempo, ambos crean un archivo temporal de sólo lectura, en uso exclusivo y hasta que lo liberen puede ser usado por el otro proceso). Falla si ya existe el archivo.
O_CNOTTY	En terminal devices , evita que la terminal llegue a ser la <i>controlling terminal</i> del proceso.
O_SYNC	Sólo para archivos ordinarios o especiales de bloque. El proceso se bloquea hasta que los datos son pasados del cache al dispositivo en un write().

write().- Envía los n bytes que están en el arreglo buf hacia el archivo cuyo *file descriptor* es fd.

Sintaxis:

```
int write(int fd, char *buf, unsigned nbytes)
/** regresa -1 si error o el numero de bytes escritos exitosamente **/
```

fd.- Es un *file descriptor* previamente obtenido en un open().

buf.- Es el arreglo de caracteres que serán enviados, en el caso de archivos de dispositivo, con la configuración default o previamente definida.

nbytes.- Es el número de caracteres a enviar, tomados de la cadena buf.

NOTA.- En nuestro caso, cuando se hace un write() en el archivo especial, los caracteres son puestos en una cola de salida *output queue*. Los caracteres en ella son transmitidos tan pronto como sean enviados los caracteres previamente escritos. Si un proceso produce caracteres de salida más rápidamente de lo que pueden ser enviados, el proceso se suspende cuando la cola de salida excede su límite, se reanuda en el momento en que se han enviado suficientes caracteres

como para desalojar la cola de salida.

read().- Hace una petición de n bytes al archivo cuyo *file descriptor* es fd, la lectura la pone en el arreglo de caracteres buf.

Sintaxis:

```
int read(int fd, char *buf, unsigned nbytes)
```

```
/** regresa -1 si error o el número de bytes leídos o 0 si se encontró EOT **/
```

fd.- Es un *file descriptor* previamente obtenido en un open()

buf.- Es el arreglo de caracteres en donde serán guardados los datos leídos ; la transmisión se hace siguiendo una configuración previa o default en caso de archivos de dispositivo.

nbytes.- Es el número de caracteres requeridos.

NOTA.-En este caso, el desempeño de un read() depende , como ya se explicó, de si se trabaja en modo canónico o no canónico y de las banderas O_NONBLOCK y O_NDELAY del open().

close().- Su función básica es liberar el *file descriptor* del archivo para que pueda ser reutilizado.

Sintaxis:

```
int close(int fd)
```

```
/** regresa 0 en éxito y -1 en error **/
```

fd.- Es un *file descriptor* previamente obtenido en un open().

NOTA.- En este caso, el último proceso que realice un close() sobre el archivo de dispositivo, provocará que cualquier salida no enviada o que haya sido suspendida sea transferida. Este último close **siempre** bloquea hasta que toda la salida ha sido enviada. Cualquier entrada recibida pero no leída es desechada. Ya que la conexión es directa, el *driver* no afecta ninguna línea de control del modem (ver páginas de manual termio y modem para otros casos).

ioctl().- Esta llamada al sistema es muy versátil. Su sintaxis varía de acuerdo al *driver* que se maneje y la función que se desee realizar. Para la configuración de los parámetros referentes al puerto serial se tiene que:

A.Sintaxis:

```
#include </usr/include/sys/termio.h>
```

```
int ioctl(int fd, int cmd, struct termio *arg);
```

```
/** regresa 0 si éxito o -1 si error **/
```

los comandos cmd que usan esta forma sirven para configurar el puerto usando la estructura termio, y son:

COMANDO	DESCRIPCION
TCGETA	Pregunta por la configuración actual del puerto y almacena la información en la estructura arg.
TCSETA	Define una nueva configuración del puerto según el contenido de la estructura arg. El cambio es inmediato, si existen caracteres aún saliendo, el resultado es impredecible.
TCSETAW	Espera que se haya completado la salida antes de actualizar la configuración del puerto según la estructura arg.
TCSETAF	Espera que se complete la salida y limpia la cola de entrada antes de reconfigurar el puerto según parámetros definidos por la estructura arg. Nota: la bandera IEXTEN no puede ser vista ni cambiada por esta llamada, siempre se considera prendida.

B. Sintaxis:

```
int ioctl(int fd, int cmd, int *arg);
```

```
/** regresa 0 si éxito y -1 si error; note que es independiente de termio **/
```

los comandos siguientes son capaces de controlar las entradas y salidas:

COMANDO	DESCRIPCION
TCSBRK	Espera que la salida se complete; si arg=0 envía un BREAK; es decir bits en 0 por lo menos 0.25 seg.
TCXONC	Si arg=0 suspende la salida de datos. Si arg=1 reanuda una salida suspendida. Si arg=2 transmite el caracter STOP

	Si arg=3	transmite el caracter START.
	Si arg es otro valor,	se produce un error.
TCFLSH	Si arg=0	limpia la cola de entrada.
	Si arg=1	limpia la cola de salida.
	Si arg=2	limpia ambos buffers.
	Si arg es otro valor,	se produce un error.

C. Sintaxis:

```
int ioctl(int fd, int cmd, long *arg);
```

```
/** regresa 0 si éxito y -1 si error; note que es independiente de termio **/
```

el siguiente comando observa en el bufer de entrada.

COMANDO	DESCRIPCION
FIONREAD	Regresa en arg el número de caracteres inmediatamente legibles desde el archivo de dispositivo.

Existen otras opciones en la llamada `ioctl` para el manejo del *driver* serial, las mencionadas son las opciones básicas y todo esta muy bien documentado en la página de manual termio.

FUNCIONES ESPECIALES.- Existen algunas funciones íntimamente relacionadas con el manejo del puerto serial. Hasta ahora se ha hablado de la estructura `termio` que es la estándar, sin embargo HP propuso el uso de otra estructura: `termios`; definida en `termios.h` cuya utilización no se hace a través de llamadas `ioctl()` sino a través de llamadas a ciertas funciones especiales.

```
#include <termios.h>
```

<code>cfgetospeed()</code>	Pregunta el <i>output baud rate</i>
<code>cfget ispeed()</code>	Pregunta el <i>input baud rate</i>
<code>cfsetospeed()</code>	Define el <i>output baud rate</i>
<code>cfset ispeed()</code>	Define el <i>input baud rate</i>
<code>tcgetattr()</code>	Preguntar configuración
<code>tcsetattr()</code>	Configurar

Algunas otras funciones útiles , independientes de la estructura que se utilice (termio o terminos) son:

tcsendbreak()	Envía un break
tcdrain()	Espera hasta que se ha completado la transferencia de salida
tcflush()	Limpia alguna de las colas o ambas
tcflow()	Suspende o reanuda las entradas o las salidas
tcgetpgrp()	Pregunta el <i>foreground process group id</i>
tcsetpgrp()	Define el <i>foreground process group id</i> .

Estas funciones tienen sus llamadas ioctl() equivalentes.

NOTA. En caso de que la conexión no fuera directa y atendiera a las líneas de control del modem, podrían usarse llamadas ioctl() descritas en la página de manual modem y termiox.

3.5.-Biblioteca de funciones para el intercambio de datos con el CAMAC.

3.5.1.- Las subrutinas del estándar.

Uno de los documentos del estándar CAMAC, el denominado IEEE Standard Subroutines for CAMAC [3.7], define un conjunto de rutinas recomendables para el intercambio de datos con sistemas CAMAC. De primera intención, se buscaba seguir con el estandar en este trabajo; sin embargo se encontró que tal estandar marcaba la generación de un número grande de subrutinas para efectuar operaciones funcionalmente similares.

El estándar demanda incluir el prefijo “Cf” en el nombre de la subrutina, donde *f* corresponde a una de las siguientes letras, dependiendo del tipo de rutina:

C	para control
D	declaración
F	transferencia de 24 bits
G	llamada a una <i>subaddress</i>
S	transferencia menor a 24 bits
T	test

Por otro lado, el estandar marca varios conjuntos de subrutinas.

- Subrutinas básicas.
- Subrutinas de una simple acción.
- Subrutinas para transferencias de bloques, acciones múltiples y declaraciones inversas.
- Subrutinas dependientes del sistema.
- Transferencias de palabras cortas (menores a 24 bits).

A continuación se mencionan las subrutinas del estandar y el objetivo que persiguen, según el conjunto al que pertenecen.

SUBRUTINAS BASICAS

CDREG Definir la dirección de una entidad CAMAC (*Declare CAMAC Register*).

CFSA Realizar operación CAMAC en un módulo (*Perform single CAMAC action*).

SUBRUTINAS DE UNA SIMPLE ACCION

CCCZ Generar señal Z (*Generate Dataway Initialize*).

CCCC Generar señal C (*Generate Crate Clear*).

CCCI Generar o limpiar I (*Set or Clear dataway Inhibit*).

CTCI Habilitar o deshabilitar la Crate Demand (*Enable or Disable Crate Demand*).

CTCD Consultar el estado de la Crate Demand (*Test Crate Demand Enabled*).

CTGL Verificar si hay una Crate Demand presente (*Test Crate Demand Present*).

CDLAM Declarar LAM (*Declare LAM*).

CCLM Habilitar o deshabilitar LAM (*Enable or disable LAM*).

CCLC Limpiar LAM (*Clear LAM*).

CTLM Test LAM (*Test LAM*).

CCLNK Atender una LAM (*Link LAM to a service procedure*).

SUBRUTINAS PARA TRANSFERENCIAS DE BLOQUES, ACCIONES MÚLTIPLES Y DECLARACIONES INVERSAS

CFGA Ejecución de varias acciones en varios módulos (*General Multiple Action*).

CFMAD Ejecución de una función en varios módulos (*Address Scan Block Transfer*).

CFUBC Transferencia de bloques supervisada por el controlador (*Controller-Synchronized Block Transfer*).

CFUBL	Transferencia de bloques controlada por LAM (<i>LAM-Synchronized Block Transfer</i>).
CFUBR	Transferencia de bloques con Repeat Mode (<i>Repeat Mode Block Transfer</i>).
CGLAM	Devuelve el valor del identificador LAM (<i>Analyze LAM identifier</i>).
CGREG	Devuelve el valor del identificador del registro (<i>Analyze Register Identifier</i>).
SUBROUTINAS DEPENDIENTES DEL SISTEMA	
CCINIT	Ejecuta la señal BZ (<i>Branch Initialize</i>).
CTSTAT	Verifica el estado de la última acción (<i>Test Status of preceding action</i>).
TRANSFERENCIAS DE PALABRAS CORTAS (MENORES A 24 BITS)	
CSSA	Realizar operación CAMAC simple de palabra corta (<i>Perform Single Camac Action</i>).
CSGA	Realizar operaciones múltiples (<i>General Multiple Action</i>).
CSMAD	Ejecución de una función en varios módulos (<i>Address Scan Block Transfer</i>).
CSUBC	Transferencia de bloques supervisada por el controlador (<i>Controller-Synchronized Block Transfer</i>).
CSUBL	Transferencia de bloques controlada por LAM (<i>LAM-Synchronized Block Transfer</i>).
CSUBR	Transferencia de bloques con Repeat Mode (<i>Repeat Mode Block Transfer</i>).

Así pues, se observó que las llamadas “Subrutinas de una simple acción” consistían mayoritariamente en leer, prender o apagar un bit de alguno de los registros internos del controlador.

Por otro lado, se observó que todas las subrutinas para transferencias de bloques no serían necesarias para esta aplicación debido a que, aún cuando el controlador soporta transferencias de bloques, ninguno de los otros módulos que se usaron las soporta.

En cuanto a las subrutinas definidas para la transferencia de palabras cortas es exactamente lo mismo que las transferencias de 24 bits, funcionalmente hablando.

Así pues, se decidió seguir el estandar en tan solo unas cuantas subrutinas y codificar otras tantas que no se encuentran explícitamente en él pero que resultaron ser necesarias.

3.5.2. Biblioteca de funciones.

El archivo especial que se utilizó para las transferencias de datos con el CAMAC fue `/dev/plt_rs232_a` (el cual estaba inicialmente pensado para transferencias con plotters seriales, sin embargo, satisfizo plenamente los requerimientos de este trabajo).

La única preparación adicional que hubo que hacerle al archivo fue agregarle los permisos de escritura ejecutando

```
chmod go+w plt_rs232_a
```

con lo cual, cualquier usuario puede leer y/o escribir en ese archivo especial. Esto sólo fue posible debido a que la aplicación en cuestión no tiene requerimientos críticos en cuanto a seguridad; sin embargo, es importante señalar que dar los permisos de escritura a todos los usuarios no es lo más conveniente y se recomienda que para aplicaciones futuras se cree un grupo de trabajo (el encargado de interactuar con el CAMAC por ejemplo) y tan solo a él se le den permisos de lectura y escritura sobre el archivo. Esta tarea deberá desempeñarla un superusuario o root.

Se decidió codificar en lenguaje C un conjunto de funciones que facilitaran el intercambio de datos con el CAMAC y su interpretación. El código fuente de tales funciones quedó en el archivo `cmc_lib.c`. Para acceder estas funciones es necesario ligar el o los objetos de la aplicación con el objeto `cmc_lib.o`. Por ejemplo, podría ejecutarse:

```
cc -o nomexe nomarch.o cmc_lib.o
```

donde *nomexe* es el nombre del archivo ejecutable que correrá la aplicación, *nomarch.o* es el nombre del archivo objeto de la aplicación y `cmc_lib.o` es el archivo objeto de la biblioteca de funciones.

A continuación se describe la “biblioteca de funciones” (se le ha llamado así informalmente, ya que para ser una biblioteca de funciones formalmente en términos de UNIX debiera definírsele como tal). Se han agrupado las funciones según su desempeño en: miscelaneo, configuración del puerto, y adquisición de datos e información.

MISCELANEO

void syserr(char *msg)

En caso de error, envía a la salida estandar de errores un mensaje definido por msg, el código del error y la descripción del sistema para ese error, finalmente termina el proceso.

void pausa(unsigned long secs, long micro)

Hace una pausa por el número de segundos y microsegundos definidos por secs y micro, está limitada por la granularidad del reloj de la computadora que en el caso en cuestión es de 10,000 μ s; es decir el valor mínimo para micro es de 10 000 (una centésima de segundo).

void timestart()

Toma el tiempo del sistema.

void timestop()

Cuenta el tiempo transcurrido desde la llamada previa timestart() en segundos, envía el valor a la salida estándar de errores, la resolución de esta función es de un segundo.

CONFIGURACION DEL PUERTO

void config()

Configura el puerto RS232 para transferencias canónicas según las necesidades CAMAC.

void config_NOCANON()

Configura el puerto RS232 para transferencias no canónicas según las necesidades CAMAC con MIN=20 y TIME=1.

void restore()

Restaura la configuración original del puerto RS232, la que se mantenía antes de alguna llamada previa config() o config_NOCANON().

ADQUISICION DE DATOS E INFORMACION

int CDREG(char c, char n, char a, char *dir)

Construye en dir una dirección en formato CAMAC donde c indica la CAMAC address, n la dirección del módulo y a la subaddress. Devuelve el número de caracteres en la cadena dir.

int CDINT24(unsigned char dato_a, unsigned char dato_b, unsigned char dato_c, char *data)

Construye en data una cadena de datos en formato CAMAC de 24 bits, a partir de los números dato_a, dato_b y dato_c correspondientes a DH, DM y DL del formato CAMAC.

Devuelve el número de caracteres en la cadena.

int CDINT16(unsigned char dato_a, unsigned char dato_b, char *data)

Construye en data una cadena de datos en formato CAMAC de 16 bits, a partir de los números dato_a y dato_b. Devuelve el número de caracteres en la cadena.

int CDINT8(unsigned char dato_a, char *data)

Construye en data una cadena de datos en formato CAMAC de 8 bits, a partir del número dato_a. Devuelve el número de caracteres en la cadena.

int CFSA(char f, char *dir, char *data)

Construye comando CAMAC simple según la dirección dir, el código de función f y los datos dados por data. Envía el comando, lee la respuesta CAMAC y la almacena en data. Regresa el número de caracteres leídos como respuesta CAMAC o -1 si se ingresó un código de función no válido.

int CTCI(char *resp, unsigned char *i_data)

De una respuesta CAMAC resp, selecciona sólo los datos y los almacena en el arreglo de enteros i_data. Regresa el número de enteros en el arreglo (pueden ser hasta cuatro enteros correspondientes a DH, DM, DL y STATUS).

int CHECK_BIT(unsigned char *i_data, int byte, char bit)

Verifica si está prendido el bit (de 1 a 32) en el arreglo de enteros i_data que sólo puede contener 4 enteros o bytes; el argumento byte informa el número de bytes que tiene i_data y el bit 1 será el de más a la derecha y el de más a la izquierda será el bit 32 (si existen los 4 bytes) o el bit 24 (si sólo son 3 bytes) o el bit 16 (si sólo son 2 bytes) o el bit 8 (si es únicamente un byte). Regresa 1 si está prendido y 0 si apagado. Por ejemplo si se tuviera un arreglo de enteros i_data={0,0,1,0}, ejecutar CHECK_BIT(i_data,4,9) regresa 1.

void espera_NULL()

Espera la llegada de un caracter. Debe usarse sólo en el caso de tener la certeza de que CAMAC enviará un caracter. Debe usarse en modo canónico con EOF=NULL o bien en modo no canónico con MIN>0.

void ver_resp(char *data)

Muestra en pantalla la cadena data que se supone ser una respuesta CAMAC, también

muestra el código ASCII de cada uno de los caracteres que la componen.

```
void ver_bytes(int num, unsigned char *i_data)
```

Muestra en pantalla tantos enteros del arreglo `i_data` como se especificuen en `num`.

NOTA: En el apéndice 1 se incluye el código de esta biblioteca de funciones.

3.5.3. Un primer programa: Control de Temperatura

Con la finalidad de probar la biblioteca de funciones y el desempeño de las transferencias no canónicas entre el CAMAC y la computadora, se decidió comenzar echando a andar un programa para control de temperatura antes de trabajar con el problema de interferometría.

Se busca mantener una barra de latón a una temperatura de control dentro de un rango razonable (± 2 décimas de grado, por ejemplo). La barra es enfriada por medio de hielo (ver fig 3.5) y puede ser calentada a través de un calefactor. La participación del CAMAC en este sistema es de “encender” el calefactor si la barra se enfría más allá del rango o bien de “apagar” el calefactor si la barra se calienta más allá del rango permisible.

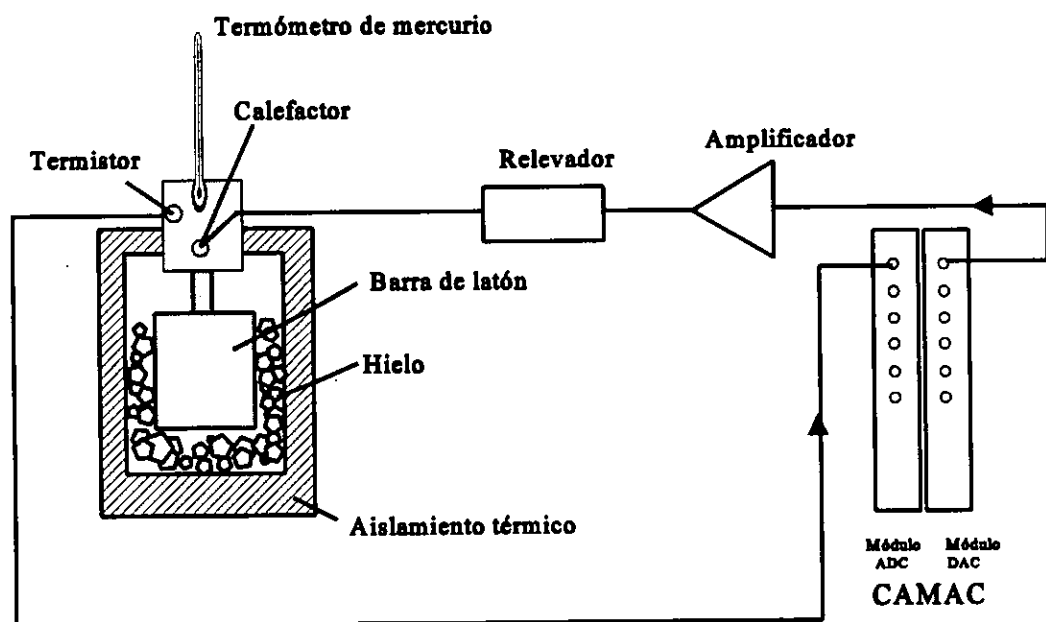


Fig. 3.5. Sistema de control de temperatura con diagrama de bloques de la electrónica.

Un termistor efectúa la operación: temperatura de control - temperatura real, y envía un voltaje proporcional a esa diferencia. Ese voltaje es recibido por el módulo CAMAC ADC 3527 *Analog Input*. El objetivo del programa de control es verificar con una periodicidad adecuada si el voltaje recibido es mayor que cero (ha bajado la temperatura); si es así se debe generar otro voltaje a través del módulo DAC 3112_M1A el cual encenderá el calefactor. De lo contrario, si el voltaje de entrada no es mayor que cero (no ha bajado la temperatura), deberá generarse un voltaje por el módulo DAC que apagará el calefactor. De esa manera se logra mantener la temperatura dentro del rango deseado.

Es muy importante mencionar que el cableado con los módulos CAMAC debe tenderse correctamente, de lo contrario puede ocurrir que en lugar de mantenerse la temperatura dentro de un nivel constante, se aleje de él.

Este programa fue llamado volt08.exe , utiliza transferencias no canónicas. En el apéndice 2 se incluye su código fuente como un ejemplo de ellas (ya que el problema de interferometría se resolvió utilizando transferencias canónicas pues resultaron ser más rápidas). Asimismo, se pretende que sirva como un ejemplo más de la utilización de la biblioteca de funciones.

CAPITULO 4.- DETERMINACION INTERFEROMETRICA DE LA AMPLITUD DE UN OSCILADOR ARMONICO.

4.1.- Descripción del problema.

En febrero de 1997, fue descrito un instrumento basado en un interferómetro tipo Michelson, capaz de medir en forma “simple, precisa y sin contacto” la amplitud y la frecuencia de un oscilador [4.1]. En dicho trabajo, se consideró que el oscilador puede estar pegado al espejo móvil del interferómetro, o bien puede tener una superficie altamente reflejante que haga las veces de espejo. Un fotodetector registra la señal interferométrica la cual es procesada por un circuito RC (recibe un tratamiento analógico) y finalmente llega a una tarjeta contadora en una PC-AT 386. Así es como llegan a generar una tabla de amplitudes y/o una tabla de frecuencias del oscilador.

El presente trabajo está basado en esa idea pero con ciertas variaciones importantes:

- El equipo que se ha utilizado en este trabajo es mucho más poderoso y hace posible, como se verá, obtener más información acerca del movimiento oscilatorio.
- Se detectó un error y se realizó la corrección a la fórmula para determinar la amplitud del oscilador.
- El tratamiento es absolutamente digital; una vez que la señal sale del amplificador, no se hace ningún tratamiento analógico.

El estudio de oscilaciones tiene importancia en la Física y en la Ingeniería pues se utiliza para determinar ciertas características físicas de los materiales (las propiedades elásticas y la fricción interna son un ejemplo tratado en la referencia [4.1]). La idea de realizarlo haciendo uso de técnicas interferométricas tiene la ventaja de que el instrumento o la técnica de medición no influye significativamente sobre el fenómeno estudiado ya que no existe ningún contacto mecánico. Pueden medirse oscilaciones inducidas o libres en forma indistinta.

Para hacer las mediciones, es necesario que el espejo móvil del interferómetro se encuentre

pegado al cuerpo oscilante, o bien que éste cuente con una superficie altamente reflejante (ver fig. 4.1).

En este caso, las distancias d_2 y d_1 corresponden a las distancias entre D y E2 y entre D y E1 respectivamente. Cuando el sistema se encuentra en una posición inicial de reposo, puede definirse $\Delta d_0 = d_2 - d_1$ como la diferencia inicial de distancias (o camino óptico). La diferencia inicial de trayectorias Δx_0 inicial es pues, $\Delta x_0 = 2 \Delta d_0$.

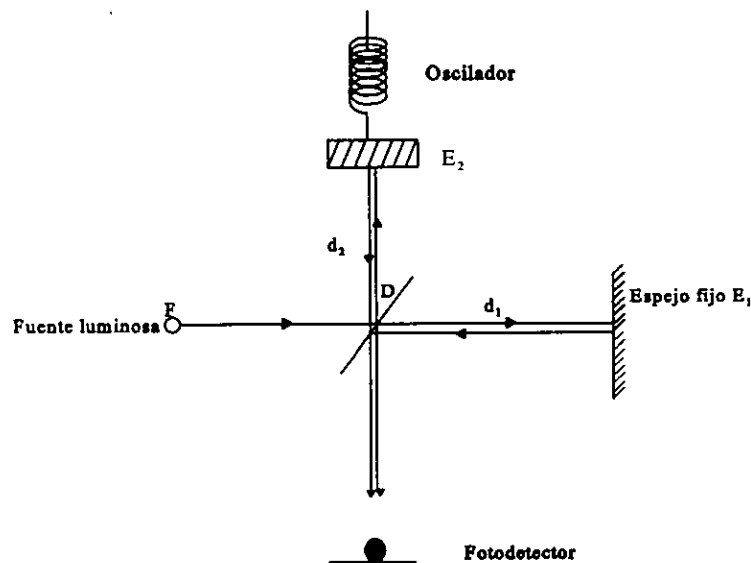


Fig. 4.1. Disposición interferométrica para el estudio de oscilaciones.

Dado que el oscilador sigue un movimiento armónico (hacia arriba y hacia abajo), la diferencia de distancias Δd una vez que comienza el movimiento está dada por

$$\Delta d = \Delta d_0 + A \text{sen} [(2\pi f t) + \phi_0] , \quad (4.1)$$

donde Δd_0 es la diferencia inicial de distancias, A es la amplitud de la oscilación, f es la frecuencia de la oscilación y ϕ_0 es una posible fase constante de la oscilación.

Note que Δd puede interpretarse como el desplazamiento del oscilador respecto a la posición de reposo o equilibrio Δd_0 (fig. 4.2).

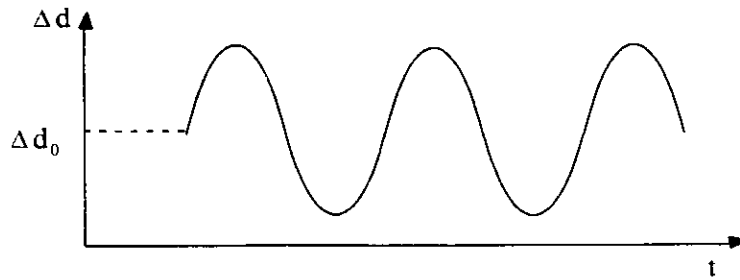


Fig. 4.2. Gráfica del desplazamiento del oscilador en el tiempo.

La diferencia de trayectorias Δx de los rayos que interfieren, una vez que ha comenzado el movimiento está dada por

$$\Delta x = 2 \Delta d ;$$

sustituyendo Δd según la ecuación (4.1) se tiene que

$$\Delta x = 2 (\Delta d_0 + A \text{sen} [(2\pi f t) + \phi_0]) .$$

Dicho ésto, y según lo que se expuso en el capítulo 1, las variaciones en la intensidad que son captadas por el fotodetector están dadas por la ecuación (1.7); o sea

$$I = 4 I_0 \cos^2 \left[\frac{2\pi (\Delta d_0 + A \text{sen} [(2\pi f t) + \phi_0])}{\lambda} \right] . \quad (4.2)$$

Se presentarán máximos de intensidad toda vez que

$$\Delta x = m\lambda \quad \text{con } m=0, \pm 1, \pm 2, \dots ;$$

es decir, cuando el desplazamiento Δd sea

$$\Delta d = m\lambda/2 \quad \text{con } m=0, \pm 1, \pm 2, \dots$$

Así las cosas, la distancia entre cualesquiera dos máximos de intensidad es de $\lambda/2$ (una vez que el oscilador se encuentra en una posición de máxima intensidad, requiere un desplazamiento de $\lambda/2$ para volver a encontrar otro máximo).

En la figura 4.3 se muestra la correspondencia que habría entre los máximos de intensidad, con el movimiento oscilatorio visto en términos de longitudes de onda. Entiéndase k (esta k es diferente a la k en el capítulo 1) como el número de máximos de intensidad que pueden observarse cuando el oscilador recorre una distancia $2A$; o lo que es lo mismo, el número de máximos que se presentan en la mitad de un período de oscilación $T/2$ (en la figura puede verse como si se

desplazara de la posición $-A$ a la posición A).

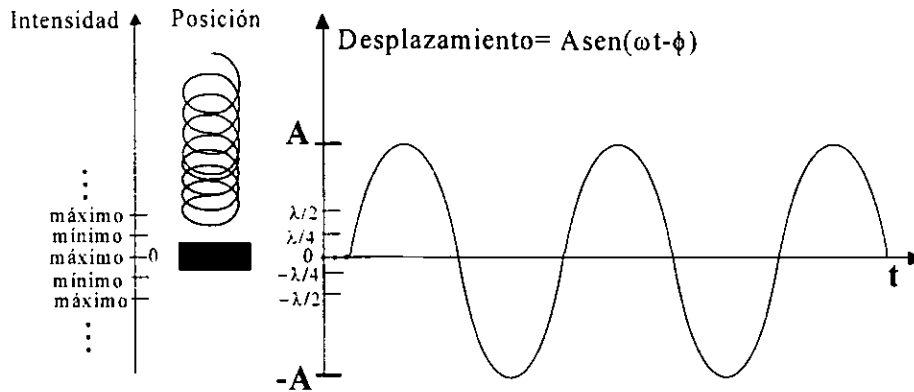


Fig. 4.3. Gráfica del desplazamiento del oscilador y su correspondencia con la intensidad I de la señal interferométrica.

En este caso, la distancia $2A$ puede estimarse como

$$2A = \lambda/2 (k-1) + c ,$$

con la constante c en el intervalo $0 \leq c < \lambda$. Es decir:

$$2A \geq \lambda/2 (k-1) ,$$

$$2A < \lambda/2 (k-1) + \lambda ;$$

dicho de otro modo:

$$\lambda/2 (k-1) \leq 2A < \lambda/2 (k-1) + \lambda ,$$

$$k \lambda/2 - \lambda/2 \leq 2A < k \lambda/2 - \lambda/2 + \lambda ,$$

$$k \lambda/2 - \lambda/2 \leq 2A < k \lambda/2 + \lambda/2 ;$$

esto puede expresarse como

$$2A = k \lambda/2 \pm \lambda/2 ;$$

entonces la amplitud A es

$$A = k \lambda/4 \pm \lambda/4 . \tag{4.3}$$

De manera que puede estimarse A como $A = k \lambda/4$ con un error de $\pm \lambda/4$; este resultado constituye una corrección a lo que se postula en la referencia [4.1] en donde se dice que

$$A \geq k \lambda/4 \pm \lambda/2 .$$

Note que este tratamiento es válido para cualquier valor de Δd_0 .

Lo que sigue es encontrar una manera de determinar el valor de k . Obsérvese el patrón de intensidad que se muestra en la figura 4.4, en la que por claridad se consideró que la amplitud no es un múltiplo de $\lambda/4$. Los puntos P1 y P2 mostrados corresponden a los momentos en que el oscilador se encuentra en su máximo desplazamiento hacia arriba o hacia abajo, así, k es el número de máximos o picos que pueden contarse entre P1 y P2, el tiempo que transcurre entre estos puntos es $T/2$ donde T es el periodo del movimiento del oscilador medible también experimentalmente.

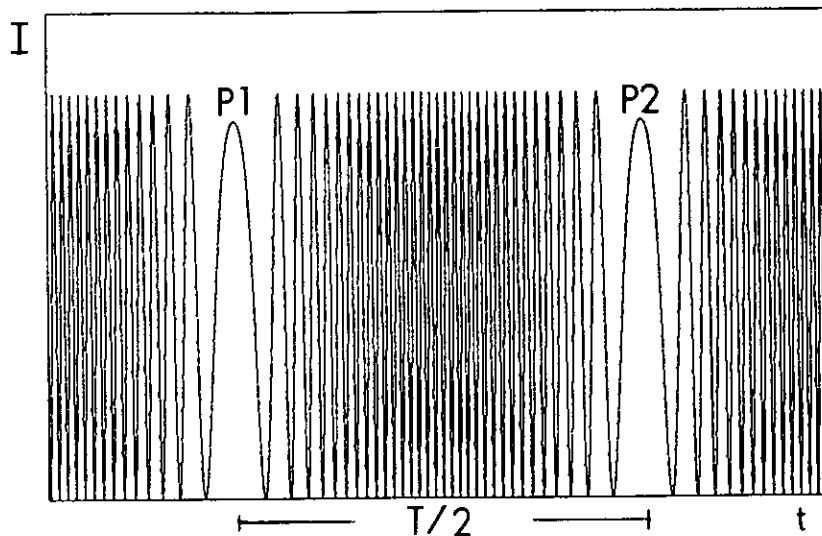


Fig. 4.4. Patrón teórico de intensidad

Así pues, el equipo debe ser capaz de detectar los puntos P1 y P2 y contar los máximos de intensidad, al mismo tiempo debe ser capaz de contar el tiempo transcurrido. De esa forma se podrá calcular el valor de A y obtener información acerca del periodo T y/o de la frecuencia f del oscilador.

La técnica descrita en la referencia [4.1] hace uso de métodos analógicos para detectar los puntos P1 y P2, de modo que el proceso de cómputo que utiliza, únicamente arroja una tabla de

valores de A obtenidos periódicamente. En contraste, para este trabajo, absolutamente todo el análisis será digitalizado, haciendo uso del sistema de adquisición de datos y de la estación de trabajo UNIX. Los resultados, como se verá, ofrecen mucha más información acerca del oscilador que tan solo una tabla de amplitudes.

La estrategia que se siguió para atacar el problema fue la siguiente:

- Detectar cada uno de los máximos de intensidad. Como se hace evidente en la figura 4.4, la razón de máximos en el tiempo se altera periódicamente, es decir, existen regiones "más densas" (que corresponden a mínimos de tiempo entre dos máximos de intensidad) y regiones "menos densas" (que corresponden a máximos de tiempo entre dos máximos de intensidad).
- Contar el tiempo entre máximos de intensidad. Precisamente los puntos P1 y P2 se encuentran en consecutivas regiones menos densas, así, detectando dos máximos consecutivos de tiempo se estará detectando la posición en el tiempo de los puntos P1 y P2. Esto tiene la ventaja adicional de proporcionar información acerca de la frecuencia del oscilador y de su periodo T.
- Determinar el número de máximos de intensidad entre dos regiones menos densas (o sea determinar k). Esto es, contar el número de máximos de intensidad entre puntos P1 y P2, y obtener así información acerca de la amplitud A del oscilador.

Estos pasos se siguieron en cada muestra, durante todo el tiempo de muestreo.

4.2.- Configuración del equipo.

DISPOSICION DEL EQUIPO

Dado que el objeto de este estudio es la técnica a utilizar para la medición de la amplitud de un movimiento armónico arbitrario, y no resultaba de interés analizar al movimiento de un objeto en particular; se dispuso el equipo de la manera que se muestra en la fig. 4.5.

Obsérvese que se obtuvo un movimiento armónico dando un pequeño impulso a la esquina de cubo: como ésta se encuentra apoyada de la forma en que se muestra en la figura, oscila por un periodo de tiempo (girando hacia arriba y hacia abajo). Gracias a las características ópticas de

la esquina de cubo (el rayo no cambia la dirección de su trayectoria ante movimientos sobre el eje de la esquina de cubo), este movimiento es equivalente al mostrado en la figura 4.1. y matemáticamente se describe igual (pues sólo se toma en cuenta la componente vertical del movimiento). Así pues, la esquina de cubo hizo las veces de oscilador.

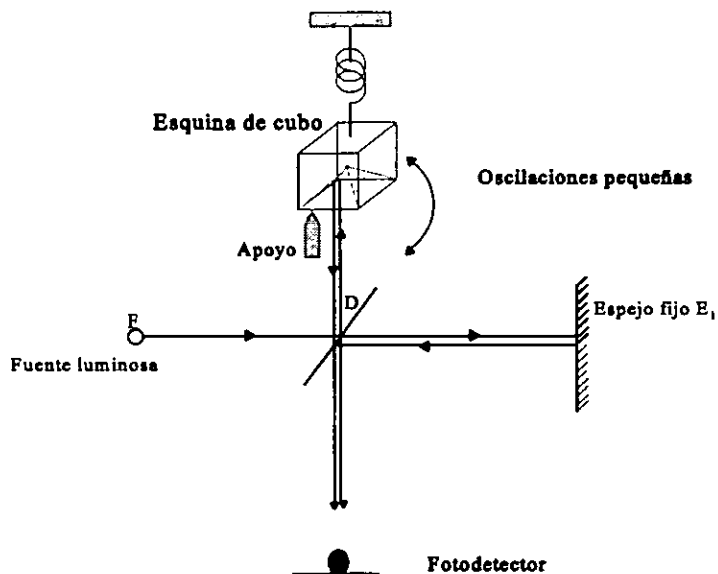


Fig. 4.5. Disposición del oscilador en el interferómetro.

En el capítulo 1 se dijo que la señal del fotodetector resulta finalmente en una señal eléctrica continua amplificada, correspondiente a la intensidad I según la ecuación (4.2); (de cualquier forma se seguirá hablando de ella como señal interferométrica). La adquisición y proceso de datos involucra además la utilización del CAMAC y de la estación de trabajo.

Como resultaba predecible, dada la versatilidad del CAMAC, para resolver el mismo problema existían muchas configuraciones posibles. La señal interferométrica podría ser señal de entrada del módulo secuenciador de listas o de cualquier módulo contador. Lo más adecuado, y que resultó en la configuración más simple, fue la utilización del módulo 3664-L1A *Time Interval Counter*; el cual, como se describió brevemente en el capítulo 2, por medio de cada uno de sus seis canales se encarga de contar el tiempo que transcurre entre una señal de inicio (que es , o bien una señal externa o bien un comando CAMAC) y la llegada de un número predefinido de pulsos en

ese canal (fig. 4.6).

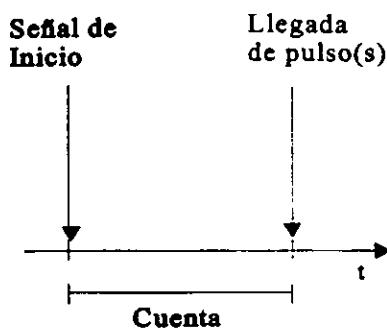


Fig. 4.6. Representación de la cuenta de pulsos dada por el *Interval Counter*.

El reloj interno del interval counter es configurable, puede dar TICKS con frecuencias desde 1Hz (un TICK por segundo) hasta 10 MHz (10 millones de TICKS por segundo), de modo que las cuentas de tiempo que arroja pueden estar en unidades desde segundos hasta décimas de microsegundo.

En la documentación se define que cuenta pulsos, es decir, en cada canal espera una señal del tipo definido por el estandar TTL, y en términos generales cuenta los pulsos de la manera que se describe a continuación (fig. 4.7).

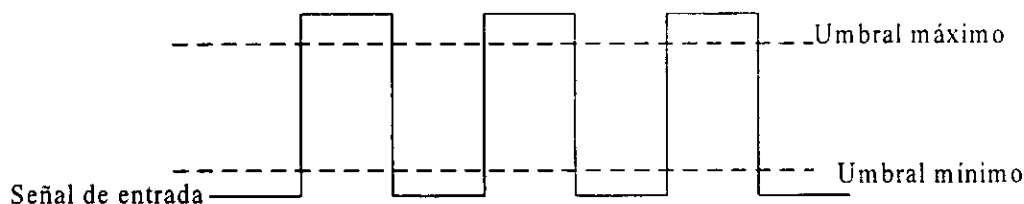


Fig. 4.7. Representación esquemática de una señal y los umbrales del estandar TTL.

Considera un pulso sólo si la señal viene del umbral mínimo, llega o sobrepasa el umbral máximo y luego decrece hasta el umbral mínimo (los valores de umbral máximo y umbral mínimo están definidos en el estandar TTL y su propósito es el de evitar que en señales ruidosas se hagan cuentas espurias).

Aun cuando la señal interferométrica no es cuadrada como la que se describe previamente.

sino senoidal, las mismas características antes mencionadas se aplican a ella (fig. 4.8). El *interval counter* cuenta como un pulso cada vez que la señal senoidal pasa del umbral mínimo al máximo y luego regresa al umbral mínimo, lo que más o menos equivale a decir que cuenta cada vez que la señal cambia de pendiente positiva a pendiente negativa; o sea, **cuenta máximos** .

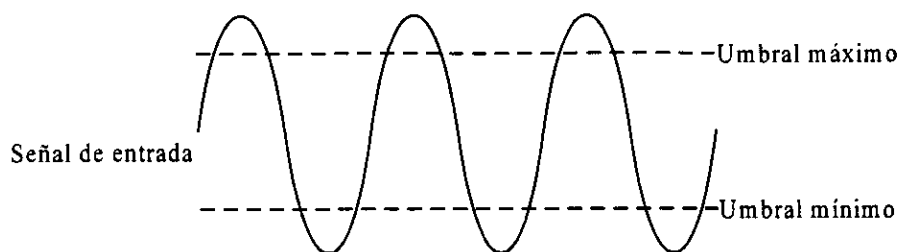


Fig. 4.8. Representación de la señal interferométrica y los umbrales del estándar TTL.

Los umbrales fueron definidos para que ruido pequeño no afecte la cuenta de pulsos; así, de una señal como la que se presenta en la fig. 4.9, el *Interval Counter* contará 4 pulsos.

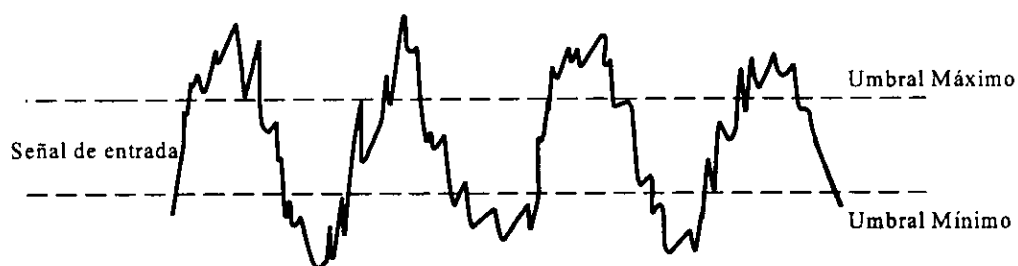


Fig. 4.9. Representación de una señal interferométrica ruidosa y los umbrales del estándar TTL.

Una consideración más se hace necesaria, el *interval counter* acepta frecuencias de entrada máximas de 5MHz, lo que quiere decir que soporta un tiempo mínimo entre dos pulsos mayor o igual a dos décimas de microsegundo.

Así las cosas, el *interval counter* detecta los máximos de intensidad (que son aceptados como pulsos). Para que cuente el tiempo entre dos máximos de intensidad es necesario hacer un arreglo más: sincronizar la señal de inicio del *interval counter* con el último pulso de entrada, sin

olvidar guardar el dato de tiempo transcurrido.

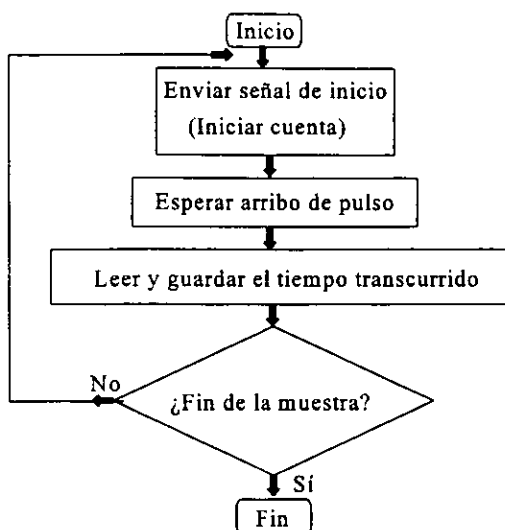


Fig. 4.10. Diagrama de flujo del proceso de adquisición de datos.

En la figura 4.10 se presenta el diagrama de flujo del proceso de adquisición de datos. El *interval counter* debe recibir una señal de inicio para empezar a contar el tiempo, en cuanto arriba un pulso el *interval* deja de contar y esa cuenta está lista para ser leída y guardada en algún medio de almacenamiento (no reinicia la cuenta hasta recibir otra señal de inicio), acto seguido, y lo antes posible, debe enviarse otra vez la señal de inicio de modo que la cuenta represente el tiempo entre pulsos con un error mínimo.

Para lograr aprovechar al máximo el equipo y lograr atender frecuencias de pulsos de entrada mayores, fue necesaria la utilización de un módulo más: el 3982-Z3B *List Sequencing Crate Controller* o Secuenciador de Listas.

El secuenciador de listas dota el medio de almacenamiento (aunque tiene una capacidad de memoria muy limitada) y el tiempo de respuesta adecuado; en su papel de controlador auxiliar es capaz de mandar ejecutar comandos; la señal de inicio para el *interval counter* por ejemplo.

Pudo haberse utilizado directamente la computadora pero por las características de la interfaz RS232 hubiera sido imposible analizar frecuencias mayores a 45.3 Hz, ya que estaría involucrada la transferencia de al menos 53 caracteres a 19200 bauds antes de poder hacer dos lecturas consecutivas ($19200 \text{ bits por segundo} / (53 \text{ caracteres} * 8 \text{ bits por caracter}) = 45.283 \text{ ciclos}$

por segundo).

Así pues, al secuenciador de listas deben cargársele los comandos necesarios para que envíe la señal de inicio, lea la cuenta del *interval counter* en cuanto arribe un pulso y la almacene en su memoria, tantas veces como se requiera o tantas veces como soporte la capacidad de memoria (sólo puede almacenar 4096 cuentas).

Con todo esto se logran los dos primeros puntos en la estrategia: detectar los máximos de intensidad y contar el tiempo entre ellos.

Para lograr el tercer punto de la estrategia, los datos almacenados en la memoria del secuenciador de listas deben ser procesados; encontrar primero las cuentas máximas (máximos de tiempo correspondientes a “regiones menos densas”) y con esos datos obtener información acerca de la amplitud *A* y del periodo *T* del oscilador. En cuanto se obtiene una muestra adecuada, los datos se envían a la computadora para su proceso.

En resumen, la disposición del equipo y la función que realiza cada elemento puede describirse por el diagrama en la fig. 4.11.

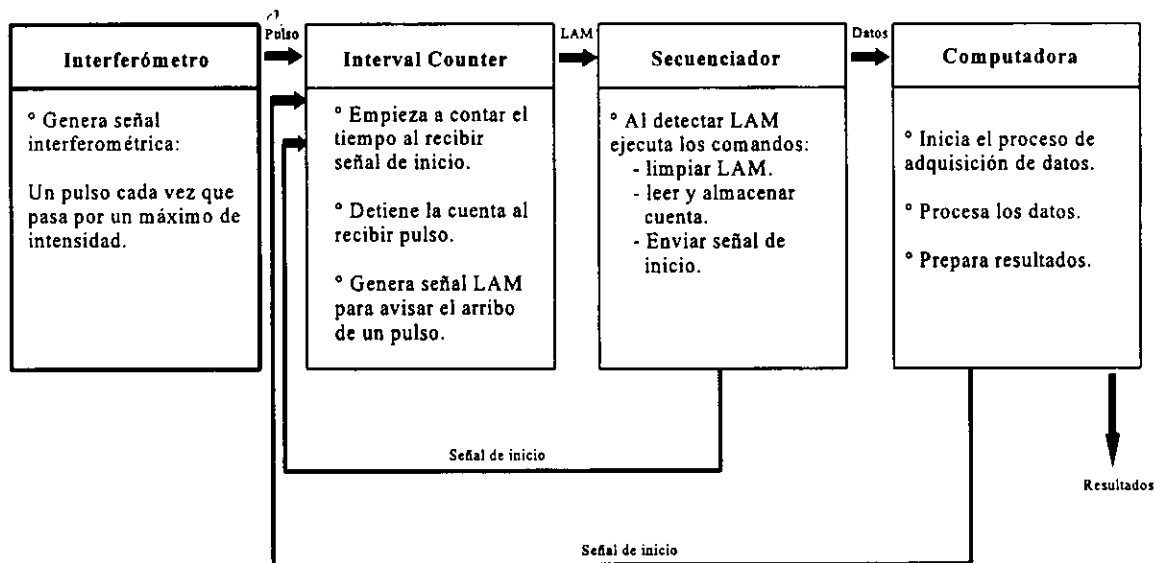


Fig. 4.11. Diagrama de bloques de la función que realiza cada elemento del equipo.

Con esta configuración del equipo, el tamaño de la muestra está limitado a la capacidad de almacenamiento del secuenciador de listas (4096 cuentas). El límite máximo de frecuencias

de entrada es de $196.078 \text{ KHz} \pm 3.92 \text{ KHz}$ para que no se pierda ningún pulso y la cuenta máxima de tiempo es de 2^{24} TICKS del reloj del *interval counter* (ello se ha medido experimentalmente). La precisión de las cuentas puede ser de décimas de microsegundo, sin embargo deben ser ajustadas sumándoles el tiempo perdido (t_p) que transcurre entre la recepción del pulso y la señal de inicio (fig. 4.12). Ese tiempo t_p corresponde al tiempo que requiere el CAMAC para ejecutar los comandos en el secuenciador de listas.

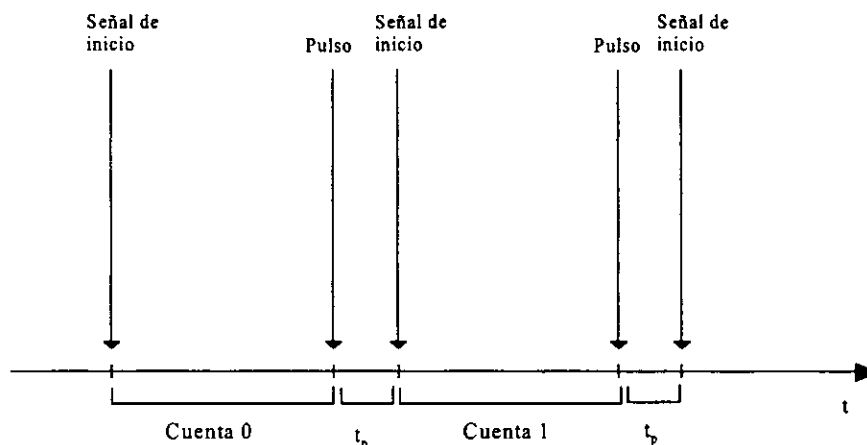


Fig. 4.12. Representación de la cuenta de tiempo entre pulsos.

LIMITACIONES DEL EQUIPO

El *interval counter* se configuró para trabajar con su reloj a 10 MHz, esto significó frecuencias de entrada máximas de 5 MHz (según define el manual) y mínimas de $10 \text{ MHz}/2^{24}$ es decir 0.596 Hz. Sin embargo, el tiempo t_p que ocupa el secuenciador de listas en ejecutar sus comandos influye también en las limitaciones del equipo. El tiempo t_p fue medido experimentalmente y resultó ser $t_p = 5.1 \pm 0.1 \mu\text{s}$ para esta aplicación específica, por lo tanto, las limitaciones del equipo quedan:

frecuencias máximas de entrada: $196.078 \text{ KHz} \pm 3.92 \text{ KHz}$.

frecuencias mínimas de entrada: 0.596 Hz.

referentes a frecuencias de la señal interferométrica y con un reloj del *interval counter* a 10 MHz.

ADQUISICION Y PROCESO DE DATOS

Para efectos prácticos, el problema se dividió en dos partes: adquisición de datos (que lleva

a cabo los dos primeros puntos de la estrategia) y proceso de datos (que lleva a cabo el tercer punto de la estrategia). Dos programas diferentes se encargaron de cada parte; esta decisión se tomó para tener la opción de inspeccionar la muestra antes de mandarla a proceso.

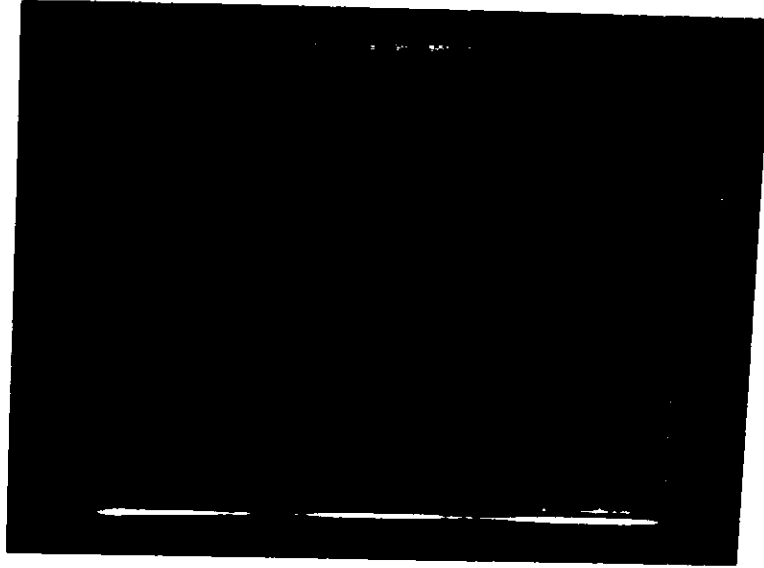


Fig. 4.13. Fotografía de la señal de frecuencia modulada por voltaje vista en un osciloscopio.

Los programas para la adquisición y para el proceso de datos, se probaron con una señal simulada que se logró haciendo uso de dos generadores de señales: el Generador de Funciones IMAC GF-D01 y el Hewlett Packard 8111A *Pulse/Function Generator*. Fue una señal de frecuencia modulada por voltaje, equivalente a la que el interferómetro arroja (fig. 4.13). En el apéndice 6 se muestran los resultados obtenidos de la simulación.

Esa fue señal de entrada tanto para el *interval counter* como para un osciloscopio Tektronix 7704A Amplifier 7A15A lo que permitió hacer una inspección visual.

Finalmente, se hicieron algunas tomas con el sistema interferométrico y se obtuvo información real acerca del movimiento del oscilador armónico, también se hizo una inspección visual a través del osciloscopio.

4.3.- Adquisición de datos

El programa de adquisición de datos `adqdata.c` es un programa en lenguaje C para UNIX que utiliza la “biblioteca de funciones” descrita en el capítulo anterior, y que se encarga de:

1. Configurar el puerto RS232 adecuadamente

→ Se usa modo canónico para aprovechar la característica CAMAC de enviar un caracter ETX al final de la respuesta en cada comando.

→ Se prenden los bits para control de flujo XON / XOFF.

→ Se usan 2 bits de stop, 8 bits por caracter, paridad impar y 19200 *baudrate*.

→ El arreglo de caracteres especiales queda de la siguiente manera:

```
c_cc[0]=127  c_cc[2]=8    c_cc[4]=0    c_cc[6]=255
c_cc[1]=28   c_cc[3]=21   c_cc[5]=3    c_cc[7]=255
```

2. Preparar al controlador para que supervise adecuadamente la adquisición de datos.

→ Se verifica que el controlador esté ON-LINE.

→ Se genera una señal Z y se inicializa al controlador para transferencias simples de 24 bits y STATUS *disable*.

→ Se define que el controlador envíe un BREAK en caso de recibir una LAM del secuenciador de listas, ésto para avisar a la computadora que la muestra ha llegado a su fin.

3. Preparar al *interval counter* para la correcta adquisición de datos.

→ Se define frecuencia de reloj a 10 Mhz.

→ Se define el número de pulsos a considerar; sólo uno en esta aplicación.

→ Se habilita el canal 0 para enviar una LAM en cuanto reciba un pulso.

4. Preparar al secuenciador de listas para la correcta adquisición de datos

El módulo debe estar configurado externamente como controlador auxiliar, el reloj externo debe estar en modo *disabled*, debe conectarse la LAM TRIGGER A-5, así iniciará la ejecución de la lista al llegar la LAM del *interval*. El *time-out* debe estar *disabled* y la *retransmit option disabled*. Por *software*:

→ Se define la velocidad de ejecución de cada comando a la máxima posible (1MHz).

→ Se deshabilita la repetición automática de la lista (*sequence recycle*).

→ Se habilita la transferencia de bloques (*block transfer mode*), con ello se asegura

que la ejecución de cada comando será aproximadamente cada 1.1 microsegundos.

→ Se carga la lista de comandos a ejecutar:

- Leer cuenta del canal 0 del *interval counter*.
- Limpiar LAM (esto es para que el secuenciador sea sensible a la siguiente LAM).
- Enviar señal de inicio al *interval counter*.

→ Se borran todos los datos previamente almacenados y se prepara para recibir los nuevos.

→ Se define que el secuenciador envíe una LAM en caso de que se haya sobrepasado la capacidad de almacenamiento; esto indicará al controlador que la muestra ha llegado a su fin.

→ Se habilita la ejecución de la lista (a partir de este momento, se iniciará una secuencia en cuanto se reciba la LAM del *interval counter* debido a la recepción de un pulso).

5. Enviar señal de inicio al *interval counter*.
6. Esperar por el BREAK que envía el controlador en cuanto la muestra llega a su fin.
7. Hacer la transferencia de los datos del CAMAC a la computadora.
8. Guardar en un archivo, definido por el usuario, lo siguiente (en columnas):
 - **Número de cuenta** (consecutivo de 1 a 4095, el dato 0 se desprecia pues es espurio).
 - **Cuenta** arrojada por el *interval counter*, ajustada al tiempo $t_p = 5.1 \mu s$ (esta cuenta está expresada en TICKS del reloj del *interval counter*; 1 TICK=0.1 μs). En lo subsecuente se usarán indistintamente los términos cuenta y cuenta ajustada para referirse a esta misma magnitud.
 - **Acumulado** de las cuentas ajustadas.
 - **Inverso** de las cuentas ajustadas; es decir, $1/(\text{cuenta ajustada})$.
9. Mandar a la salida estándar de errores el tiempo que toma adquirir los datos desde que se da la señal de inicio al *interval counter*, expresado en segundos.
10. Presentar en pantalla las siguientes gráficas:
 - Cuenta contra Número de cuenta.

- Cuenta contra Cuenta acumulada.
- Inverso de cuenta contra Número de cuenta.
- Inverso de cuenta contra Cuenta acumulada.
- Número de cuenta contra Cuenta acumulada.

en forma adicional, se genera un archivo llamado **adqdata.gnu** que contiene las directrices necesarias para volver a generar las gráficas en forma externa haciendo uso del graficador **gnuplot**.

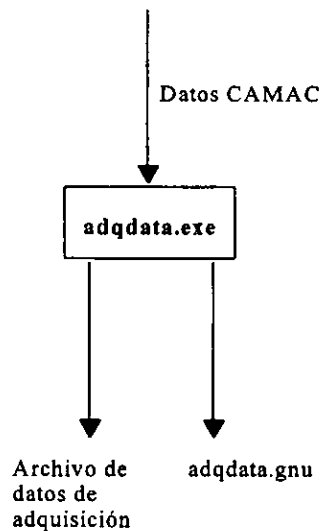


Fig. 4.14. Diagrama de bloques del programa de adquisición de datos.

Así pues, el programa debe mandarse ejecutar de la siguiente manera:

\$ adqdata.exe nomarch

donde **nomarch** es el nombre del archivo donde se desea que sean guardados los datos; si ya existe, el programa lo sobrescribe y si no existe, el programa lo crea.

Este programa puede interrumpirse en cualquier momento presionando la tecla de interrupción (generalmente es <SUPR>).

En la figura 4.14 se presenta el diagrama de bloques del programa de adquisición de datos; en él se muestran los archivos que genera. Las entradas requeridas para este programa son tomadas directamente del CAMAC. En el apéndice 3 se incluye el código del programa.

Ejecutar **adqdata.exe** tiene los siguientes requerimientos:

1. El usuario debe contar con el permiso de ejecución y su PATH debe incluir el directorio donde se encuentra `adqdata.exe`.
2. El CAMAC debe estar debidamente conectado a la computadora por RS232 puerto A.
3. Debe existir el archivo de dispositivo asociado: `/dev/plt_232_a`.
4. El CAMAC debe estar encendido.
5. La señal del interferómetro (o la señal simulada) debe estar conectada al canal 0 del *interval counter*.
6. Debe contarse con el *software* de ambiente gráfico X11 para la presentación de las gráficas en pantalla (X11 viene instalado junto con el HP-UX).
7. Debe contarse con el graficador `gnuplot`. Este paquete está formado por los archivos:
 - `/usr/local/lib/gnuplot.gih`
 - `/usr/local/bin/gnuplot`
 - `/usr/local/bin/gnuplot_x11`
 (Este paquete también se distribuye por internet en forma gratuita).

Finalmente, si una vez que se han adquirido los datos se desea hacer una inspección adicional de las gráficas, debe llamarse al `gnuplot` de la siguiente manera:

```
$ gnuplot adqdata.gnu
```

esto mostrará las gráficas correspondientes a la última toma de datos realizada.

4.4.- Proceso de datos

El programa de proceso de datos `procddata.c` es un programa en lenguaje C para UNIX que utiliza la “biblioteca de funciones” (descrita en el capítulo 3) más ciertas funciones que se describen más adelante. Se encarga de:

1. Leer un archivo de datos de adquisición (espera un archivo con los datos que arroja `adqdata.exe`: Número de cuenta, cuenta, cuenta acumulada e inverso de cuenta).
2. Hacer una doble suavización de los datos por el método de suavización exponencial.
3. De los datos suavizados (cuenta suavizada), encontrar los máximos y registrar el número de cuentas observadas entre un máximo y otro. Tal como se describió anteriormente, esto

corresponde al número k , es decir, la amplitud del movimiento del oscilador armónico expresado en unidades de $\lambda/4$: $A=k \lambda/4$ con un error de $\pm\lambda/4$ según la ecuación (4.3). Para el laser He-Ne $\lambda=633$ nm.

4. Calcular el tiempo transcurrido entre máximos, según las cuentas originales. Estos datos corresponden a consecutivos valores de la mitad del periodo $T/2$ de una oscilación en unidades de TICKS del reloj del *interval counter*.
5. Guardar en un archivo definido por el usuario , los siguientes datos (en columnas):
 - **Número de cuenta** suavizada (es un consecutivo de 1 a 4095).
 - **Cuenta suavizada** (expresada en TICKS).
 - **Acumulado** de las cuentas suavizadas.
 - **Número k** .
 - **Número $T/2$** (expresado en TICKS).
6. Presentar en pantalla las siguientes gráficas:
 - Cuenta suavizada contra Número de cuenta.
 - Cuenta suavizada contra Acumulado.
 - Números k encontrados.
 - Números $T/2$ encontrados.

En forma adicional, se genera un archivo llamado `procddata.gnu` que contiene las directrices necesarias para volver a generar las gráficas en forma externa haciendo uso del `gnuplot`.

Así pues, el programa puede mandarse ejecutar de la siguiente manera:

\$ `procddata.exe nomarch1 nomarch2`

donde `nomarch1` debe ser el nombre de un archivo del tipo del que arroja `adqdata.exe`; es decir, se espera que contenga cuatro columnas con los datos: Número de cuenta, cuenta, acumulado de cuenta e inverso de cuenta. Por otro lado, `nomarch2` es el nombre del archivo donde se desea sean guardados los datos, si ya existe es sobrescrito y si no existe es creado.

El programa puede ser interrumpido en cualquier momento presionando la tecla de interrupción (generalmente `<SUPR>`).

En la figura 4.15 se presenta el diagrama de bloques del programa de proceso de datos; en él se muestran los archivos de entrada que requiere y los archivos que genera. En el apéndice

4 se incluye el código del programa.

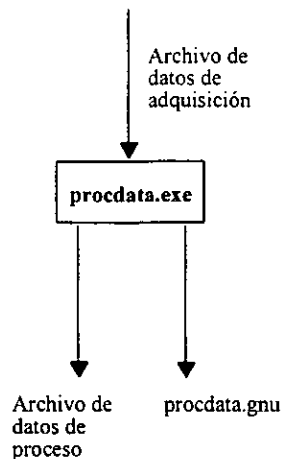


Fig. 4.15. Diagrama de bloques del programa de proceso de datos.

Ejecutar procddata.exe tiene los siguientes requerimientos:

1. El usuario debe contar con el permiso de ejecución y su PATH debe incluir el directorio donde se encuentra procddata.exe.
2. Debe existir el archivo correspondiente nomarch1 con el formato adecuado.
3. Debe contarse con el *software* de ambiente gráfico X11 para la presentación de las gráficas en pantalla (X11 viene instalado junto con el HP-UX).
4. Debe contarse con el graficador gnuplot.

Finalmente, si una vez que se han adquirido los datos se desea hacer una inspección adicional de las gráficas, debe llamarse al gnuplot de la siguiente manera:

\$ gnuplot procddata.gnu

esto mostrará las gráficas correspondientes al último proceso de datos realizado.

Adicionalmente, se generó un tercer programa para facilitar la visualización y/o impresión de las gráficas tanto de adquisición como de proceso, referentes a cierta toma de datos. Es un programa escrito en lenguaje C para UNIX llamado **grafdata.c**.

Requiere que los archivos de datos generados por adqdata.exe y procddata.exe referentes a una misma muestra, lleven por nombre adqnn.dat y procnn.dat, donde nn es un sufijo formado por un número de dos dígitos; por ejemplo adq05.dat y proc05.dat.

Así pues, el programa puede mandarse ejecutar de la siguiente manera:

\$ grafdata.exe nn

donde *nn* es el sufijo que se mencionaba.

Lo que este programa hace es preparar los datos necesarios con cierto formato, y luego hacer llamadas sucesivas a una utilería para UNIX llamada *texdrive* (que en el caso de la máquina que se está usando, se llama *td150*). Esta utilería es capaz de procesar y/o imprimir textos en formato de *tex*, *latex* y/o *postscript*.

Cada llamada a *td150* tendrá listos los datos para visualizar y/o imprimir gráficas en el siguiente orden:

llamada	grafica
1	Cuenta contra Número de cuenta
2	Cuenta contra Cuenta acumulada
3	Inverso de cuenta contra número de cuenta
4	Inverso de cuenta contra cuenta acumulada ..
5	Número de cuenta contra Cuenta acumulada.
6	Cuenta suavizada contra Número de cuenta
7	Cuenta suavizada contra Acumulado de cuenta suavizada
8	Números k.
9	Números T/2.

Para visualizar y/o imprimir cualquier gráfica debe elegirse sucesivamente del menú que aparece en pantalla (que es el propio de *td150*):

- L Para procesar los datos en formato *latex*. Esto es indispensable para ver o imprimir gráficas.
- V Para ver en pantalla la gráfica. Sólo es indispensable si se desea ver la gráfica.
- D Para generar archivo *postscript*. Sólo es indispensable para imprimir la gráfica.
- 2 Para generar archivo de impresora. Sólo es indispensable para imprimir la gráfica.
- P Para mandar imprimir la gráfica.
- q Para salir. Es indispensable para pasar a la siguiente gráfica, si se ha visto la última gráfica relativa a esa toma de datos, el programa termina.

En la figura 4.16 se presenta el diagrama de bloques del programa auxiliar para impresión o visualización de gráficas; en él se muestran los archivos de entrada que requiere y los archivos que genera. En el apéndice 5 se incluye el código del programa.

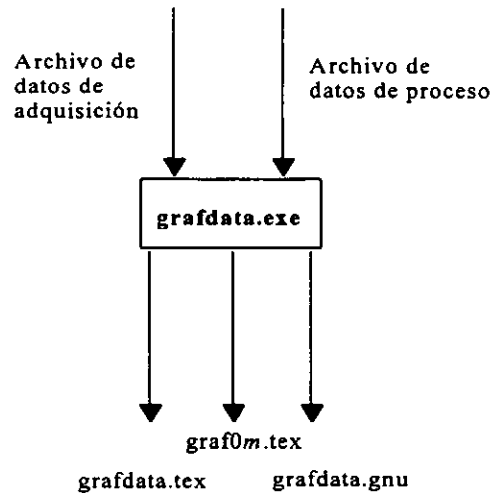


Fig. 4.16. Diagrama de bloques del programa auxiliar de graficación.

Ejecutar `grafdata.exe` tiene pues los siguientes requerimientos:

1. El usuario debe contar con el permiso de ejecución y su PATH debe incluir el directorio donde se encuentra `grafdata.exe`.
2. Deben existir los archivos correspondientes `adqnn.dat` y `procnn.dat` con el formato adecuado.
3. Debe contarse con el graficador `gnuplot`. Este graficador será el encargado de generar, para cada gráfica, el archivo de entrada para `td150` con formato latex llamado `graf0m.tex`; el programa también genera un archivo llamado `grafdata.gnu` que contiene las directrices para la graficación de los datos en `gnuplot`.
4. Debe contarse con el `texdrive (td150)`. (El programa genera un archivo que contiene las variables de ambiente que deben ser incluidas en latex, tal archivo es llamado `grafdata.tex`)

FUNCIONES UTILIZADAS EN EL PROCESO DE DATOS

La función `max_min()` es la encargada de localizar las regiones “menos densas” y contar

los máximos de intensidad o pulsos entre dos regiones de ese tipo, para determinar el número k. A continuación se describe su desempeño.

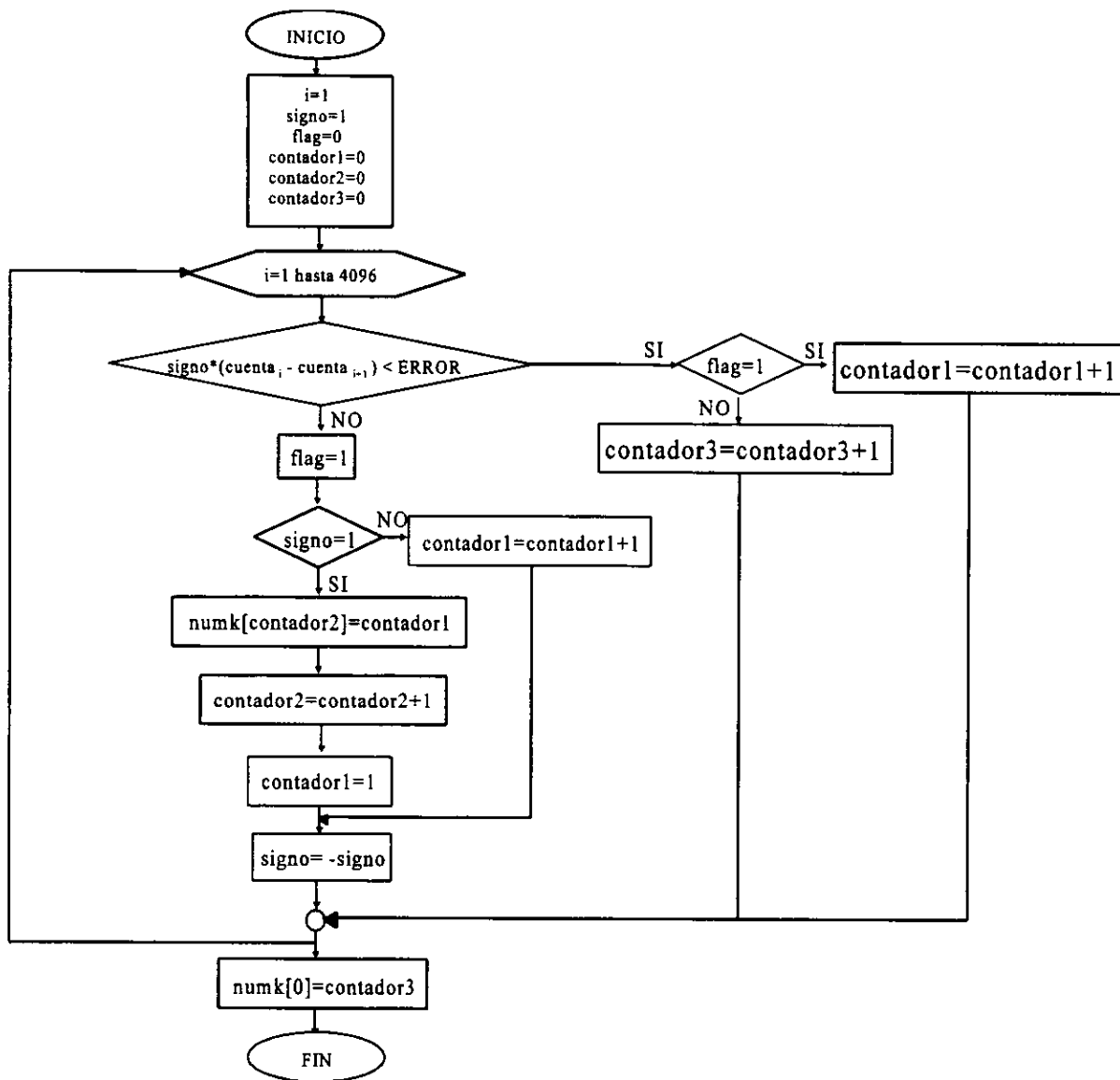


Fig. 4.17. Diagrama de flujo de la función max_min().

Dado un arreglo de 4095 datos, y un valor enter de ERROR procede:

1. Encuentra el primer valor máximo.
2. Cuenta los datos registrados hasta el siguiente mínimo.

3. Sigue contando los datos registrados hasta encontrar el siguiente máximo. Detiene la cuenta y la guarda en el arreglo numk.
4. Encuentra el siguiente valor máximo.
5. Repite desde el paso 2 tantas veces como máximos existan en la muestra.

En la fig. 4.17 se presenta el diagrama de flujo de la función `max_min()`.

`numk` es un arreglo. Cada uno de sus elementos representa el número de pulsos entre dos máximos excepto en el primer elemento donde se almacena la cuenta de pulsos antes del primer máximo. Contador1 lleva la cuenta de pulsos entre máximos. Contador2 lleva la cuenta del número de máximos encontrados. Contador3 lleva la cuenta de pulsos antes del primer máximo.

El prototipo de la función es:

```
int max_min(double cuenta[],int numk[], int error);
```

y devuelve el número de elementos del arreglo `numk`.

En este algoritmo, `ERROR` es una constante. Normalmente debería tener el valor de cero, si los datos siguieran un patrón limpio [...decreciente-creciente-decreciente...] como el de la fig. 4.18.

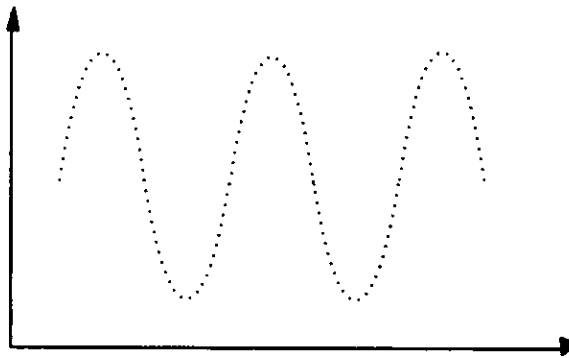


Fig. 4.18. Ejemplo de datos que siguen un patrón no ruidoso.

si `ERROR = 0` se estarán localizando los puntos en que la pendiente cambia de positiva a negativa o de negativa a positiva. Sin embargo, el patrón real de los datos, como se verá más adelante, no es tan limpio. Aún en el caso en que se tuvieran frecuencias de entrada constantes (y no variables como es el caso) los datos estarían dispuestos como se muestra en la fig. 4.19. Observe que la dispersión se debe a error experimental o ruido.

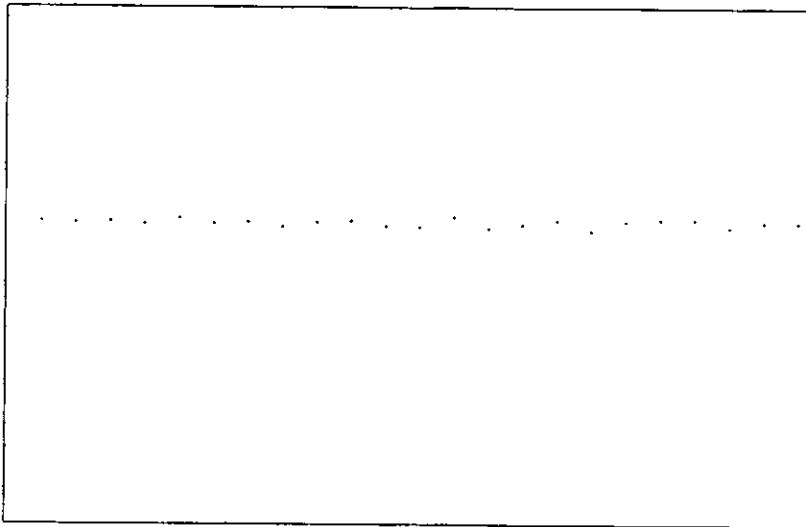


Fig. 4.19. Ejemplo de datos con frecuencia de entrada constante.

El algoritmo anterior, con $ERROR = 0$ sería muy sensible y detectaría máximos y mínimos lo que no es deseable pues son producto de ruido. Para evitar esta situación, conviene incrementar el valor de $ERROR$ lo cual equivale a decir: “Considerese un máximo sólo si es mayor que el siguiente dato por el valor $ERROR$ ”.

Los datos reales resultaron tener patrones típicos de la forma que se muestra en la fig. 4.20.

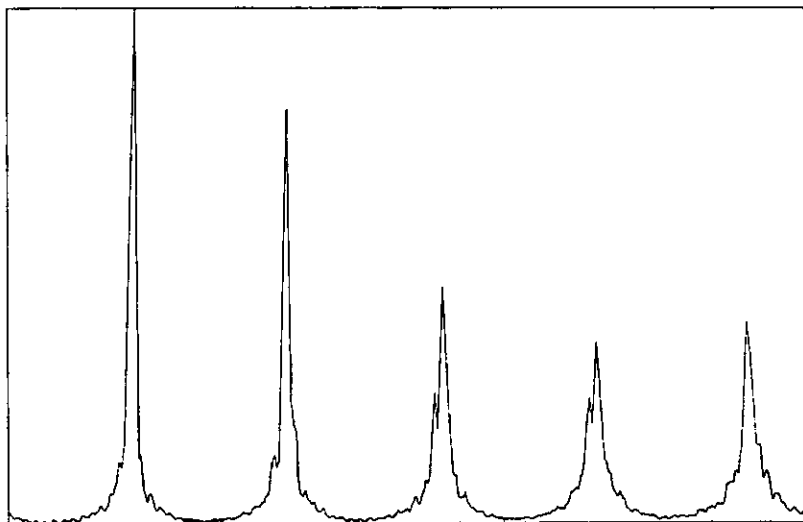


Fig. 4.20. Patrón de datos típico que se ha unido con líneas para mayor claridad.

Un mismo valor de $ERROR$ no resultaba aplicable a diferentes tomas de datos. Podían

existir máximos o mínimos locales que por error experimental cumplieren con el valor de ERROR (fig. 4.21).

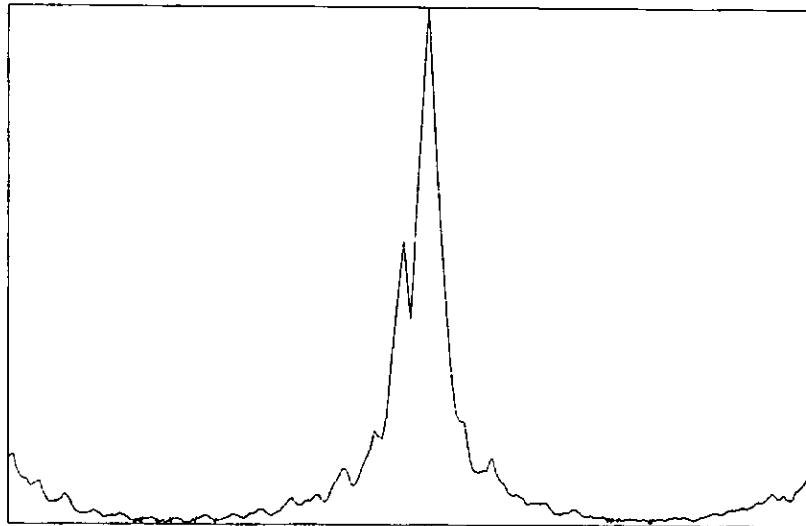


Fig. 4.21. Ejemplo de una toma de datos ruidosa. Los puntos se han unido con líneas para mayor claridad.

La componente ruidosa, hizo necesaria una suavización de curva. Esta es la justificación para la función suavexp(), que hace una suavización exponencial [4.2]. Se tomó la decisión de utilizar este método pues arrojó mejores resultados frente al de promedios móviles (que también se probó). Este método, usualmente utilizado para la elaboración de pronósticos trabaja de la siguiente manera.

Dado un conjunto de 4095 datos cuenta_t y un valor α llamado constante de suavización o factor de descuento ($0 < \alpha < 1$) procede:

1. Elegir una estimación inicial (promedio₁).
2. Calcular las subsecuentes 4094 estimaciones con la fórmula

$$\text{promedio}_t = \alpha * \text{cuenta}_{t-1} + (1-\alpha) \text{promedio}_{t-1}$$

En la fig. 4.22 se presenta el diagrama de flujo de la función suavexp().

Se eligió el primer dato cuenta₁ como estimación inicial.

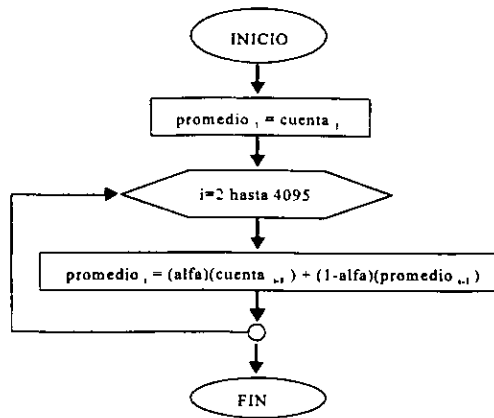


Fig. 4.22. Diagrama de flujo de la función suavexp().

El conjunto de estimaciones o valores suavizados a quienes se ha denominado promedio_i puede verse como la suma ponderada del dato anterior y el promedio anterior, es decir

$$F_i = \alpha \text{cuenta}_{i-1} + \alpha(1-\alpha)\text{cuenta}_{i-2} + \alpha(1-\alpha)^2 \text{cuenta}_{i-3} + \dots ;$$

el suavizamiento exponencial da el mayor peso a la observación actual y pesos decrecientes a las observaciones anteriores. Como originalmente es para pronósticos, sólo toma en cuenta los datos anteriores, pero fué suficiente para los fines de este trabajo.

Con un valor de α pequeño, los promedios resultantes son más suaves, un valor de α mayor, tiende a conservar la variabilidad de los datos originales.

La función suavexp() prototipo es

```
void suavexp(double cuenta[], int alfa, double promedio[]);
```

y fue llamada dos veces en el programa de proceso de datos.

En suma, por la variabilidad inherente a los datos experimentales obtenidos, fue necesario hacer una doble suavización exponencial y determinar un valor de ERROR para la identificación de máximos.

En todas las pruebas que se hicieron, se usaron los valores para ERROR y α :

$$\text{ERROR}=100, \alpha=0.1 ,$$

y resultaron satisfactorios; su elección fue hecha en forma empírica.

4.5.- Resultados

A continuación se presentan los resultados obtenidos para un ejemplo de movimiento oscilatorio. Por facilidad, se usará el término pulso en vez de máximo de intensidad de la señal

interferométrica.

He aquí un fragmento del archivo que arrojó el programa de adquisición de datos:

1	20819	20819	0.00004803304673615448	21	16289	358962	0.00006139112284363682
2	20480	41299	0.00004882812500000000	22	16170	375132	0.00006184291898577613
3	19384	60683	0.00005158893933140735	23	15723	390855	0.00006360109393881575
4	18956	79639	0.00005275374551593163	24	15510	406365	0.00006447453255963895
5	18421	98060	0.00005428586938819825	25	15583	421948	0.00006417249566835655
6	17956	116016	0.00005569169079973268	26	15291	437239	0.00006539794650447976
7	17494	133510	0.00005716245569909683	27	15331	452570	0.00006522731720044354
8	16601	150111	0.00006023733510029516	28	15286	467856	0.00006541933795629989
9	16388	166499	0.00006102025872589700	29	14696	482552	0.00006804572672836147
10	15734	182233	0.00006355662895640016	30	14588	497140	0.000068549492733753
11	15475	197708	0.00006462035541195476	31	14616	511756	0.00006841817186644773
12	15428	213136	0.00006481721545242416	32	13886	525642	0.00007201497911565606
13	15094	228230	0.00006625149065853981	33	13083	538725	0.00007643506840938622
14	15359	243589	0.00006510840549514942	34	14952	553677	0.00006688068485821294
15	15779	259368	0.00006337537233031244	35	13813	567490	0.00007239556939115326
16	16073	275441	0.00006221613886642194	36	13954	581444	0.00007166403898523721
17	16620	292061	0.00006016847172081829	37	13954	595398	0.00007166403898523721
18	16830	308891	0.00005941770647653000	38	14006	609404	0.00007139797229758674
19	16943	325834	0.00005902142477719412	39	14321	623725	0.00006982752601075344
20	16839	342673	0.00005938594928439931	40	14502	638227	0.00006895600606812854

como ya se mencionó en la sección 4.3, en todos los archivos generados por este programa, cada columna tiene los siguientes datos:

Columna 1.- (número de pulso) Es el registro de llegada de cada pulso (pulso1, pulso2, etc), correspondiente a un desplazamiento $\lambda/2$ del oscilador.

Columna 2.- (periodo interferencial) Es el tiempo transcurrido entre cada dos pulsos desde la llegada de un pulso inicial (pulso 0) o en otras palabras, es el tiempo empleado por el oscilador para recorrer la distancia $\lambda/2$. Sus unidades son décimas de microsegundo.

Columna 3.- (tiempo) Es el tiempo que ha transcurrido desde la llegada del pulso 0. Sus unidades son décimas de microsegundo (TICKS).

Columna 4.- (frecuencia interferencial) Puede interpretarse como la frecuencia de los pulsos de entrada; es decir, los pulsos por unidad de tiempo; o bien, como los desplazamientos $\lambda/2$ que lleva a cabo el oscilador en una unidad de tiempo; esto es, como la velocidad media del oscilador evaluada cada $\lambda/2$. La unidad de tiempo es una décima de microsegundo.

Los datos de las columnas 1, 2, 3 y 4, se presentan en las figuras 4.23, 4.24, 4.25, 4.26 y 4.27. En el apéndice 3 se incluye la totalidad del archivo que arrojó el programa de adquisición de datos para el ejemplo que está tratándose.

Periodo Interferencial (TICK)

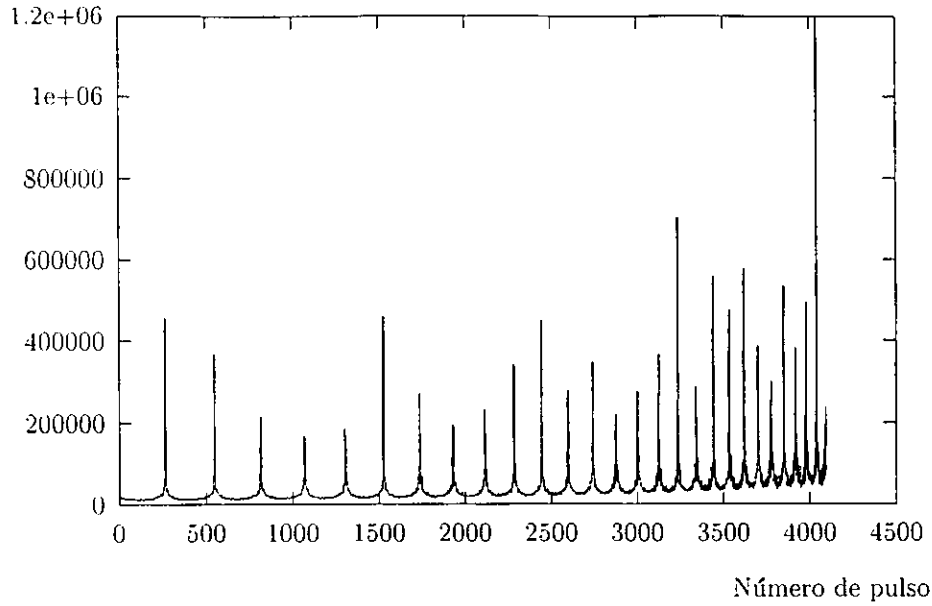


Fig. 4.23. Gráfica del periodo interferencial contra el número de pulso. Para mayor claridad, los puntos experimentales están unidos con rectas.

Periodo Interferencial (TICK)

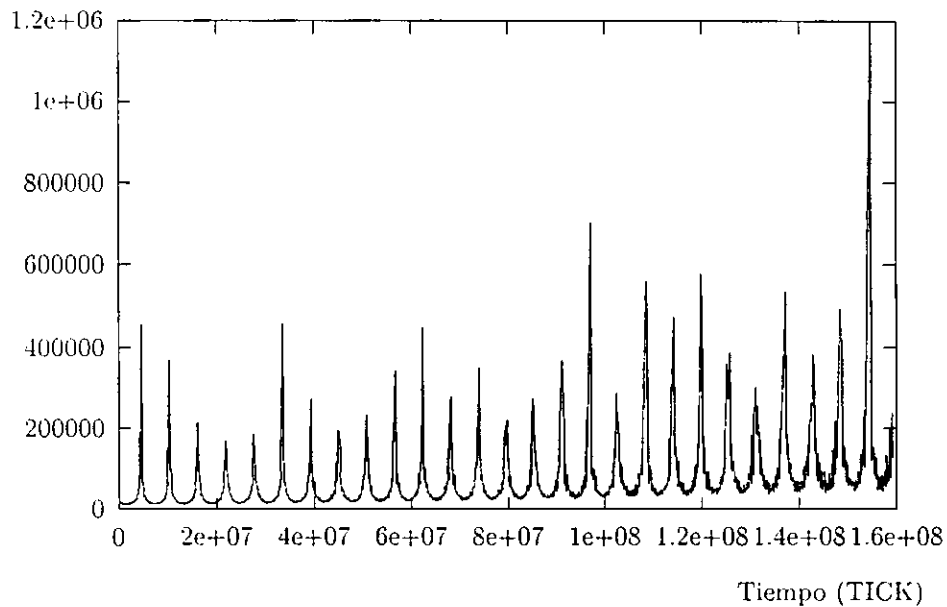


Fig. 4.24. Gráfica del periodo interferencial contra el tiempo. Para mayor claridad, los puntos experimentales están unidos con rectas.

Frecuencia Interferencial (1/TICK)

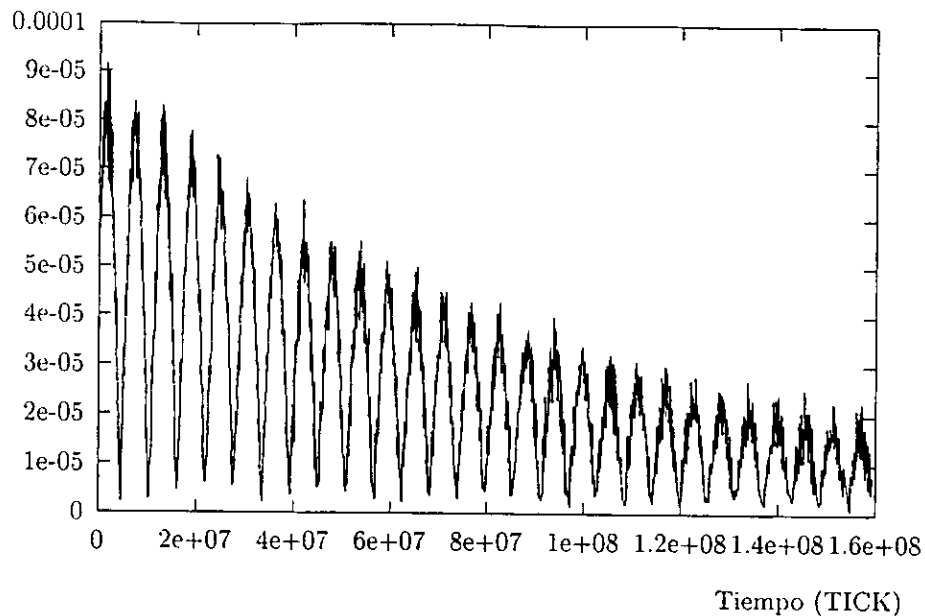


Fig. 4.25. Gráfica de la frecuencia interferencial contra el tiempo. Para mayor claridad, los puntos experimentales están unidos con rectas.

Frecuencia Interferencial (1/TICK)

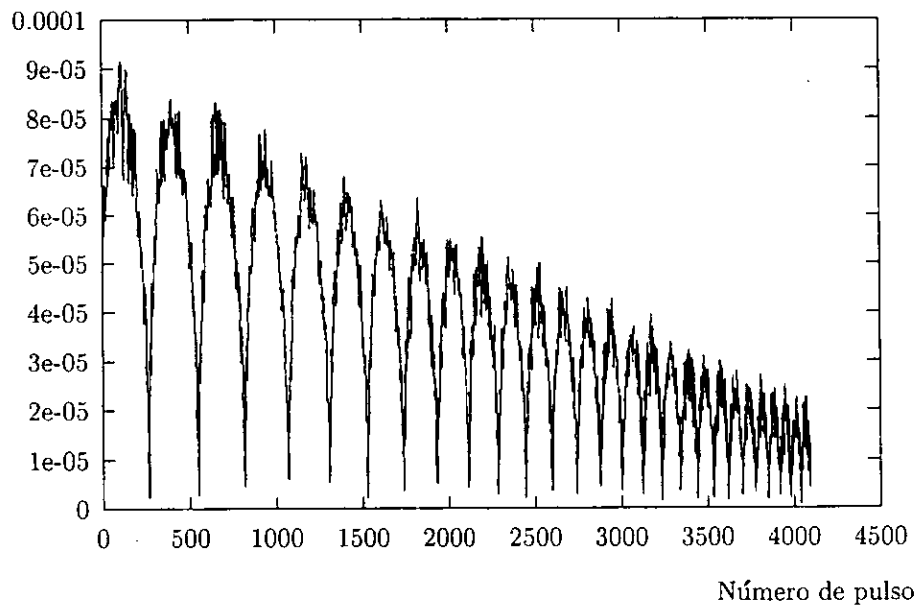


Fig. 4.26. Gráfica de la frecuencia interferencial contra el número de pulso. Para mayor claridad, los puntos experimentales están unidos con rectas.

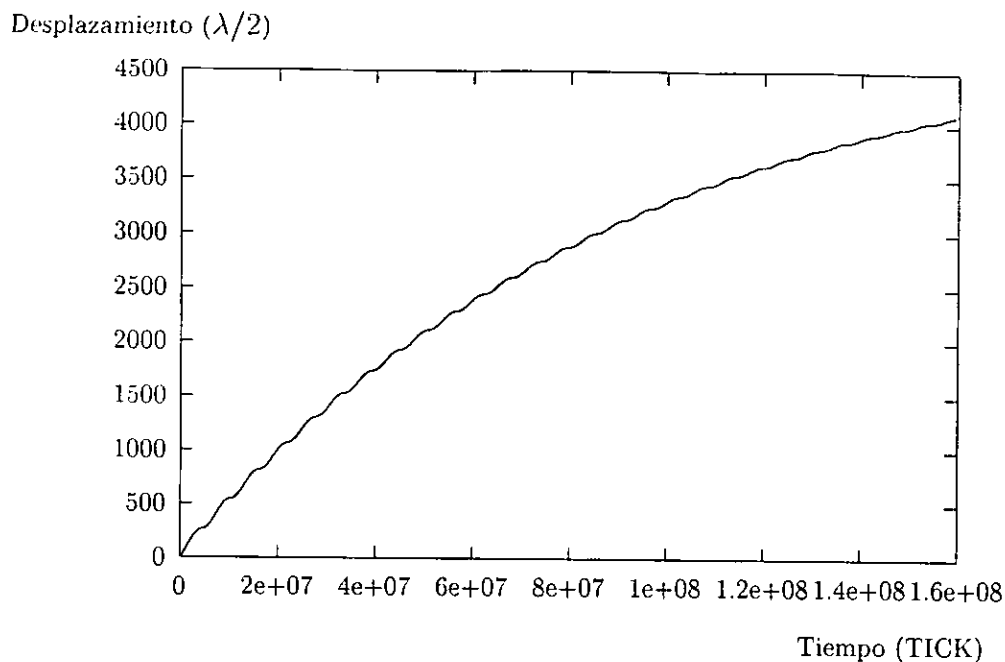


Fig. 4.27. Gráfica del desplazamiento del oscilador mecánico contra el tiempo. Para mayor claridad, los puntos experimentales están unidos con rectas.

En la fig. 4.23 se muestran los resultados para una muestra típica de la señal interferométrica proporcionada por un oscilador mecánico. Puesto que el periodo interferencial es proporcional al inverso de la velocidad del oscilador mecánico, ocurre que se tienen singularidades en que la señal se va a infinito. Estas singularidades corresponden a la velocidad igual a cero, en donde el oscilador mecánico retorna.

Estos retornos pueden observarse como máximos pronunciados del periodo interferencial. La disminución en la distancia entre estas singularidades es consecuencia de la disminución de la amplitud del oscilador mecánico (amortiguamiento); considerando que el número de pulso es equivalente a desplazamientos $\lambda/2$ del oscilador. Por otro lado, se puede notar que el valor de los mínimos del periodo interferencial se va incrementando con el número de pulso. Esto corresponde al amortiguamiento de la velocidad en el oscilador mecánico.

En la fig. 4.24 se muestra la misma información que en la 4.23, excepto por el hecho de que se ha graficado el periodo interferencial contra el tiempo en vez de contra el número de pulso. Puede notarse en esta gráfica la isocronía del oscilador (frecuencia constante) pues el tiempo entre singularidades es constante.

Puesto que las singularidades mencionadas corresponden a puntos de retorno del oscilador armónico (velocidad=0), constituyen una excelente referencia para localizar los semiperiodos del oscilador mecánico o regiones de mínima densidad (según se dijo en la sección 4.1). Esto quiere decir que la amplitud del oscilador armónico es el número de pulsos entre dos singularidades, en unidades de $\lambda/4$.

Esta característica de la gráfica hizo posible la localización de máximos de densidad por métodos digitales y no analógicos como en el trabajo de la referencia [4.1].

Considerando que la frecuencia interferencial es proporcional a la velocidad del oscilador, las figuras 4.25 y 4.26 muestran la señal senoidal rectificada de tal velocidad. Como se sabe, si la posición del oscilador mecánico está dada por

$$x = A \text{sen}(\omega t + \phi_0),$$

entonces la velocidad es

$$v = dx/dt = A \omega \cos(\omega t + \phi_0);$$

donde A es la amplitud del oscilador y $\omega = 2\pi f$. La señal interferométrica no detecta el signo de la velocidad, es por ello que se obtiene siempre positiva; esto explica la rectificación antes mencionada. En la gráfica los valores son siempre positivos.

La fig. 4.25 muestra claramente el amortiguamiento del oscilador mecánico pues la amplitud de la velocidad va decayendo exponencialmente en el tiempo. Este resultado se puede deducir de la expresión de la posición del oscilador mecánico en la que A se sustituye por

$$A = A_0 e^{-pt},$$

donde p es la constante de amortiguamiento. Derivando con respecto al tiempo la nueva posición, se obtiene la velocidad

$$v = dx/dt = A_0 \omega e^{-pt} \cos(\omega t + \phi_0),$$

en donde se ha despreciado la derivada de la parte exponencial por ser muy pequeña. La fórmula anterior muestra que la amplitud de la velocidad decae exponencialmente en el tiempo si la amplitud de la oscilación lo hace.

Por otro lado, en la fig. 4.26 se observa que la amplitud de la velocidad va decayendo linealmente respecto a la distancia que es recorrida (considerando que el número de pulso es equivalente a desplazamientos $\lambda/2$ del oscilador). Esto sólo se explica por el hecho de que el

amortiguamiento es exponencial. Obsérvese en la fig. 4.28 que se ha presentado al ángulo θ como constante (la amplitud decae linealmente); lo cual se deduce según el siguiente tratamiento.

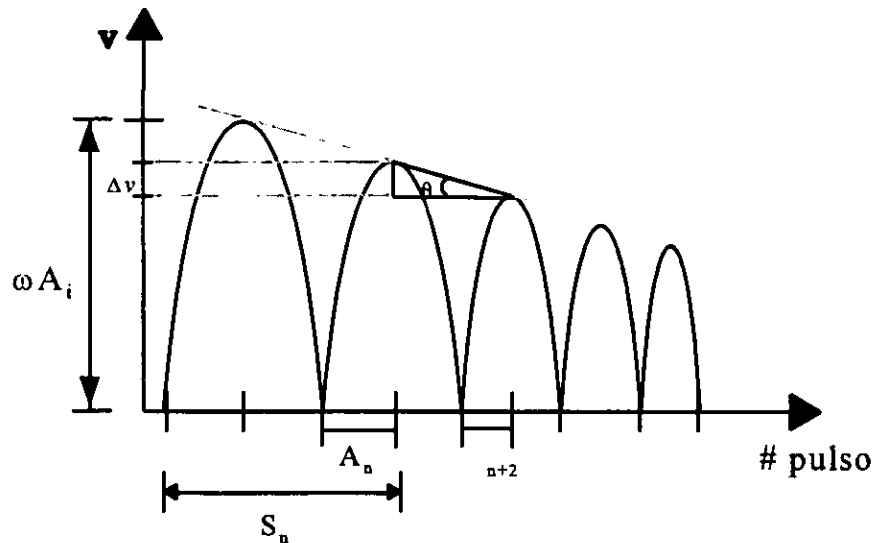


Fig. 4.28. Decaimiento lineal de la amplitud de la velocidad.

Dado que se tiene un mínimo toda vez que el movimiento oscilatorio alcanza su máxima amplitud y un máximo toda vez que el movimiento pasa por la posición de equilibrio, las posiciones máximas y mínimas (puntos críticos) de esta gráfica corresponden cada una de ellas a desplazamientos iguales a la amplitud del oscilador.

Así pues, sea $S_n = A_1 + A_2 + \dots + A_n$ la distancia recorrida hasta el punto crítico n , con los valores A_i correspondientes a valores de la amplitud en unidades de $\lambda/2$.

Sea también $S_{n+2} = A_1 + A_2 + \dots + A_n + A_{n+1} + A_{n+2}$ la distancia recorrida hasta el punto crítico $n+2$; entonces $S_{n+2} - S_n = A_{n+1} + A_{n+2}$ es la distancia entre dos máximos adyacentes.

Por otro lado, se sabe que la amplitud de la velocidad en el punto crítico n está dada por ωA_n en unidades de $\lambda/2$, donde $A_n = A_0 e^{-p(t+nT/4)}$ y $\omega = 2\pi f$.

Así pues, el elemento $\Delta v = \omega A_{n+2} - \omega A_n$. Con todo ello es posible plantear:

$$\operatorname{tg} \theta = \frac{\Delta v}{S_{n+2} - S_n} = \frac{\omega (A_{n+2} - A_n)}{A_{n+1} + A_{n+2}} \quad ; \quad (4.4)$$

ahora bien, definiendo ΔA_1 y ΔA_2 como

$$\begin{aligned} \Delta A_1 &= A_{n+1} - A_n = A_0 e^{-p(t+nT/4)} (e^{-p(T/4)} - 1), \\ \Delta A_2 &= A_{n+2} - A_n = A_0 e^{-p(t+nT/4)} (e^{-2p(T/4)} - 1); \end{aligned}$$

y sustituyendo en la ecuación (4.4) se tiene:

$$\operatorname{tg} \theta = \frac{\omega (\Delta A_2 + A_n - A_n)}{\Delta A_1 + A_n + \Delta A_2 + A_n} ,$$

$$\operatorname{tg} \theta = \frac{\omega (\Delta A_2)}{\Delta A_1 + 2A_n + \Delta A_2} ,$$

dividiendo todo por A_n :

$$\operatorname{tg} \theta = \frac{\omega (\Delta A_2)/A_n}{(\Delta A_1/A_n) + (\Delta A_2/A_n) + 2} ,$$

$$\operatorname{tg} \theta = \frac{\omega (e^{-2p(T/4)} - 1)}{(e^{-p(T/4)} - 1) + (e^{-2p(T/4)} - 1) + 2} ,$$

dado que ω , p y T son constantes, entonces $\operatorname{tg} \theta$ también es una constante. Obsérvese que de los datos experimentales es posible estimar $\operatorname{tg} \theta$, T así como ω ; de modo que más adelante sería posible calcular el valor de p (la constante de amortiguamiento) si ello fuera de interés.

En la fig. 4.27 se puede ver el desplazamiento total del oscilador en unidades de $\lambda/2$. Las pequeñas ondulaciones de la curva, muestran las variaciones en velocidad del oscilador.

Ahora, obsérvese un fragmento del archivo que arrojó el programa de proceso de datos:

1	20819.000	20819.000	267	5066048	21	18534.140	420921.253	84	5600132
2	20819.000	41638.000	284	5824703	22	18393.971	439315.224	82	5918703
3	20819.000	62457.000	268	5818416	23	18259.384	457574.608	77	5728125
4	20815.610	83272.610	253	5827649	24	18129.475	475704.083	71	5829306
5	20798.548	104071.158	238	5835993	25	18000.184	493704.267	69	5844784
6	20766.302	124837.460	221	5838856	26	17870.556	511574.823	63	5698625
7	20716.730	145554.191	210	5865083	27	17742.682	529317.505	59	5954026
8	20648.970	166203.160	194	5714903	28	17614.588	546932.093	41	3294161
9	20562.534	186765.694	183	5854519	29	17487.995	564420.088	#	#
10	20452.905	207218.599	172	5761906	30	17363.435	581783.523	#	#
11	20323.457	227542.057	161	5800012	31	17235.867	599019.390	#	#
12	20172.710	247714.767	152	5780268	32	17106.058	616125.449	#	#
13	20003.627	267718.394	144	5817967	33	16976.013	633101.462	#	#
14	19820.914	287539.308	135	5722679	34	16839.776	649941.237	#	#
15	19625.647	307164.956	127	5770343	35	16691.856	666633.093	#	#
16	19424.815	326589.770	122	5876824	36	16554.642	683187.736	#	#
17	19225.682	345815.453	111	5771399	37	16416.083	699603.819	#	#
18	19032.858	364848.311	106	5699688	38	16279.229	715883.048	#	#
19	18852.542	383700.853	99	5914641	39	16145.125	732028.173	#	#
20	18686.260	402387.113	93	5768793	40	16015.110	748043.282	#	#

Como se describió en la sección 4.4, todos los archivos generados por este programa, contienen columnas con los siguientes datos:

Columna 1.- (número de pulso) Representa el registro de cada pulso (pulso 1, pulso 2, etc.).

Columna 2.- (periodo interferencial suavizado) Son los datos obtenidos, relativos al tiempo

transcurrido entre cada dos pulsos, después de haberles aplicado la doble suavización. Sus unidades son décimas de microsegundo. Sobre estos datos se aplicó el algoritmo para encontrar regiones “menos densas”.

Columna 3.- (tiempo suavizado) Puede verse como el tiempo transcurrido desde la llegada del pulso 0 según los datos suavizados. Sus unidades son décimas de microsegundo.

Columna 4.- (número K) Cada elemento de esta columna representa la amplitud del oscilador en unidades de $\lambda/4$ ($A = k \lambda/4$) en media oscilación; con $\lambda = 633 \text{ nm}$. Puede conocerse el número de oscilaciones muestreadas con:

$$\text{número de oscilaciones} = (\text{número de elementos de la columna 4})/2 .$$

Columna 5.- (número T/2) Cada elemento de esta columna contiene el tiempo transcurrido en la mitad de una oscilación; es decir, cada elemento es T/2 la mitad del periodo en unidades de décimas de microsegundo.

En el apéndice 4 se incluye la totalidad del archivo que arrojó el programa de proceso para el ejemplo que está tratándose. Las gráficas asociadas fueron las que se presentan en las figuras 4.29, 4.30, 4.31 y 4.32.

Periodo Interferencial Suavizado (TICK)

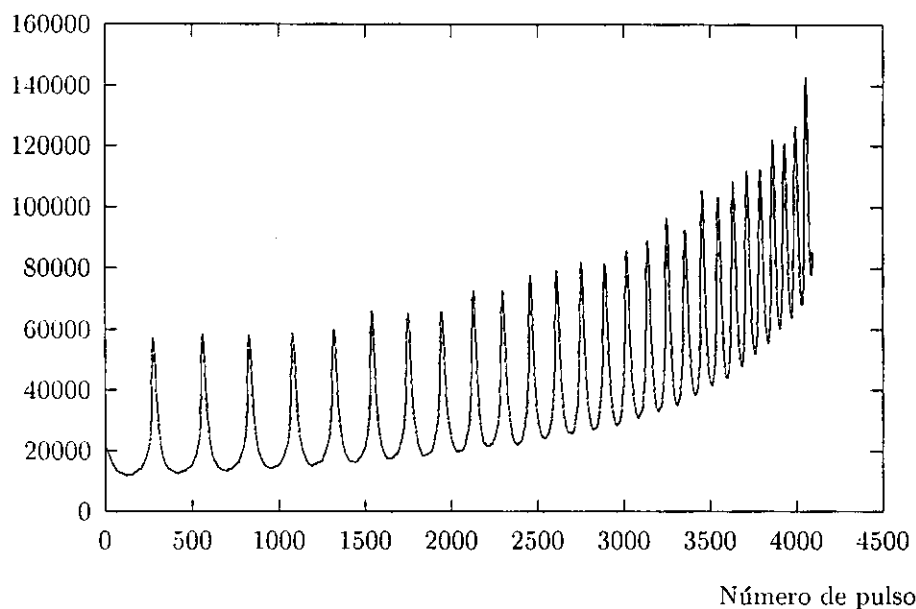


Fig. 4.29. Gráfica del periodo interferencial suavizado contra el número de pulso. Para mayor claridad, los puntos de proceso están unidos con rectas.

Periodo Interferencial Suavizado (TICK)

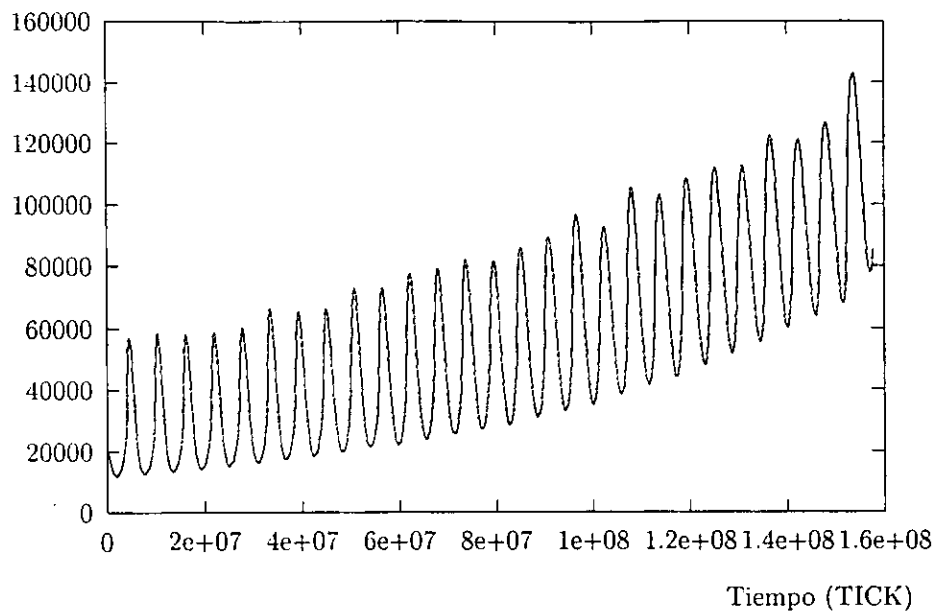


Fig. 4.30. Gráfica del periodo interferencial suavizado contra el tiempo. Para mayor claridad, los puntos de proceso están unidos con rectas.

Amplitud del Oscilador ($\lambda/4$)

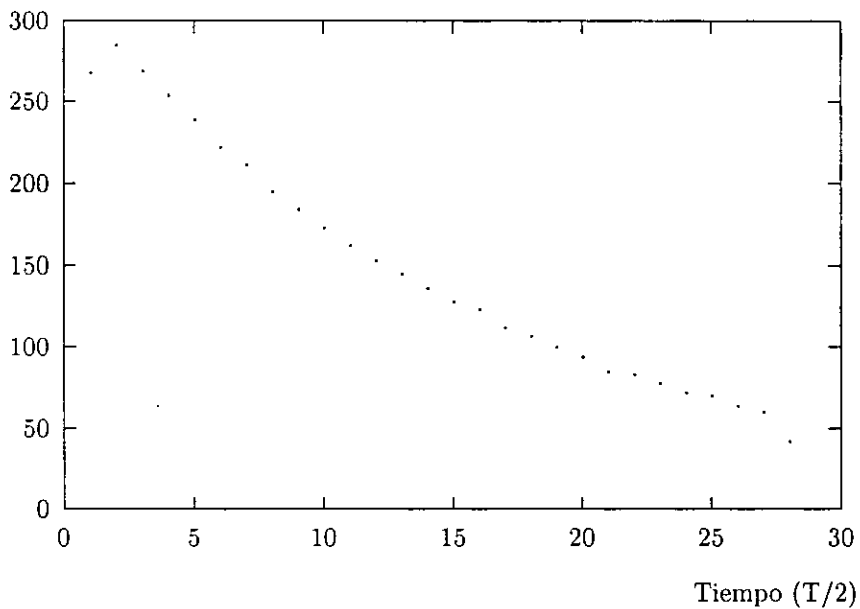


Fig. 4.31. Gráfica de la amplitud del oscilador contra el tiempo.

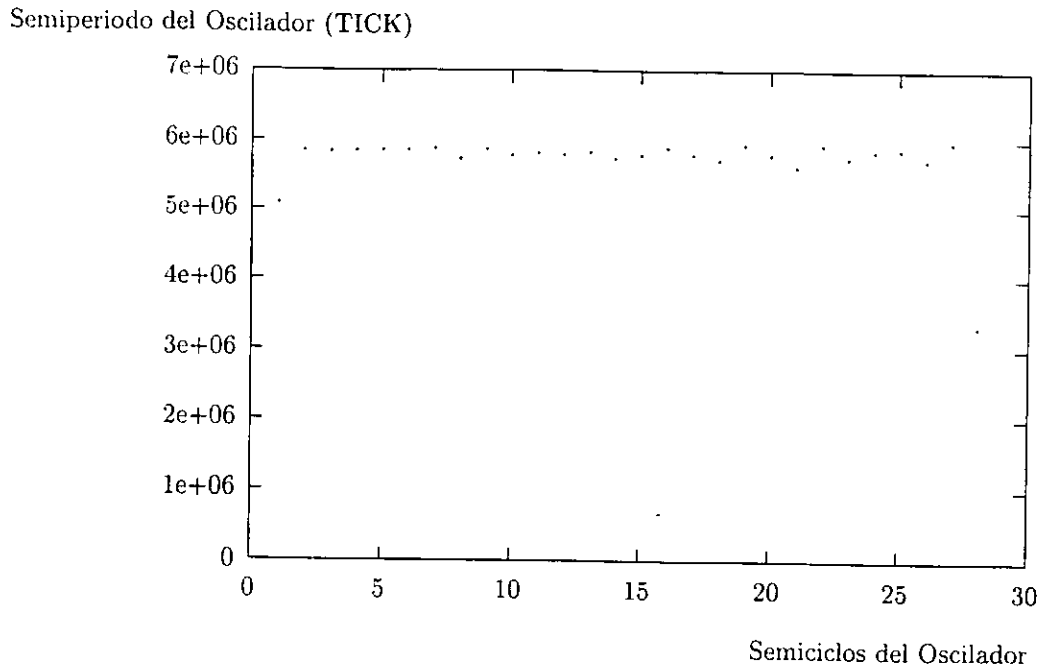


Fig. 4.32. Gráfica del semiperiodo del oscilador ($T/2$).

Las figuras 4.29 y 4.30 corresponden a los datos suavizados, se presentan aquí para contrastar contra los datos reales (figuras 4.23 y 4.24). Obsérvense los efectos de la suavización y nótese que los máximos no cambian de posición respecto de los datos reales.

La fig. 4.31 muestra la gráfica de los números k encontrados o en otras palabras es la gráfica de la amplitud del oscilador muestreada en periodos $T/2$. Obsérvense los efectos del amortiguamiento exponencial.

La fig. 4.32 muestra la gráfica de los números $T/2$ (el tiempo transcurrido en la mitad de una oscilación) que en teoría debe describir una línea recta. Note sin embargo el rango en que se mueve esta variable; tiene una dispersión mínima debida a error experimental.

Es importante mencionar que los puntos primero y último de las figuras 4.31 y 4.32 deben discriminarse dado que el algoritmo maneja un conjunto incompleto de datos al inicio y al final de la muestra.

Así pues, las gráficas no sólo ofrecen información acerca de la amplitud de la oscilación, sino que también se puede analizar el movimiento oscilatorio en cuanto a su velocidad, frecuencia,

desplazamiento y amortiguamiento.

Por ser de gran interés, en el apéndice 6 se muestran también los resultados que corresponden a la transición de movimiento armónico a ruido. La señal de ruido corresponde al registro de la microsismicidad que está presente en el local del laboratorio. Asimismo se incluyen en el apéndice las gráficas correspondientes a una señal exclusivamente de ruido.

Es necesario apuntar que no se hizo análisis alguno del ruido interferométrico, dado que salía fuera del alcance y los objetivos de este trabajo; sin embargo las muestras tomadas dejaron ver al análisis de señales interferométricas de ruido como un interesante campo abierto de investigación para quienes se apasionan de conjuntar el análisis matemático o estadístico con las potencialidades de la computación.

CONCLUSIONES

El presente trabajo de tesis consistió en la conexión física y lógica entre un sistema interferométrico y una estación de trabajo UNIX, a través de un sistema de adquisición de datos CAMAC. Todo ello con el objetivo primero de determinar la amplitud de un oscilador mecánico arbitrario haciendo uso de técnicas interferométricas.

Esta tesis es un ejemplo del indudable espíritu multidisciplinario del mundo de las computadoras, en este caso, aplicado al campo de la Física.

En primera instancia, fue necesario adentrarse un poco en los terrenos de la Óptica para lograr la comprensión del funcionamiento del interferómetro de Michelson, ya que éste se basa en el fenómeno de interferencia de la luz. Muy importante fue describir matemáticamente la señal eléctrica que arroja el sistema interferométrico ya que ésta sería la señal de entrada del sistema de adquisición de datos; es decir, la señal interferométrica que se analizaría con el objeto de estudiar un fenómeno físico: el movimiento de un oscilador armónico.

Por otro lado, hubo que hacer un estudio detallado del funcionamiento del sistema de adquisición de datos CAMAC, en cuanto a sus especificaciones, sus parámetros de transmisión y protocolos de transferencia de datos. Hubo que evaluar el desempeño de los módulos con que se contaba para elegir aquellos a usar en la aplicación y hubo que preparar a la computadora para lograr la comunicación con el CAMAC.

Una vez que se logró la conexión física entre CAMAC y computadora por RS232, se prosiguió a la conexión lógica. De esta actividad resultó una “biblioteca de funciones” programada en lenguaje C para UNIX, diseñada especialmente para resolver la comunicación con el gabinete multitarjetas y apoyar en el intercambio y proceso de datos. Esta biblioteca se probó echando a

andar un sistema de control de temperatura que se eligió por ser simple y de funcionamiento conocido.

Finalmente se diseñó el experimento para determinar la amplitud del movimiento del oscilador armónico. En este punto fue necesario generar los programas necesarios de adquisición y proceso de los datos. En una primera fase, se hizo una simulación sencilla de la señal interferométrica haciendo uso de dos generadores de funciones, finalmente se cerró la conexión con el interferómetro y se corrió la aplicación.

Una vez que se obtuvieron resultados, fue posible constatar que no solamente se había tenido éxito en la adquisición de datos del sistema interferométrico haciendo uso del CAMAC, sino además que la técnica utilizada para la determinación de la amplitud del oscilador resultaba ser una mejora a la técnica descrita en la referencia [4.1] en cuanto a que:

- Absolutamente todo el análisis de la señal interferométrica es digitalizado y semiautomático, gracias al uso del sistema de adquisición de datos.
- También gracias al uso del CAMAC y a la estrategia de contar el tiempo entre pulsos es posible obtener mucho más que una tabla de amplitudes. Es posible obtener información acerca de la frecuencia, velocidad, desplazamiento y amortiguamiento del móvil.
- Se detectó un error en la referencia [4.1] en cuanto a la fórmula para calcular la amplitud y se efectuó la corrección pertinente.

Es por ello que puede afirmarse que se hizo una aportación (aunque modesta) a la interferometría. Por supuesto que no todo en este trabajo es miel sobre hojuelas; es posible destacar que no se utilizó la tecnología de punta que rige a los sistemas de adquisición de datos. Esto significa que el uso de interfaces de mayor velocidad (como SCSI o GI) o el uso de gabinetes multitarjetas de mayor capacidad (como VXI) hubiese podido arrojar aún mejores resultados; muestras más grandes de datos, posiblemente disponibles en tiempo real y/o un rango más amplio de frecuencias interferométricas medibles.

Así pues, de este trabajo puede concluirse que:

- Efectivamente se logró conectar al sistema interferométrico con la estación de trabajo UNIX, teniendo como intermediario a un sistema de adquisición de datos CAMAC. La “biblioteca de funciones” y demás programas que se presentan en este trabajo (incluyendo al control de temperatura), la descripción que se hace de los gabinetes multitarjetas (principalmente de CAMAC) y la compilación de información acerca del manejo de dispositivos en UNIX puede ser de utilidad para futuras aplicaciones especialmente relativas a instrumentación, en las inmediaciones del laboratorio.
- Se tuvo éxito en el análisis de la señal interferométrica. Los resultados obtenidos superaron las expectativas al ofrecer información tanto de la amplitud del movimiento del oscilador como de su velocidad, frecuencia, amortiguamiento y desplazamiento. Resulta muy satisfactorio haber propuesto una técnica mejorada para el análisis de un oscilador.
- Finalmente se lanza el reto para quien decida aceptarlo de hacer análisis digitalizado de señales interferométricas de ruido con sus posibles implicaciones en el área de la sismografía, considerando que estas señales constituyen registros de microsismicidad.

APENDICE 1

A continuación se presenta el código fuente de la “biblioteca de funciones” `cmc_lib.c` :

```

/*****
NOMBRE      cmc_lib.c
AUTOR       Brenda Eugenia Olivos Fuentes
FECHA DE CREACION: 14/jul/97
DESCRIPCION Este programa define las funciones necesarias
              para la comunicacion con el CAMAC, a traves del
              archivo plt_rs232_a, se configura para transmision
              serial a B19200, palabras de 8b, 2 bits de stop
              y paridad non.
*****/
#include <stdio.h>
#include <errno.h>
#include <sys/types.h>
#include </usr/include/sys/fcntl.h>
#include </usr/include/sys/termio.h>
#include <stdlib.h> /** para atoi() **/
#include <math.h> /** para pow() **/
#include <signal.h> /** para signal() **/
#include <time.h> /** para setitimer() **/

#define BUFSIZE 20
#define TAM_CMD 35
#define SOH 1
#define STX 2
#define ETX 3
#define TICKS 100.

struct tms {
    time_t tms_utime;
    time_t tms_stime;
    time_t tms_cutime;
    time_t tms_cstime;
};
static struct tms tbuf1;
static long real1;
static struct termio tbufsave,tbuf;
int fd;

void syserr(msg)
char *msg;
{
    /** MANDA A LA SALIDA ESTANDAR, LA CONDICION DE ERROR msg, EL CODIGO
        errno Y LA DESCRIPCION DEL SISTEMA sys_errlist PARA ESE ERROR **/
    extern int errno, sys_nerr;
    extern char *sys_errlist[];
    fprintf(stderr,"error: %s (%d)", msg, errno);
    if (errno>0&&errno<sys_nerr)

```



```

    fprintf(stderr,"%s\n",sys_errlist[errno]);
else
    fprintf(stderr,"\n");
exit(1);
}
static void nada()
{
    /** NO HACE NADA **/
}
void pausa(secs,micro)
unsigned long secs;
long micro;
{
    /** HACE UNA PAUSA POR EL NUMERO DE SEGUNDOS Y MICROSEGUNDOS
        DEFINIDOS POR secs y micro **/
    struct itimerval timer1,timerold;

    timer1.it_value.tv_sec=secs;
    timer1.it_value.tv_usec=micro;
    timer1.it_interval.tv_sec=0;
    timer1.it_interval.tv_usec=0;

    if(signal(SIGALRM,nada)==BADSIG)
        syserr("cambiar default de SIGALRM");
    setitimer(ITIMER_REAL,&timer1,&timerold);
    pause();
}
void timestart()
{ /** TOMA EL TIEMPO DEL SISTEMA **/
    real1=times(&tbuf1);
}
void timestop(msg)
char *msg;
{ /** VUELVE A TOMAR EL TIEMPO DEL SISTEMA Y MANDA A LA SALIDA ESTANDAR
    DE ERRORES LOS TIEMPOS QUE TRANSCURRIERON DESDE LA TOMA PREVIA
    CON timestart() **/
    struct tms tbuf2;
    long real2;

    real2=times(&tbuf2);
    fprintf(stderr,"%s:real%.4f;user%.4f;sys%.4f\n",msg,(real2-real1)/TICKS,(tbuf2.tms_utime-tbuf1.tms_utime)/
        TICKS,(tbuf2.tms_stime-tbuf1.tms_stime)/TICKS);
}

void config()
{ /** CONFIGURA EL PUERTO PARA TRANSFERENCIAS CANONICAS **/
    fd=open("/dev/plt_rs232_a",O_RDWR,0666);
    if(fd==-1)
        syserr("abrir /dev/plt_rs232_a");
    if(ioctl(fd,TCGETA,&tbuf)==-1)
        syserr("recuperar configuracion");
    tbufsave=tbuf;
    /** VALORES COMPATIBLES CON EL CAMAC EN TRANSFERENCIAS CANONICAS: **/
    tbuf.c_iflag=IXON|IXOFF;
    tbuf.c_lflag=ICANON;
    tbuf.c_cflag=B19200|CS8|CSTOPB|CREAD|PARODD;
    tbuf.c_cc[2]=8; /** ERASE = BS **/
    tbuf.c_cc[3]=21; /** KILL = NAK **/
}

```

```

tbuf.c_cc[4]=0; /** EOF = NULL **/
tbuf.c_cc[5]=3; /** EOL = ETX **/
/*****

if(ioctl(fd,TCSETAF,&tbuf)==-1)
    syserr("configurar puerto");
}
void config_NOCANON()
{ /** CONFIGURA EL PUERTO PARA TRANSFERENCIAS NO CANONICAS **/
fd=open("/dev/plt_rs232_a",O_RDWR,0666);
if(fd==-1)
    syserr("abrir /dev/plt_rs232_a");
if(ioctl(fd,TCGETA,&tbuf)==-1)
    syserr("recuperar configuracion");
tbufsave=tbuf;
/** VALORES COMPATIBLES CON EL CAMAC EN TRANSFERENCIAS NO CANONICAS: **/
tbuf.c_lflag&=~ICANON;
tbuf.c_iflag=IXON|IXOFF;
tbuf.c_cflag=B19200|CS8|CSTOPB|CREAD|PARODD;
tbuf.c_cc[4]=BUFSIZE; /** MIN **/
tbuf.c_cc[5]=1; /** TIME **/
if(ioctl(fd,TCSETAF,&tbuf)==-1)
    syserr("configurar puerto");
}

void restore()
{ /** RESTAURA LA CONFIGURACION ORIGINAL **/
if(ioctl(fd,TCSETAF,&tbufsave)==-1)
    syserr("restaurar configuracion");
}

int CDREG(c,n,a,dir)
char c,n,a;
char *dir;
{ /** declara CAMAC register **
CONSTRUYE LA DIRECCION EN FORMATO CAMAC
ALMACENA LA DIRECCION EN dir Y DEVUELVE EL No DE
CARACTERES EN LA CADENA SIN CONTAR EL '\0' **/
int total;
total=sprintf(dir,"%c%d,%d",c,n,a);
return total;
}

int CDINT24(dato_a,dato_b,dato_c,data)
unsigned char dato_a,dato_b,dato_c;
char *data;
{ /** declara datos **
CONSTRUYE LOS DATOS EN FORMATO CAMAC 24b
LOS ALMACENA EN data Y DEVUELVE EL No DE
CARACTERES EN LA CADENA SIN CONTAR EL '\0' **/
int total;
total=sprintf(data,"%d,%d,%d",dato_a,dato_b,dato_c);
return total;
}

int CDINT16(dato_a,dato_b,data)
unsigned char dato_a,dato_b;
char *data;
{ /** declara datos **
CONSTRUYE LOS DATOS EN FORMATO CAMAC 16b

```

```

    LOS ALMACENA EN data Y DEVUELVE EL No DE
    CARACTERES EN LA CADENA SIN CONTAR EL '\0' **/
    int total;
    total=sprintf(data,"%d,%d",dato_a,dato_b);
    return total;
}
int CDINT8(dato_a,data)
unsigned char dato_a;
char *data;
{ /** declara datos **
  CONSTRUYE LOS DATOS EN FORMATO CAMAC 8b
  LOS ALMACENA EN data Y DEVUELVE EL No DE
  CARACTERES EN LA CADENA SIN CONTAR EL '\0' **/
  int total;
  total=sprintf(data,"%d",dato_a);
  return total;
}
int CFSA(f,dir,data)
char f, *dir, *data;
{
  /** CONSTRUYE COMANDO CAMAC SIMPLE SEGUN f,dir,data (NO CONSIDERA CHKSUM)
  ENVIA COMANDO, LEE RESPUESTA Y LA ALMACENA EN data , regresa el
  NUMERO DE CARACTERES LEIDOS COMO RESPUESTA CAMAC O -1 SI FUE UN
  CODIGO DE FUNCION NO VALIDO**/
  char comando[TAM_CMD];
  int total,lo_i;
  char *lectura="%c%s%d%c%c";
  char *escritura="%c%s%d%c%s%c";
  char *control="%c%s%d%c";
  if(f>=0&&f<=7)
    total=sprintf(comando,lectura,SOH,dir,f,STX,ETX);
  else if(f>=16&&f<=23)
    total=sprintf(comando,escritura,SOH,dir,f,STX,data,ETX);
  else if((f>=8&&f<=15)||((f>=24&&f<=31)))
    total=sprintf(comando,control,SOH,dir,f,STX);
  else { fprintf(stderr,"funcion no valida");
    return -1;
  }
}

if(write(fd,comando,total)==-1)
  syserr("enviar comando");
total=read(fd,data,BUFSIZE);
if(total==-1)
  syserr("leer respuesta");
else {
  data[total]='\0';
  return total;
}
}
int CTCI(resp,i_data)
/** SELECCIONA SOLO LOS DATOS DE LA RESPUESTA CAMAC
  LOS ALMACENA EN EL ARREGLO DE ENTEROS i_data , REGRESA
  EL NUMERO DE ENTEROS EN EL ARREGLO (HASTA 4, PUEDE INCLUIR
  STATUS **/
char *resp;
unsigned char *i_data;

{

```

```

int lo_i,lo_j;
char a_data[4], *p_resp;
for(lo_i=0;lo_i<4;lo_i++)
    i_data[lo_i]=0;
p_resp=resp;
p_resp+=3;
lo_j=0;
while((*p_resp)!='\0'){
    lo_i=0;
    while((*p_resp!=',')&&(*p_resp!=ETX)){
        a_data[lo_i]=*p_resp;
        ++p_resp;
        ++lo_i;
    }
    a_data[lo_i]='\0';
    i_data[lo_j]=atoi(a_data);
    ++lo_j;
    ++p_resp;
}
return lo_j;
}

```

```

int CHECK_BIT(i_data,byte,bit)

```

```

unsigned char *i_data;

```

```

int byte;

```

```

char bit;

```

```

{

```

```

    /** CHECA SI ESTA PRENDIDO EL BIT bit (de 1 a 32) EN EL ARREGLO
    DE ENTEROS i_data QUE SOLO PUEDE CONTENER 4 ENTEROS O BYTES,
    EL USUARIO INFORMA EL NUMERO DE BYTES QUE TIENE i_data Y EL BIT
    UNO SERA EL DE MAS A LA DERECHA Y EL DE MAS A LA IZQUIERDA SERA
    EL BIT 32 SI 4 BYTES, EL BIT 24 SI 3 BYTES, EL BIT 16 SI 2 BYTES
    O EL BIT 8 SI UN BYTE. REGRESA 1 SI ESTA PRENDIDO Y 0 SI APAGADO
    *****/

```

```

    unsigned char dato;

```

```

    if(byte<1||byte>4){
        fprintf(stderr,"NUMERO DE BYTES NO VALIDO");
        return -1;
    }

```

```

    if(bit>=1&&bit<=8){
        dato=(unsigned char)pow(2.0,(double)(bit-1));
        --byte;
    }

```

```

    else if(bit>=9&&bit<=16&&byte>1){
        dato=(unsigned char)pow(2.0,(double)(bit-9));
        byte-=2;
    }

```

```

    else if(bit>=17&&bit<=24&&byte>2){
        dato=(unsigned char)pow(2.0,(double)(bit-17));
        byte-=3 ;
    }

```

```

    else if(bit>=25&&bit<=32&&byte==4){
        dato=(unsigned char)pow(2.0,(double)(bit-25));
        byte-=4 ;
    }
}

```

```

else {
    fprintf(stderr,"BIT NO VALIDO");
    return -1;
}
if(i_data[byte]&dato)
    return 1;
else
    return 0;
}
void espera_NULL()
{ /** PARA EL MANEJO DEL BREAK; ESPERA LA LLEGADA DE UN CARACTER,
    DEBE USARSE SOLO EN EL CASO DE TENER LA CERTEZA DE QUE CAMAC
    ENVIARA UN CARACTER. DEBE CONFIGURARSE MODO CANONICO Y EOF=NUL
    O BIEN MODO NO CANONICO CON MIN>0 **/
char data[2];

    if(read(fd,data,1)==-1)
        syserr("esperar BREAK");
}
void ver_resp(data)
char *data;
{
    /** MUESTRA EN PANTALLA LA CADENA data QUE SE SUPONE SER UNA
        RESPUESTA CAMAC, TAMBIEN MUESTRA EN PANTALLA EL CODIGO ASCII
        DE CADA UNO DE LOS CARACTERES QUE LA COMPONEN **/
    int i;
    printf("\nrespuesta: (%s), ASCII: [",data);
    for(i=0;data[i]!='\0';i++)
        printf("%d.",data[i]);
    printf("]");
}
void ver_bytes(num,i_data)
int num;
unsigned char *i_data;
{
    /** MUESTRA EN PANTALLA TANTOS ENTEROS DEL ARREGLO i_data
        COMO SE ESPECIFIQUE EN num **/

    int i;
    printf("\nbytes: [");
    for(i=0;i<num;i++)
        printf("%d.",i_data[i]);
    printf("]");
}

```

APENDICE 2

A continuación se presenta el código fuente del programa para control de temperatura volt08.c :

```
/******  
NOMBRE      volt08.c  
AUTOR       Brenda Eugenia Olivos Fuentes  
DESCRIPCION Este programa trabaja con transferencias de 24b. Lee el voltaje del canal  
             0 del modulo ADC (N8) y envia voltaje por el canal 3 del modulo DAC (N1).  
             Debera compilarse con cmc_lib.o.  
*****/  
#include <stdio.h>  
#include <math.h>  /** para pow() **/  
#define TAM_ADD 9  
#define BUFSIZE 20  
main()  
{  
    char c,n,a,f,bit,signo;  
    unsigned char x,y,z,i_data1[4], i_data2[4],i_data3[4],lo=15,hi=128;  
    char dir[TAM_ADD], data[BUFSIZE+1],garbage[BUFSIZE+1],info[BUFSIZE+1];  
    char *datos=" ";  
    int i,lo_i,j,k=1,exponente;  
    double voltaje,fraccion;  
  
    config_NOCANON();  /** Configura el puerto para transferencias no canonicas **/  
    c=65;  
    n=30;  
    a=0;  
    CDREG(c,n,a,dir);  /**define direccion del controlador**/  
    CFSA(17,dir,"0,0,0"); /** TRANSFERENCIAS DE 24b **/  
    CFSA(9,"A8,0","0,15,255"); /** RESET al modulo**/  
    CFSA(16,"A8,0","0,0,2"); /** SCAN FAST **/  
  
    pausa(0,280000);  
    /** espera 280 milisegundos, minima espera:170 miliseq en FAST SCAN despues de RESET**/  
    CFSA(17,"A8,0","0,0,0"); /**COMIENZA A LEER VOLTAJE EN CANAL 0**/  
  
    for(;;){ /**ciclo infinito**/  
  
        CFSA(0,"A8,0",data);/** lectura del modulo 8 **/  
        CTCI(data,i_data1);  /**Selecciona datos**/  
        CFSA(0,"A8,0",data);/** lectura del modulo 8 **/  
        CTCI(data,i_data2);  
    }  
    /******* A CONTINUACION SE DECODIFICA EL VOLTAJE LEIDO *****/
```

```

fraccion=0.0;
/** COMENTARIO TEMPORAL. PUEDE PRESENTARSE EL rango Y EL ac
printf("rango= %d\t",i_data2[2]&lo);
printf("ac= %d\t",i_data2[2]>>4);
**/
bit=CHECK_BIT(i_data1,3,8); /**checa el bit 8 de i_data**/
i_data1[2]=hi;
signo=CHECK_BIT(i_data1,3,16);
i_data1[1]=(i_data1[1]<<1)+bit;
exponente=i_data1[1]-128;
for(j=8,k=1;j>0;j--,k++){
    if(CHECK_BIT(i_data1,3,j))
        fraccion=fraccion+1/(pow(2.0,(double)k));
    if(CHECK_BIT(i_data2,3,j+8))
        fraccion=fraccion+1/(pow(2.0,(double)(k+8)));
}
voltaje=pow(2.0,(double)exponente)*fraccion;
if(signo<1)
    voltaje=-voltaje;
printf("voltaje= %f mV.\n",voltaje*1000); /**presenta el voltaje en milivolts**/

/** COMENTARIO TEMPORAL. PUEDE VERIFICARSE i_data, fraccion Y exponente
printf("i_data:(%d,%d,%d)fraccion:%f exponente:%d\n",i_data1[1], i_data1[2], i_data2[1], fraccion,exponente);
**/

/**A CONTINUACION SE EFECTUA EL CONTROL DE TEMPERATURA**/
if(voltaje>0){
    CDINT24(0,0,1,data); /**construye datos en formato CAMAC**/
    CFSA(16,"A1,3,",data); /**ENVIAR VOLTAJE A CANAL 3 +6.6 milivolts**/
}
else{
    CDINT24(0,15,252,data);
    CFSA(16,"A1,3,",data); /**ENVIAR VOLTAJE A CANAL 3 -5.8 milivolts (15,252)**/
}

pausa(0,180000); /** espera 180 milisegundos**/
/**tiempo minimo de espera: 150 milisegundos en FAST SCAN **/
/** PUEDE REQUERIR MAS TIEMPO SI EXISTE UN CAMBIO DE POLARIDAD
*** HASTA 180 MILISEGUNDOS COMO EN ESTE CASO ***/
CDINT24(0,0,0,garbage);
CFSA(17,"A8,0,",garbage); /** COMIENZA A LEER VOLTAJE EN CANAL 0**/
}
restore();
return 0;
}

```

APENDICE 3

Este apéndice contiene el código fuente del programa de adquisición de datos y uno de los archivos que genera. A continuación se presenta el código fuente:

```
/******  
NOMBRE      adqdata.c  
AUTOR       Brenda Eugenia Olivos Fuentes  
FECHA DE CREACION 16/jul/1997  
DESCRIPCION Este es el programa para adquirir datos desde el interferometro.  
             Requiere de una conexion desde el amplificador hasta el canal 1 (A0)  
             del interval counter (N5); involucra tambien al secuenciador de listas:  
             la secuencia se arranca por LAM, envia BREAK en FIFO FULL,  
             entonces se leen los datos desde la WS.  
             Se usan transferencias canonicas de 24bits.  
             Debera compilarse con cmc_lib.o.  
*****/  
  
#include <stdio.h>  
#define BUFSIZE 20  
#define N_DATA 4096  
#define AJUSTE 51  
#define BYTE 3  
main(argc,argv)  
int argc;  
char *argv[];  
  
{  
    FILE *fp,*fpgnu;  
    unsigned char i_data[4];  
    char *basura,info[BUFSIZE+1];  
    char datos[N_DATA][BUFSIZE+1];  
    int i,cuenta[N_DATA],tiempo,equis[5],yes[5];  
    char *titulo[5];  
  
    titulo[0]=" Periodo Vs Numero de Pulsos  ";  
    titulo[1]=" Periodo Vs Tiempo          ";  
    titulo[2]="Inverso del periodo Vs Numero de Pulsos";  
    titulo[3]=" Inverso del periodo Vs Tiempo  ";  
    titulo[4]=" Numero de Pulsos Vs Tiempo    ";  
    equis[0]=equis[2]=1;  
    equis[1]=equis[3]=equis[4]=3;  
    yes[0]=yes[1]=2;  
    yes[2]=yes[3]=4;  
  
    yes[4]=1;  
    basura=" ";
```



```

if(argc!=2){
    fprintf(stderr,"USE: adqdata.exe nomarch \n");
    exit(1);
}
fp=fopen(argv[1],"w");
if(fp==NULL){
    fprintf(stderr,"ERROR AL ABRIR EL ARCHIVO %s!!!\n",argv[1]);
    exit(1);
}

config(): /** configura puerto para transferencia canonica **/

/** PREPARANDO AL CONTROLADOR: *****/
CFSA(17,"A30,0","0,0,128");
/** generando Z, transferencias simples de 24b, STATUS disable**/
CFSA(1,"A30,0","info");
CTCI(info,i_data);
if(CHECK_BIT(i_data,BYTE,4)==0){
    fprintf(stderr,"VERIFIQUE QUE EL SISTEMA ESTE ON-LINE!!\n");
    exit(1);
}
CFSA(16,"A30,1","0,0,32"); /** generar BREAK en LAM **/
CFSA(17,"A30,13","191,255,255"); /** LAM REQUEST en N23 **/
/*****/

/** PREPARANDO AL INTERVAL COUNTER: *****/
CFSA(17,"A5,0","0,0,7");/** reloj a 10MHz**/
CFSA(16,"A5,0","0,0,1");/**numero de pulsos a esperar**/
CFSA(17,"A5,13","0,0,1");/**habilitar envio del LAM por canal 1**/
/*****/

/** PREPARANDO AL SECUENCIADOR DE LISTAS *****/
CFSA(17,"A23,0","0,0,191");
/**ejecucion a 1MHz, secuencia a 500 Hz,no-recycle, si block transfer**/
CFSA(16,"A23,2","0,0,0");/** inicializando puntero NAF**/
CFSA(16,"A23,1","0,74,2");/** cargando comando: leer cuenta del interval**/
CFSA(16,"A23,1","0,110,10"); /** cargando comando: limpiar LAM trigger **/
CFSA(16,"A23,1","0,202,25");/** cargando comando: iniciar cuenta interval**/
CFSA(9,"A23,0","basura");/** inicializando punteros FIFO y borrar datos **/
CFSA(17,"A23,13","0,1,0"); /** seq envia LAM en FIFO read full **/
CFSA(26,"A23,0","basura");/** habilitar la ejecucion de la lista **/
CFSA(10,"A23,0","basura"); /**limpiar LAM trigger **/
/*****/

/** INICIA LA ADQUISICION DE DATOS: *****/
timestart(); /**toma el tiempo del sistema**/
CFSA(25,"A5,0","basura");/** iniciar cuenta del interval **/
espera_NULL();
CFSA(2,"A30,0","basura"); /** leer interrupt request register **/
CFSA(24,"A23,0","basura"); /**abortar secuencia **/
CFSA(10,"A23,0","basura");/** limpiar LAM trigger **/
CFSA(23,"A23,12","255,255,255");/** limpiar LAM STATUS REGISTER **/
for(i=0;i<N_DATA;i++){

```

```

CFSA(0,"A23.0,",datos[i]); /** transfiere los datos a la WS**/
}
for(i=1;i<N_DATA;i++){
CTCI(datos[i],i_data); /** 'decodifica' los datos **/
cuenta[i]=((int)i_data[0]<<16)+((int)i_data[1]<<8)+(int)i_data[2];
}
for(i=1,tiempo=0;i<N_DATA;i++){
tiempo+=cuenta[i]+AJUSTE;
fprintf(fp,"%04d %08d ",i,cuenta[i]+AJUSTE);
fprintf(fp,"%011d %.20f\n",tiempo,(1.0/(double)(cuenta[i]+AJUSTE)));
/** escribe en el archivo los datos:
i,PERIODO INTERF., TIEMPO y 1/PERIODO AJUSTADOS A 5.1 MICROSEC
el tiempo, el periodo y el inverso del periodo estan expresados
en decimas de microsegundo **/
}
timestop("tiempo de adquisicion"); /**presenta en pantalla el tiempo de adquisicion**/
fclose(fp);
fpgnu=fopen("adqdata.gnu","w");
if(fpgnu==NULL){
fprintf(stderr,"ERROR AL ABRIR EL ARCHIVO adqdata.gnu!!!\n");
exit(1);
}
fprintf(fpgnu,"set term x1 l\nset output\nset nogrid\n");
for(i=0;i<5;i++){
fprintf(fpgnu,"set title '%s'\n",titulo[i]);
fprintf(fpgnu,"plot '%s' using %d:%d with lines\n",argv[1],equis[i],yes[i]);
fprintf(fpgnu,"pause -1 'Minimice el icono y pulse ENTER'\n");
}
fclose(fpgnu);
system("gnuplot adqdata.gnu"); /**llama al gnuplot para presentar graficas en pantalla**/
restore();
return 0;
}

```

A continuación se presentan los datos contenidos en el archivo adq15.dat el cual constituye una muestra ejemplo de datos:

1	20819	20819	0.00004803304673615448	30	14588	497140	0.00006854949273375377	59	11956	888173	0.00008364001338240214
2	20480	41299	0.00004882812500000000	31	14616	511756	0.00006841817186644773	60	12174	900347	0.00008214227041235420
3	19384	60683	0.00005158893933140735	32	13886	525542	0.00007201497911565606	61	12066	912413	0.00008287750704458811
4	18956	79639	0.00005275374551592163	33	13083	538725	0.00007643506840938622	62	12168	924581	0.00008218277449046680
5	18421	98060	0.00005428586938819825	34	14952	553677	0.0000688068485821294	63	12255	936836	0.00008159934720522235
6	17956	116016	0.00005569169079973268	35	13813	567490	0.00007239556939115326	64	12521	949357	0.00007986582541330565
7	17494	133510	0.00005716245569999683	36	13954	581444	0.00007166403898523721	65	12852	962209	0.0000780890133813111
8	16601	150111	0.00006023733510029516	37	13954	595398	0.00007166403898523721	66	13059	975259	0.00007662835249042145
9	16388	166499	0.0000610203872589700	38	14006	609404	0.00007139797229758674	67	14147	989406	0.00007068636460026860
10	15734	182233	0.00006355662895640016	39	14321	623725	0.00006982752601075344	68	12522	1001928	0.00007984944737262418
11	15475	197708	0.00006462033541195476	40	14502	638227	0.00006895600606812854	69	13050	1014978	0.00007662835249042145
12	15428	213136	0.00006481721545242416	41	14608	652825	0.00006845564074497937	70	14297	1029275	0.00006994474365251452
13	15094	228230	0.00006625149065853981	42	14633	667468	0.00006833868633044489	71	11994	1041269	0.00008337502084375521
14	15359	243589	0.00006510840549514942	43	14322	681190	0.00006982265046781176	72	13061	1054390	0.00007656381594058648
15	15779	259368	0.00006337537233031244	44	14212	696002	0.00007036307345904869	73	12856	1067186	0.0000778469197261979
16	16073	275441	0.00006221613886642194	45	14135	710137	0.00007074637424831977	74	12532	1079718	0.00007979572294924992
17	16620	292061	0.00006016847172081829	46	13793	723930	0.00007250054375407815	75	12442	1092160	0.00008037293039704228
18	16630	308891	0.00005941770647653000	47	13475	736405	0.00008016032064128256	76	12265	1104425	0.00008153281695882592
19	16943	325834	0.00005902142477719412	48	13266	749671	0.00007538067239559776	77	12338	1116763	0.00008105041335710812
20	16839	342673	0.00005938594928439931	49	12993	762664	0.00007696451935657662	78	12240	1129003	0.000081699346540322875
21	16289	358962	0.00006139112284363682	50	12775	775439	0.00007827788649706458	79	12138	1141141	0.00008238589553468446
22	16170	375132	0.00006184291898577613	51	12870	788309	0.00007770007770007770	80	12210	1153351	0.00008190008190008190
23	15723	390855	0.00006360109393881375	52	12727	801036	0.00007857311238097745	81	12114	1165465	0.00008254911672445105
24	15510	406365	0.00006447453255963895	53	12860	813896	0.00007766049766718506	82	12160	1176225	0.00008323684210526216
25	15583	421948	0.00006417249566835655	54	12745	826641	0.00007846214201647705	83	11981	1189606	0.00008346548702111677
26	15291	437239	0.00006539794650447976	55	12562	839203	0.00007960515841426525	84	12068	1201674	0.00008286377195889956
27	15331	452570	0.00006522931720044354	56	12443	851946	0.00008036647110825364	85	11922	1213596	0.00008387854386847844
28	15286	467856	0.00006541033795629989	57	12318	863884	0.0000811269815329302	86	12024	1225620	0.00008316699933466401
29	14696	482552	0.00006804572672836147	58	12333	876217	0.00008108327252087895	87	12103	1237723	0.00008262414277451872

460	14721	8070648	0.000067920101778731444
461	14734	8099698	0.000067949259310429011
462	14742	8106554	0.000068157013905888370
463	14799	8120442	0.000068748120647293370
464	15161	8125612	0.00006997870967635538
465	15282	8150809	0.000070485464170178151
466	15171	8161906	0.000070693027070420011
467	15232	8181280	0.0000721371794603510
468	15559	8196848	0.000073427442743106083
469	14902	8211902	0.0000736709198448444
470	15572	8227124	0.0000740446896786782
471	14529	8248382	0.0000743786181834262
472	14751	8265884	0.00007480403520701
473	14441	8271059	0.00007499303363853793
474	14325	8281248	0.00007513774969921
475	14415	8290772	0.0000753141754316197
476	14679	8314442	0.00007581245164384494
477	14346	8328098	0.000076242197167606
478	14646	8332966	0.0000764854641475977
479	15408	8358114	0.000076811946398348
480	14710	8375750	0.000077342481026230
481	14641	8382581	0.0000775033451350919
482	14735	8401061	0.0000776650690400411
483	14668	8412622	0.0000778017902873069
484	15232	8435244	0.00007822506287803261
485	14912	8448306	0.0000784562519728970
486	15116	8468364	0.000078976404071631
487	15741	8479401	0.0000792047058252364
488	15824	8495404	0.000079515356002907
489	16225	8511272	0.0000799408032170972
490	16956	8529208	0.000080481985597434
491	16571	8544729	0.00008094924026277085
492	17164	8561204	0.0000814584654094988
493	16546	8578460	0.0000819440540978991
494	16984	8595401	0.000082442078752501
495	16947	8612340	0.00008294074059173187
496	17221	8629681	0.0000834706750100973
497	17601	8647561	0.0000839612426388960
498	17942	8665454	0.0000844324046377847
499	18848	8683370	0.00008488421531351011
500	19659	8701028	0.000085366278721472109
501	20355	8718940	0.00008582520191424923
502	19412	8737422	0.000086261550297323111
503	19881	8756429	0.0000866692260228570
504	19422	8776162	0.00008703515564914478
505	19574	8796546	0.0000873681962588415
506	19615	8817652	0.00008766251187437291
507	19719	8839401	0.0000879251158765982
508	19824	8861806	0.0000881847322862761
509	19260	8885086	0.000088459107998486214
510	18922	8909106	0.0000887399598559115
511	18729	8933874	0.0000890248128438289
512	18522	8958768	0.0000893145844177612818
513	18295	8984841	0.00008960391599087819
514	18131	8951764	0.000089884646290012562
515	17918	8929278	0.000090158126444011415
516	17708	8907596	0.0000904347262638960
517	17501	8886549	0.00009071425290201257
518	17297	8866141	0.0000910002151159000
519	17109	8846308	0.000091292255543638
520	16928	8827030	0.0000915901191505000
521	16750	8808207	0.0000918925118125135
522	16572	8789940	0.000092190548557909
523	16400	8772248	0.00009249429242117582418
524	16230	8755104	0.00009280421650722377
525	16071	8738506	0.000093119842385846
526	15918	8722441	0.000093442910969625401
527	15767	8706904	0.00009377187906602018
528	15618	8691880	0.00009410754521456942
529	15471	8677360	0.00009444578622659106
530	15328	8663340	0.00009478934139877993
531	15188	8649820	0.000095139701255891134
532	15051	8636800	0.000095496019745108598
533	14918	8624280	0.000095863920962148784
534	14788	8612260	0.00009624481981073426
535	14661	8600740	0.0000966398400409086
536	14537	8589720	0.000097049645621209159
537	14417	8579200	0.00009747614056720099
538	14300	8569180	0.00009791662298272874
539	14187	8559560	0.0000983734129258868
540	14079	8550340	0.0000988468018942327
541	13976	8541520	0.000099327604710321
542	13878	8533100	0.000099825728346974
543	13785	8525080	0.0001003419815012612
544	13695	8517460	0.00010087483894497379
545	13610	8510240	0.00010142663779548074
546	13530	8503420	0.00010199518012612
547	13457	8496900	0.0001025933894497379
548	13390	8490780	0.000103221702528679
549	13329	8485060	0.0001038790691896712254
550	13273	8479740	0.0001045548018942327
551	13222	8474820	0.0001052464710321
552	13174	8469300	0.000105953920962148784
553	13131	8464180	0.00010667728346974
554	13091	8459460	0.000107420110321
555	13055	8455140	0.000108182940489230
556	13024	8451220	0.0001089664909446637
557	13000	8447700	0.0001097706685459979
558	12973	8444580	0.0001106061191791
559	12953	8441760	0.0001114733117070605
560	12937	8439240	0.000112379138942327
561	12925	8437020	0.000113320094189230
562	12917	8435100	0.0001143060489230
563	12913	8433580	0.0001153279445327
564	12912	8432460	0.0001163860489230
565	12913	8431740	0.0001174800489230
566	12916	8431320	0.00011861109446637
567	12921	8431000	0.0001197790489230
568	12926	8430780	0.0001209740489230
569	12932	8430660	0.0001222050489230
570	12939	8430640	0.0001234700489230
571	12947	8430720	0.0001247680489230
572	12956	8430800	0.0001260990489230
573	12966	8430980	0.0001274620489230
574	12977	8431260	0.0001288570489230
575	12989	8431640	0.0001302840489230
576	12999	8432120	0.0001317430489230
577	13012	8432700	0.0001332340489230
578	13026	8433380	0.0001347570489230
579	13042	8434160	0.0001363120489230
580	13059	8435040	0.0001378990489230
581	13077	8436020	0.0001395180489230
582	13096	8437100	0.0001411690489230
583	13117	8438280	0.0001428620489230
584	13139	8439560	0.0001445970489230
585	13163	8440940	0.0001463740489230
586	13188	8442420	0.0001481930489230
587	13215	8444000	0.0001500540489230
588	13243	8445680	0.0001519570489230
589	13273	8447460	0.0001539020489230
590	13304	8449340	0.0001558890489230
591	13336	8451320	0.0001579180489230
592	13370	8453400	0.0001599890489230
593	13406	8455580	0.0001621020489230
594	13443	8457860	0.0001642570489230
595	13482	8460240	0.0001664540489230
596	13523	8462720	0.0001686940489230
597	13565	8465300	0.0001709770489230
598	13609	8467980	0.0001733020489230
599	13654	8470760	0.0001756690489230
600	13701	8473640	0.0001780770489230

Table with columns containing numerical values (e.g., 1750, 1751, etc.) and alphanumeric identifiers (e.g., 0.000049197203546607, 0.00004317187725805654). The table is organized into three vertical sections.

Table with 4 columns containing numerical data, likely a financial or statistical dataset. The first column is the most visible, showing values from 2220 down to 2600.

2692	24812	7145298	0.000040507015524746
2693	25157	7147250	0.00003974829672010178
2694	25524	7150278	0.00003936144416178511
2695	25893	7153287	0.00003896746692274626
2696	26264	7156296	0.000038580127909201978
2697	26637	7159305	0.00003820207582643135
2698	27011	7162314	0.00003782402280137135
2699	27386	7165323	0.00003744597007304481
2700	27762	7168332	0.00003706791786749772
2701	28138	7171341	0.00003668986491205482
2702	28514	7174350	0.00003631181203846712
2703	28890	7177359	0.0000359337591205482
2704	29266	7180368	0.0000355557062497413
2705	29642	7183377	0.0000351776472006713
2706	30018	7186386	0.0000347995882306713
2707	30394	7189395	0.0000344215292416713
2708	30770	7192404	0.0000340434702516713
2709	31146	7195413	0.0000336654112616713
2710	31522	7198422	0.0000332873522716713
2711	31898	7201431	0.0000329092932816713
2712	32274	7204440	0.0000325312342916713
2713	32650	7207449	0.0000321531753016713
2714	33026	7210458	0.0000317751163116713
2715	33402	7213467	0.0000313970573216713
2716	33778	7216476	0.0000310189982816713
2717	34154	7219485	0.0000306409392916713
2718	34530	7222494	0.0000302628803016713
2719	34906	7225503	0.0000298848213116713
2720	35282	7228512	0.0000295067623216713
2721	35658	7231521	0.0000291287033316713
2722	36034	7234530	0.0000287506443416713
2723	36410	7237539	0.0000283725853516713
2724	36786	7240548	0.0000279945263616713
2725	37162	7243557	0.0000276164673716713
2726	37538	7246566	0.0000272384083816713
2727	37914	7249575	0.0000268603493916713
2728	38290	7252584	0.0000264822904016713
2729	38666	7255593	0.0000261042314116713
2730	39042	7258602	0.0000257261724216713
2731	39418	7261611	0.0000253481134316713
2732	39794	7264620	0.0000249700544416713
2733	40170	7267629	0.0000245919954516713
2734	40546	7270638	0.0000242139364616713
2735	40922	7273647	0.0000238358774716713
2736	41298	7276656	0.0000234578184816713
2737	41674	7279665	0.0000230797594916713
2738	42050	7282674	0.0000227017005016713
2739	42426	7285683	0.0000223236415116713
2740	42802	7288692	0.0000219455825216713
2741	43178	7291701	0.0000215675235316713
2742	43554	7294710	0.0000211894645416713
2743	43930	7297719	0.0000208114055516713
2744	44306	7300728	0.0000204333465616713
2745	44682	7303737	0.0000200552875716713
2746	45058	7306746	0.0000196772285816713
2747	45434	7309755	0.0000192991695916713
2748	45810	7312764	0.0000189211106016713
2749	46186	7315773	0.0000185430516116713
2750	46562	7318782	0.0000181649926216713
2751	46938	7321791	0.0000177869336316713
2752	47314	7324800	0.0000174088746416713
2753	47690	7327809	0.0000170308156516713
2754	48066	7330818	0.0000166527566616713
2755	48442	7333827	0.0000162746976716713
2756	48818	7336836	0.0000158966386816713
2757	49194	7339845	0.0000155185796916713
2758	49570	7342854	0.0000151405207016713
2759	49946	7345863	0.0000147624617116713
2760	50322	7348872	0.0000143844027216713
2761	50698	7351881	0.0000140063437316713
2762	51074	7354890	0.0000136282847416713
2763	51450	7357899	0.0000132502257516713
2764	51826	7360908	0.0000128721667616713
2765	52202	7363917	0.0000124941077716713
2766	52578	7366926	0.0000121160487816713
2767	52954	7369935	0.0000117379897916713
2768	53330	7372944	0.0000113599308016713
2769	53706	7375953	0.0000109818718116713
2770	54082	7378962	0.0000106038128216713
2771	54458	7381971	0.0000102257538316713
2772	54834	7384980	0.0000098476948416713
2773	55210	7387989	0.0000094696358516713
2774	55586	7390998	0.0000090915768616713
2775	55962	7394007	0.0000087135178716713
2776	56338	7397016	0.0000083354588816713
2777	56714	7400025	0.0000079573998916713
2778	57090	7403034	0.0000075793409016713
2779	57466	7406043	0.0000072012819116713
2780	57842	7409052	0.0000068232229216713
2781	58218	7412061	0.0000064451639316713
2782	58594	7415070	0.0000060671049416713
2783	58970	7418079	0.0000056890459516713
2784	59346	7421088	0.0000053109869616713
2785	59722	7424097	0.0000049329279716713
2786	60098	7427106	0.0000045548689816713
2787	60474	7430115	0.0000041768099916713
2788	60850	7433124	0.0000037987510016713
2789	61226	7436133	0.0000034206920116713
2790	61602	7439142	0.0000030426330216713
2791	61978	7442151	0.0000026645740316713
2792	62354	7445160	0.0000022865150416713
2793	62730	7448169	0.0000019084560516713
2794	63106	7451178	0.0000015303970616713
2795	63482	7454187	0.0000011523380716713
2796	63858	7457196	0.0000007742790816713
2797	64234	7460205	0.0000003962200916713
2798	64610	7463214	0.0000000160501016713
2799	64986	7466223	0.000000000001116713
2800	65362	7469232	0.0000000000001216713

APENDICE 4

Este apéndice contiene el código fuente del programa de proceso de datos y uno de los archivos que genera. A continuación se presenta el código fuente:

```

/*****
NOMBRE      prodata.c
AUTOR      Brenda Eugenia Olivos Fuentes
DESCRIPCION Este es el programa que procesa los datos arrojados por el programa
            de adquisicion. Necesita que el archivo de entrada contenga los datos
            del tipo requerido y en el orden en que los arroja el programa de
            adquisicion. Su objetivo es calcular los numeros k (amplitud del
            oscilador) y los numeros T/2 (semiperiodo del oscilador).
*****/

#include <stdio.h>
#define N_DATA 4096
void suavexp(cuenta,alfa,promedio)
double cuenta[N_DATA],promedio[N_DATA];
int alfa;
{
    /*** SUAVIZACION EXPONENCIAL ***/
    int i;
    float alfa2;
    double temp[N_DATA];
    if(alfa<=0||alfa>=100)
        printf("CONSTANTE DE SUAVIZACION INVALIDA!!!");
    else{
        alfa2=(float)alfa/100;
        temp[1]=cuenta[1];
        for(i=2;i<=N_DATA;i++)
            temp[i]=alfa2*cuenta[i-1]+(1-alfa2)*temp[i-1];
        for(i=1;i<=N_DATA;i++)
            promedio[i]=temp[i];
    }
}

int max_min(cuenta,numk,error)
double cuenta[N_DATA];
int numk[N_DATA],error;
{
    /*** ENCUENTRA EL NUMERO DE PULSOS ENTRE DOS MAXIMOS ***/

    int i,signo,flag,contador, contador2,contador3=0;
    for(i=1,contador=0,contador2=0,signo=1,flag=0;i<N_DATA;i++){
        if((signo*(cuenta[i]-cuenta[i+1]))<=error){
            if(flag==1)
                ++contador;
            else
                ++contador3;
        }
        else{
            flag=1;
            if(signo==1){
                numk[contador2]=contador;
                ++contador2;
                contador=1;
            }
        }
    }
}

```

```

    }
    else
        ++contador;

    signo=-1*(signo);
}
}
numk[0]=contador3; /** numero de pulsos antes del primer maximo **/
return contador2;
}

main(argc,argv)
int argc;
char *argv[];

{
FILE *fpin, *fpout, *fpgnu;
int i,j,k,flag,contador, numk[N_DATA],temp;
int error=100,alfa=10,equis[4],yes[4];
double promedio[N_DATA],tiempo,numT[N_DATA];
double cuenta[N_DATA];
char *titulo[6];

titulo[2]="    Periodo suavizado Vs pulsos    ";
titulo[3]="    Periodo suavizado Vs tiempo    ";
titulo[4]="        Numero de pulsos            ";
titulo[5]="        Tiempo entre pulsos            ";

equis[0]=equis[2]=equis[3]=1;
equis[1]=3;
yes[0]=yes[1]=2;
yes[2]=4;
yes[3]=5;

if(argc!=3){
    fprintf(stderr,"USE: procdat.exe nomarch nomarch \n");
    exit(1);
}
fpin=fopen(argv[1],"r");
if(fpin==NULL){
    fprintf(stderr,"ERROR AL ABRIR EL ARCHIVO %s!!!!\n",argv[1]);
    exit(1);
}
fpout=fopen(argv[2],"w");
if(fpout==NULL){
    fprintf(stderr,"ERROR AL ABRIR EL ARCHIVO %s!!!!\n",argv[2]);
    exit(1);
}
}
/*****
                FORMATO ESPERADO DEL ARCHIVO:
                i   periodo T   t acumulado       f=1/T
                (int) (int)   (int)   (double)
*****/

for(i=1;flag!=EOF;i++){
    /** lee archivo de datos **/
    flag=fscanf(fpin,"%*s %d %*s %*s",&temp);
    cuenta[i]=(double)temp;
}
/** doble suavizacion exponencial **/
suavexp(cuenta,alfa,promedio);
suavexp(promedio,alfa,promedio);

/** calculo del numero de pulsos numk entre maximos **/
contador=max_min(promedio,numk,error);

```

```

/** calculo del tiempo numT entre maximos **/
for(i=0.k=1;i<contador;i++){
  for(j=1.tiempo=0.0;j<=numk[i];j++){
    tiempo=tiempo+cuenta[k];
    ++k;
  }
  numT[i]=tiempo;
}

/** envia resultados al archivo de salida ***/
for(i=1.tiempo=0.0;i<N_DATA;i++){
  tiempo=tiempo+promedio[i];
  fprintf(fpout,"%4d %12.3f %15.3f",i.promedio[i].tiempo);
  if(i<contador)
    fprintf(fpout,"%8d %11d\n".numk[i].(int)numT[i]);
  else
    fprintf(fpout,"%8s %11s\n"."#"."#");
}

fclose(fpin);
fclose(fpout);

fpgnu=fopen("procddata.gnu","w");
if(fpgnu==NULL){
  fprintf(stderr,"ERROR AL ABRIR EL ARCHIVO procddata.gnu!!!\n");
  exit(1);
}
fprintf(fpgnu,"set term x11\nset output\nset nogrid\n");
for(i=0;i<4;i++){
  fprintf(fpgnu,"set title '%s'\n",titulo[i+2]);
  fprintf(fpgnu,"plot '%s' using '%d:%d'",argv[2],equis[i].yes[i]);
  if(i<2)
    fprintf(fpgnu,"with lines\n");
  else
    fprintf(fpgnu,"with points\n");
  fprintf(fpgnu,"pause -1 'Minimice el icono y pulse ENTER'\n");
}
fclose(fpgnu);
system("gnuplot procddata.gnu"); /**llama al gnuplot para presentar las graficas en pantalla**/
}

```

A continuación se presentan los datos contenidos en el archivo proc15.dat, el cual constituye el proceso de la muestra ejemplo de datos:

1	20819 000	20819 000	267	5066048	22	18393 971	439315 224	82	5918703	43	15676 726	795392 786	#	#
2	20819 000	41638 000	284	5824703	23	18259 384	457574 608	77	5728125	44	15582 700	810975 486	#	#
3	20819 000	62457 000	268	5818416	24	18129 475	475704 083	71	5829306	45	15493 932	826469 419	#	#
4	20819 610	83272 610	253	5827649	25	18000 184	493704 267	69	5844784	46	15409 211	841878 630	#	#
5	20798 548	104071 158	238	5835993	26	17870 556	511574 823	63	5699625	47	15327 845	857206 474	#	#
6	20766 302	124837 460	221	5838856	27	17742 682	529317 505	59	5954026	48	15246 589	872453 064	#	#
7	20716 730	145554 191	210	5865083	28	17614 588	546932 093	41	3294161	49	15153 057	887606 120	#	#
8	20648 970	166203 160	194	5714903	29	17487 995	564420 088	#	#	50	15058 425	902664 545	#	#
9	20562 534	186765 694	183	5854519	30	17362 435	581783 523	#	#	51	14961 119	917625 664	#	#
10	20483 905	207218 599	172	5761906	31	17235 867	599019 390	#	#	52	14860 439	932486 103	#	#
11	20325 457	227542 057	161	5800012	32	17106 058	616125 449	#	#	53	14758 985	947245 088	#	#
12	20172 710	247714 767	152	5780268	33	16976 013	633101 462	#	#	54	14656 487	961901 575	#	#
13	20003 627	267718 394	144	5817967	34	16839 776	649941 237	#	#	55	14555 499	976457 073	#	#
14	19820 914	287539 308	135	5722679	35	16691 836	666633 093	#	#	56	14455 593	990912 666	#	#
15	19625 647	307164 956	127	5770343	36	16554 642	683187 736	#	#	57	14355 734	1005268 400	#	#
16	19424 815	326589 770	122	5876824	37	16416 083	699603 819	#	#	58	14255 720	1019524 120	#	#
17	19225 682	345815 453	111	5771399	38	16279 229	715883 048	#	#	59	14154 532	1033678 652	#	#
18	19032 858	364848 311	106	5699688	39	16145 125	732028 175	#	#	60	14054 355	1047733 007	#	#
19	18852 542	383700 853	90	5914641	40	16015 110	748043 282	#	#	61	13953 227	1061685 234	#	#
20	18686 260	402387 113	93	5768793	41	15892 856	763916 138	#	#	62	13851 722	1075396 956	#	#
21	18534 140	420921 253	84	5600132	42	15779 922	779716 060	#	#	63	13752 455	1089289 410	#	#

64	13056 204	1102945 615	*	*	188	13765 229	2657005 790	*	*	312	25692 291	6023251 559	*	*
65	13564 229	1116509 844	*	*	189	13797 497	2670803 827	*	*	313	25190 010	6047441 570	*	*
66	13479 297	1129989 141	*	*	190	13820 590	2684623 877	*	*	314	24704 243	6072145 812	*	*
67	13404 229	1139309 378	*	*	191	13834 119	2698457 997	*	*	315	24227 968	6096373 781	*	*
68	13339 882	1156731 253	*	*	192	13840 087	2712208 083	*	*	316	23764 277	6120188 057	*	*
69	13295 832	1170029 085	*	*	193	13838 969	2726137 052	*	*	317	23313 323	6143541 381	*	*
70	13252 413	1183281 498	*	*	194	13831 065	2739968 117	*	*	318	22870 308	6166321 689	*	*
71	13215 230	1196496 717	*	*	195	13817 631	2753785 748	*	*	319	22437 113	6188758 801	*	*
72	13195 911	1209692 628	*	*	196	13800 574	2767656 522	*	*	320	22000 833	6210765 833	*	*
73	13168 251	1222860 879	*	*	197	13777 941	2781364 263	*	*	321	21581 119	6233250 753	*	*
74	13144 775	1236005 654	*	*	198	13757 590	2795121 853	*	*	322	21171 089	6253521 842	*	*
75	13122 871	1249128 525	*	*	199	13739 109	2808860 962	*	*	323	20769 173	6274291 015	*	*
76	13099 220	1262227 745	*	*	200	13725 299	2822596 261	*	*	324	20381 350	6294672 368	*	*
77	13073 491	1275301 226	*	*	201	13740 349	2836336 011	*	*	325	20010 660	6314883 025	*	*
78	13044 565	1288345 801	*	*	202	13757 587	2850094 197	*	*	326	19658 824	6334341 819	*	*
79	13014 070	1301359 871	*	*	203	13790 713	2863884 911	*	*	327	19329 569	6353761 470	*	*
80	12981 623	1314341 492	*	*	204	13847 039	2877731 950	*	*	328	19025 347	6372696 766	*	*
81	12946 914	1327288 411	*	*	205	13926 502	2891658 452	*	*	329	18745 243	6391442 009	*	*
82	12911 426	1340199 838	*	*	206	14023 712	2905682 163	*	*	330	18489 207	6409931 716	*	*
83	12874 706	1353074 545	*	*	207	14140 725	2919822 889	*	*	331	18256 605	6428187 821	*	*
84	12837 817	1365912 361	*	*	208	14273 348	2934096 237	*	*	332	18038 332	6446226 154	*	*
85	12799 368	1378711 729	*	*	209	14414 770	2948511 007	*	*	333	17834 468	6464060 621	*	*
86	12760 911	1391472 640	*	*	210	14559 433	2963070 440	*	*	334	17639 022	6481699 644	*	*
87	12721 371	1404194 011	*	*	211	14703 237	2977773 677	*	*	335	17455 472	6499155 145	*	*
88	12682 371	1416876 382	*	*	212	14842 375	2992616 052	*	*	336	17284 641	6516439 756	*	*
89	12644 986	1429521 368	*	*	213	14976 143	3007592 195	*	*	337	17123 382	6533633 538	*	*
90	12607 374	1442200 628	*	*	214	15105 804	3022697 999	*	*	338	16972 028	6550553 166	*	*
91	12574 956	1454704 700	*	*	215	15236 261	3037924 260	*	*	339	16828 609	6567433 666	*	*
92	12544 728	1467249 418	*	*	216	15338 229	3053262 489	*	*	340	16690 069	6584303 835	*	*
93	12520 205	1479769 633	*	*	217	15439 181	3068701 679	*	*	341	16542 297	6600596 132	*	*
94	12501 560	1492371 193	*	*	218	15530 910	3084232 570	*	*	342	16408 040	6617004 172	*	*
95	12489 541	1504760 539	*	*	219	15616 231	3099854 310	*	*	343	16263 090	6633468 252	*	*
96	12484 591	1517245 125	*	*	220	15702 609	3115554 919	*	*	344	16119 951	6649387 351	*	*
97	12484 518	1529729 643	*	*	221	15785 964	3131340 883	*	*	345	15980 138	6665367 215	*	*
98	12488 303	1542217 846	*	*	222	15875 132	3147216 015	*	*	346	15839 028	6681206 378	*	*
99	12493 396	1554711 642	*	*	223	15975 157	3163191 172	*	*	347	15701 329	6696907 208	*	*
100	12499 138	1567210 780	*	*	224	16091 145	3179269 317	*	*	348	15578 939	6712864 647	*	*
101	12501 624	1579712 404	*	*	225	16227 404	3195509 722	*	*	349	15449 834	6727936 482	*	*
102	12498 761	1592211 165	*	*	226	16376 320	3211886 042	*	*	350	15340 221	6743276 703	*	*
103	12491 445	1604702 610	*	*	227	16531 809	3228474 851	*	*	351	15237 782	6758514 485	*	*
104	12478 485	1617178 485	*	*	228	16691 717	3245409 598	*	*	352	15131 489	6773645 972	*	*
105	12451 633	1629630 118	*	*	229	16849 619	3262958 617	*	*	353	15034 566	6788676 539	*	*
106	12420 072	1642050 190	*	*	230	17001 739	3278960 156	*	*	354	14952 403	6803632 942	*	*
107	12382 136	1654432 326	*	*	231	17145 675	3296106 031	*	*	355	14862 727	6818495 669	*	*
108	12337 497	1666769 923	*	*	232	17286 197	3313392 228	*	*	356	14781 582	6833277 251	*	*
109	12287 244	1679057 168	*	*	233	17424 197	3330924 423	*	*	357	14715 459	6847911 010	*	*
110	12233 426	1691290 994	*	*	234	17565 525	3348381 958	*	*	358	14638 785	6862629 495	*	*
111	12176 649	1703467 243	*	*	235	17712 166	3366094 916	*	*	359	14569 121	6877198 616	*	*
112	12118 283	1715585 527	*	*	236	17861 833	3383955 949	*	*	360	14512 142	6891710 257	*	*
113	12059 134	1727644 661	*	*	237	18020 325	3401976 275	*	*	361	14445 997	6906156 574	*	*
114	12001 673	1739964 313	*	*	238	18184 841	3420120 116	*	*	362	14381 833	6920861 824	*	*
115	11945 101	1752191 434	*	*	239	18390 180	3438560 196	*	*	363	14325 048	6935860 073	*	*
116	11891 718	1764383 152	*	*	240	18621 992	3457182 387	*	*	364	14270 859	6949126 931	*	*
117	11840 920	1776524 071	*	*	241	18888 179	3476070 567	*	*	365	14223 086	6962330 017	*	*
118	11795 444	1788519 610	*	*	242	19174 580	3495245 529	*	*	366	14186 370	6975556 287	*	*
119	11756 935	1800376 550	*	*	243	19479 252	3514704 370	*	*	367	14141 845	6988806 252	*	*
120	11724 381	1810600 932	*	*	244	19729 550	3534433 129	*	*	368	14097 982	7002796 315	*	*
121	11700 980	1822201 911	*	*	245	19977 277	3554411 206	*	*	369	14050 664	7016446 378	*	*
122	11689 694	1833991 606	*	*	246	20199 562	3574610 768	*	*	370	14000 959	7030347 330	*	*
123	11681 686	1845883 292	*	*	247	20403 188	3595011 085	*	*	371	13949 593	7044706 937	*	*
124	11717 063	1857400 355	*	*	248	20579 916	3615591 031	*	*	372	13899 644	7059696 571	*	*
125	11747 337	1869147 693	*	*	249	20746 032	3636337 034	*	*	373	13851 384	7075347 955	*	*
126	11800 666	1880948 359	*	*	250	21005 456	3657242 489	*	*	374	13804 431	7090952 166	*	*
127	11850 466	1892798 825	*	*	251	21265 774	3678308 264	*	*	375	13760 225	7107112 311	*	*
128	11920 679	1904698 306	*	*	252	21540 065	369948 328	*	*	376	13717 256	7116829 848	*	*
129	11983 235	1916502 740	*	*	253	21835 369	3720909 697	*	*	377	13673 813	7127100 681	*	*
130	12046 583	1928749 323	*	*	254	21677 612	3742661 309	*	*	378	13642 998	7141150 679	*	*
131	12108 605	1940857 982	*	*	255	22001 923	3764663 232	*	*	379	13612 541	7155763 220	*	*
132	12164 644	1953022 626	*	*	256	22411 455	3787074 687	*	*	380	13586 756	7171349 976	*	*
133	12212 645	1965315 581	*	*	257	22844 882	3809949 368	*	*	381	13565 793	7187941 645	*	*
134	12252 678	1977488 259	*	*	258	23379 539	3833328 907	*	*	382	13544 685	7198460 544	*	*
135	12283 197	1989771 456	*	*	259	23937 377	3857266 284	*	*	383	13523 940	7211984 394	*	*
136	12302 485	2002073 940	*	*	260	24571 613	3881837 897	*	*	384	13499 937	7225484 331	*	*
137	12311 693	2014385 633	*	*	261	25284 347	3907122 244	*	*	385	13474 206	7238958 257	*	*
138	12323 065	2026697 698	*	*	262	26049 739	39326 083	*	*	386	13447 630	7252106 168	*	*
139	12304 646	2039002 344	*	*	263	27148 589	3960375 572	*	*	387	13421 350	7265827 518	*	*
140	12289 310	2051291 654	*	*	264	28354 191	3988729 763	*	*	388	13396 068	7279223 586	*	*
141	12268 834	206360 880	*	*	265	29633 757	4018363 520	*	*	389	13370 919	7292594 506	*	*
142	12241 961	2076003 389	*	*	266	31212 058	4049048 228	*	*	390	13347 140	7305941 645	*	*
143	12209 826	2088012 214	*	*	267	36471 322	4086316 900	*	*	391	13323 297	7319264 942	*	*
144	12172 907	2100185 121	*	*	268	41524 622	4127841 522	*	*	392	13295 731	7332560 672	*	*
145	12132 843	2112317 964	*	*	269	45694 819	4173536 341	*	*	393	13266 845	7345827 517	*	*
146	12092 823	2124410 787	*	*	270	49181 830	4222718 170	*	*	394	13233 579	7359061 197	*	*
147	12051 599	2136462 296	*	*	271	51949 881	4274668 151	*	*	395	13198 538	7372768 315	*	*
148	12013 289	2148475 580	*	*	272	54023 333	4328720 484	*	*	396	13160 520	7385420 105	*	*
149	11980 058	2160455 644	*	*	273	55569 845	4384290 329	*	*	397	13122 569	7398542 724	*	*
150	11952 991	2172408 635	*	*	274	56554 421	4440844 750							

436	13054 448	7896925 730	560	58654 010	10355228 579	684	13676 204	13336643 703
437	13085 616	7929116 646	561	58529 472	10473759 451	685	13664 988	13350308 691
438	13112 945	7923114 292	562	58233 212	10471992 663	686	13650 003	13363358 684
439	13138 063	7936262 355	563	57747 625	10529740 289	687	13630 446	13375589 140
440	13160 367	7949422 722	564	57093 384	10586833 672	688	13605 320	13391194 459
441	13181 270	7966061 292	565	56294 234	10643127 906	689	13576 714	13404771 173
442	13200 900	7975804 901	566	55830 131	10698508 337	690	13545 867	13418145 040
443	13222 599	7989027 591	567	54383 366	10752891 402	691	13508 452	13431823 452
444	13242 859	8002270 450	568	53312 282	10806203 685	692	13473 801	13445297 293
445	13261 332	8015531 782	569	52181 302	10858384 986	693	13440 896	13458738 189
446	13289 761	8028811 143	570	51019 205	10910544 191	694	13419 804	13472157 992
447	13302 421	8042123 546	571	49845 524	10962749 715	695	13405 761	13485562 753
448	13324 865	8055448 429	572	48682 727	11010922 442	696	13385 769	13498940 522
449	13342 176	8068780 605	573	47539 615	11055472 057	697	13384 278	13512333 800
450	13351 587	8082132 192	574	46397 117	11101869 174	698	13390 827	13525724 627
451	13357 433	8095480 625	575	45244 833	11147114 057	699	13405 003	13539129 630
452	13374 325	8108865 950	576	44084 255	11191198 312	700	13427 466	13552557 096
453	13390 464	8122254 414	577	42917 363	11234115 676	701	13460 797	13566017 893
454	13405 691	8135660 005	578	41748 818	11275864 494	702	13501 996	13579521 300
455	13421 809	8149081 814	579	40589 407	11316453 901	703	13556 508	13593078 398
456	13439 097	8162520 911	580	39452 231	11355906 132	704	13621 068	13606699 466
457	13461 139	8175982 051	581	38344 636	11394250 768	705	13693 300	13620292 766
458	13480 152	8189471 203	582	37274 587	11431525 355	706	13769 665	13634162 431
459	13500 191	8202994 394	583	36242 532	11469767 887	707	13845 395	13648007 826
460	13528 261	8216546 646	584	35240 522	11507928 410	708	13916 301	13661824 127
461	13598 764	8230149 420	585	34331 839	11547360 249	709	13997 633	13675901 760
462	13644 595	8245794 016	586	33463 497	11570823 746	710	14028 585	13689930 315
463	13687 812	8252481 828	587	32662 795	11603486 541	711	14069 336	13703999 652
464	13732 056	8257214 457	588	31920 669	11635407 210	712	14097 076	13718096 727
465	13779 629	8264901 117	589	31251 230	11666630 459	713	14104 714	13732201 572
466	13831 480	8298825 605	590	30559 142	11697189 581	714	14113 454	13746314 896
467	13888 000	8312713 605	591	29913 849	11727103 450	715	14113 279	13760428 175
468	13945 694	8326659 208	592	29278 623	11756382 053	716	14109 404	13774437 579
469	14006 037	8340662 845	593	28659 144	11785028 197	717	14103 741	13788650 760
470	14070 917	8354743 763	594	28041 665	11813458 862	718	14093 743	13802903 433
471	14133 061	8368868 824	595	27396 570	11840442 432	719	14085 797	13816819 330
472	14196 131	8383064 956	596	26785 757	11867228 190	720	14081 322	13830900 462
473	14250 547	8397318 502	597	26186 702	11893414 891	721	14078 282	13844978 745
474	14309 428	8411620 030	598	25595 359	11919028 640	722	14081 312	13859070 249
475	14340 537	8425955 468	599	25020 009	11944044 259	723	14089 582	13873149 637
476	14373 661	8440329 128	600	24479 564	11968523 823	724	14103 495	13887253 132
477	14400 904	8454730 032	601	23953 579	11992477 402	725	14124 539	13901377 671
478	14425 751	8469154 784	602	23450 555	12016528 957	726	14152 710	13915530 382
479	14447 081	8483603 895	603	22975 170	12039893 127	727	14190 941	13929713 322
480	14465 967	8498068 532	604	22526 876	12064300 003	728	14240 249	13943961 572
481	14486 685	8512555 517	605	22106 170	12088336 173	729	14297 996	13958259 668
482	14505 760	8527061 376	606	21716 986	12105253 159	730	14359 431	13972618 999
483	14525 063	8541486 679	607	21349 237	12126612 896	731	14422 019	13987041 019
484	14543 877	8556130 816	608	21025 988	12144388 884	732	14480 555	1400151 919
485	14562 252	8570693 068	609	20716 301	12168355 185	733	14536 364	14016057 938
486	14584 834	8585277 902	610	20421 172	12188776 357	734	14583 926	14030641 864
487	14607 796	8599885 698	611	20140 216	12210896 573	735	14624 421	14045266 284
488	14631 898	8614517 596	612	19869 900	12234628 572	736	14657 468	14059962 753
489	14662 611	8629180 208	613	19605 114	12258181 686	737	14688 362	14074607 115
490	14699 203	8643879 411	614	19344 835	12282736 521	738	14703 852	14089310 966
491	14744 080	8658623 491	615	19089 071	12308625 592	739	14723 310	14104034 272
492	14802 550	8673425 841	616	18838 632	12335664 224	740	14744 749	14119279 025
493	14866 835	8688292 677	617	18596 169	12362860 794	741	14771 976	14133534 001
494	14942 040	8703223 716	618	18364 623	12390242 417	742	14808 051	14148339 052
495	15018 595	8718253 311	619	18143 951	12417869 368	743	14848 640	14163207 692
496	15100 359	8733255 670	620	17935 986	12445805 354	744	14898 462	14178106 154
497	15185 054	8748338 724	621	17744 695	12474050 049	745	14952 193	14193058 346
498	15275 116	8763481 841	622	17565 513	12502612 562	746	15011 902	14208070 249
499	15374 729	8779188 566	623	17397 700	12531413 262	747	15079 928	14223150 177
500	15480 782	8794669 349	624	17242 144	12560855 406	748	15158 670	14238308 848
501	15600 330	8810269 679	625	17096 283	12645751 689	749	15251 314	14253560 162
502	15737 751	8826007 430	626	16961 662	12682713 351	750	15352 773	14268912 935
503	15895 214	8841902 645	627	16846 437	12719655 894	751	15461 187	14284374 123
504	16065 177	8857965 820	628	16739 631	12756295 524	752	15570 970	14299945 892
505	16257 405	8874203 225	629	16646 716	12793294 240	753	15675 745	14315620 037
506	16410 886	8890614 111	630	16561 017	12830903 257	754	15773 035	14331393 872
507	16581 036	8907195 147	631	16483 521	12869596 578	755	15856 310	14347250 181
508	16749 218	8924134 365	632	16406 965	12907500 543	756	15929 059	14363172 240
509	16911 143	8940585 508	633	16332 216	12945825 759	757	15973 385	14379145 620
510	17071 531	8957927 038	634	16253 647	12984970 406	758	16011 846	14395157 471
511	17223 329	8975150 368	635	16170 320	13023149 926	759	16043 330	14411200 803
512	17369 275	8992316 641	636	16087 583	13061570 508	760	16067 580	14427268 526
513	17487 275	9010000 911	637	15977 366	13099204 874	761	16089 506	14443357 887
514	17596 163	9027547	638	15871 878	13137076 752	762	16114 191	14459472 078
515	17695 032	9045292 1	639	15762 044	13175138 796	763	16142 784	14475614 861
516	17781 484	9063076 669	640	15640 667	13213479 463	764	16177 116	14491791 978
517	17863 899	9080923 060	641	15516 416	13252000 979	765	16219 634	14507936 848
518	17940 080	9098857 579	642	15390 358	13290841 437	766	16273 278	14524282 890
519	18055 469	9116943 048	643	15290 776	13329670 213	767	16343 876	14540626 765
520	18178 409	9135121 457	644	15184 886	13368499 099	768	16436 242	14557063 607
521	18320 797	9153441 854	645	15073 247	13407867 346	769	16535 146	14573616 153
522	18480 981	9171922 837	646	14976 866	13447841 212	770	16697 177	14590315 330
523	18671 258	9190591 074	647	14872 610	13488616 822	771	16865 920	14607179 250
524	18878 052	9209472 126	648	14786 536	13530603 358	772	17057 672	14624236 922
525	19089 751	9228561 877	649	14708 650	13574831 008	773	17255 835	14641490 577
526	19323 309	9247885 186	650	14637 337	13620500 344	774	17443 390	14658933 967
527	19571 739	9267450 925	651	14573 989	13667523 333	775	17621 280	14676955 346
528	19831 500	9287288 434	652	14519 118	13714902 251	776	17786 124	14694341 371
529	20095 378	9307383 812	653	14472 123	13764514 374	777	17958 327	14712279 697
530	20354 618	9327738 450	654	14432 956	13818947 330	778	18070 578	14730359 275
531	20694 356	9348347 786	655	14402 861	13875350 971	779	18210 715	14748578 990
532	20865 361	9369211 147	656	14380 525	13933750 716	780	18362 779	14766941 770
533	21125 701	9390336 648	657	14362 658	13994093 375	781	18508 994	14785450 763
534	21405 489	9411742 337	658	14344 870	14056438 244	782	18656 237	14804107 001
535	21692 823	9433435 160	659	14324 432	14120976 676	783	18800 762	14822907 763
536	21994 415	9456420 975	660	14288 133	14188050 899	784	18947 050	14841854 812
537	22303 750	9477323 325	661	14257 440	14259308 249	785	19100 872	14860955 684
538	2							

808	25155 813	15358593 548	*	*	*
809	35620 065	15384215 610	*	*	*
810	26149 785	15410375 395	*	*	*
811	26821 722	15437195 117	*	*	*
812	27551 684	15464726 811	*	*	*
813	28321 154	15492067 965	*	*	*
814	29217 155	15522272 120	*	*	*
815	30488 054	15547743 174	*	*	*
816	32422 043	15585165 176	*	*	*
817	34607 922	15619773 139	*	*	*
818	38187 496	15657960 634	*	*	*
819	42097 865	15700054 500	*	*	*
820	45805 326	15745859 826	*	*	*
821	49715 096	15794894 922	*	*	*
822	51710 339	15844605 261	*	*	*
823	53882 752	15900488 013	*	*	*
824	56221 290	15956112 312	*	*	*
825	59908 809	16012021 121	*	*	*
826	57739 694	16070760 816	*	*	*
827	58166 643	16138971 470	*	*	*
828	58258 096	16218285 566	*	*	*
829	58073 782	16245259 348	*	*	*
830	57673 470	16320293 817	*	*	*
831	57104 803	16390007 620	*	*	*
832	56402 694	16418440 314	*	*	*
833	56347 769	16472075 083	*	*	*
834	54383 832	16526913 815	*	*	*
835	53078 045	16588911 660	*	*	*
836	53052 937	16633945 897	*	*	*
837	52064 870	16686010 767	*	*	*
838	51024 387	16737035 154	*	*	*
839	49847 902	16789893 056	*	*	*
840	48837 860	16837003 916	*	*	*
841	47694 588	16885515 504	*	*	*
842	46531 891	16930047 395	*	*	*
843	45358 528	16975405 923	*	*	*
844	44191 298	17019597 721	*	*	*
845	43046 989	17063614 310	*	*	*
846	41924 928	17104569 239	*	*	*
847	40822 790	17145339 029	*	*	*
848	39758 559	17189155 288	*	*	*
849	38717 197	17232315 485	*	*	*
850	37696 095	17276168 380	*	*	*
851	36693 001	17320826 241	*	*	*
852	35716 613	17365378 874	*	*	*
853	34775 274	17410874 148	*	*	*
854	33871 451	17457262 509	*	*	*
855	33013 649	17504562 248	*	*	*
856	32191 873	17552814 121	*	*	*
857	31409 066	17602044 087	*	*	*
858	30670 542	17650216 629	*	*	*
859	29968 642	17698383 721	*	*	*
860	29306 468	17746591 730	*	*	*
861	28684 001	17794843 740	*	*	*
862	28100 934	17843147 674	*	*	*
863	27549 620	17891512 304	*	*	*
864	27026 057	17940050 361	*	*	*
865	26519 233	17988769 593	*	*	*
866	26023 142	18037692 736	*	*	*
867	25533 338	18086826 074	*	*	*
868	25049 805	18136169 577	*	*	*
869	24573 472	18185729 348	*	*	*
870	24106 999	18235506 347	*	*	*
871	23654 155	18285400 502	*	*	*
872	23211 601	18335416 103	*	*	*
873	22780 466	18385552 620	*	*	*
874	22356 961	18435819 530	*	*	*
875	22003 053	18486206 584	*	*	*
876	21643 817	18536719 401	*	*	*
877	21303 878	18587353 279	*	*	*
878	20985 758	18638110 078	*	*	*
879	20689 714	18688992 752	*	*	*
880	20420 361	18740000 113	*	*	*
881	20178 651	18791134 624	*	*	*
882	19953 140	18842401 904	*	*	*
883	19770 184	18893789 088	*	*	*
884	19599 178	18945298 266	*	*	*
885	19438 812	18996927 078	*	*	*
886	19286 807	19048684 884	*	*	*
887	19142 096	19101579 398	*	*	*
888	18979 041	19154522 250	*	*	*
889	18823 261	19207455 510	*	*	*
890	18662 260	19260388 776	*	*	*
891	18504 018	19313322 044	*	*	*
892	18346 626	19366255 316	*	*	*
893	18190 923	19419188 588	*	*	*
894	18041 452	19472121 860	*	*	*
895	17898 828	19525054 132	*	*	*
896	17760 375	19577987 404	*	*	*
897	17624 445	19630920 676	*	*	*
898	17486 560	19683853 948	*	*	*
899	17355 548	19736786 220	*	*	*
900	17229 833	19789719 492	*	*	*
901	17105 740	19842652 764	*	*	*
902	16989 818	19895585 036	*	*	*
903	16879 508	19948518 308	*	*	*
904	16775 275	20001451 580	*	*	*
905	16680 123	20054384 852	*	*	*
906	16585 438	20107317 124	*	*	*
907	16494 709	20160250 396	*	*	*
908	16399 311	20213183 668	*	*	*
909	16290 466	20266116 940	*	*	*
910	16191 666	20319049 212	*	*	*
911	16092 762	20371982 484	*	*	*
912	15991 742	20424915 756	*	*	*
913	15884 079	20477848 028	*	*	*
914	15787 450	20530781 300	*	*	*
915	15693 367	20583714 572	*	*	*
916	15603 856	20636647 844	*	*	*
917	15519 944	20689580 116	*	*	*
918	15430 186	20742513 388	*	*	*
919	15336 690	20795446 660	*	*	*
920	15244 531	20848379 932	*	*	*
921	15154 919	20901312 204	*	*	*
922	15147 918	20954245 476	*	*	*
923	15083 922	21007178 748	*	*	*
924	15023 076	21060111 020	*	*	*
925	14968 660	21113044 292	*	*	*
926	14918 929	21165977 564	*	*	*
927	14874 805	21218910 836	*	*	*
928	14838 917	21271843 108	*	*	*
929	14811 029	21324776 380	*	*	*
930	14783 119	21377709 652	*	*	*
931	14761 791	21430642 924	*	*	*

932	14739 637	21483575 196	*	*	*
933	14719 616	21536508 468	*	*	*
934	14701 983	21589441 740	*	*	*
935	14681 281	21642374 012	*	*	*
936	14662 959	21695307 284	*	*	*
937	14645 548	21748240 556	*	*	*
938	14612 790	21801173 828	*	*	*
939	14590 959	21854106 100	*	*	*
940	14574 146	21907039 372	*	*	*
941	14560 565	21960972 644	*	*	*
942	14543 119	22013905 916	*	*	*
943	14527 876	22066838 188	*	*	*
944	14508 231	22119771 460	*	*	*
945	14485 652	22172704 732	*	*	*
946	14460 030	22225637 004	*	*	*
947	14424 517	22278570 276	*	*	*
948	14391 125	22331503 548	*	*	*
949	14371 957	22384436 820	*	*	*
950	14352 801	22437369 092	*	*	*
951	14336 067	22490302 364	*	*	*
952	14326 011	22543235 636	*	*	*
953	14319 246	22596168 908	*	*	*
954	14311 214	22649102 180	*	*	*
955	14300 386	22702035 452	*	*	*
956	14324 211	22754968 724	*	*	*
957	14310 088	22807901 996	*	*	*
958	14306 067	22860835 268	*	*	*
959	14300 459	22913768 540	*	*	*
960	14300 246	22966701 812	*	*	*
961	14300 081	23019635 084	*	*	*
962	14300 061	23072568 356	*	*	*
963	14300 041	23125501 628	*	*	*
964	14300 021	23178434 900	*	*	*
965	14300 001	23231368 172	*	*	*
966	14300 001	23284301 444	*	*	*
967	14300 001	23337234 716	*	*	*
968	14300 001	23390168 000	*	*	*
969	14300 001	23443101 272	*	*	*
970	14300 001	23496034 544	*	*	*
971	14300 001	23548967 816	*	*	*
972	14300 001	23601901 088	*	*	*
973	14300 001	23654834 360	*	*	*
974	14300 001	23707767 632	*	*	*
975	14300 001	23760700 904	*	*	*
976	14300 001	23813633 176	*	*	*
977	14300 001	23866566 448	*	*	*
978	14300 001	23919500 720	*	*	*
979	14300 001	23972433 992	*	*	*
980	14300 001	24025366 264	*	*	*
981	14300 001	24078300 536	*	*	*
982	14300 001	24131233 808	*	*	*
983	14300 001	24184166 080	*	*	*
984	14300 001	24237100 352	*	*	*
985	14300 001	24290033 624	*	*	*
986	14300 001	24342966 896	*	*	*
987	14300 001	24395900 168	*	*	*
988	14300 001	24448833 440	*	*	*
989	14300 001	24501766 712	*	*	*
990	14300 001	24554700 000	*	*	*
991	14300 001	24607633 272	*	*	*
992	14300 001	24660566 544	*	*	*
993	14300 001	24713500 816	*	*	*
994	14300 001	24766433 088	*	*	*
995	14300 001	24819366 360	*	*	*
996	14300 001	24872300 632	*	*	*
997	14300 001	24925233 904	*	*	*
998	14300 001	24978166 176	*	*	*
999	14300 001	25031100 448	*	*	*
1000	14300 001	25084033 720	*	*	*
1001	14300 001	25136966 992	*	*	*
1002	14300 001	25189900 264	*	*	*
1003	14300 001	25242833 536	*	*	*
1004	14300 001	25295766 808	*	*	*
1005	14300 001	25348700 080	*	*	*
1006	14300 001	25401633 352	*	*	*
1007	14300 001	25454566 624	*	*	*
1008	14300 001	25507500 896	*	*	*
1009	14300 001	25560433 168	*	*	*
1010	14300 001	25613366 440	*	*	*
1011	14300 001	25666300 712	*	*	*
1012	14300 001	25719233 984	*	*	*
1013	14300 001	25772166 256	*	*	*
1014	14300 001	25825100 528	*	*	*
1015	14300 001	25878033 800	*	*	*
1016	14300 001	25930966 072	*	*	*
1017	14300 001	25983900 344	*	*	*
1018	14300 001	26036833 616	*	*	*
1019	14300 001	26089766 888	*	*	*
1020	14300 001	26142700 160	*	*	*
1021	14300 001	26195633 432	*	*	*
1022	14300 001	26248566 704	*	*	*
1023	14300 001	26301500 000	*	*	*
1024	14300 001	26354433 272	*	*	*
1025	14300 001	26407366 544	*	*	*
1026	14300 001	26460300 816	*	*	*
1027	14				

1180	15513 575	24745521 389	1304	23172 851	27047602 780	1428	16464 377	30812380 823
1181	15477 742	24769994 134	1305	34569 082	27082172 762	1429	16441 697	30828822 520
1182	15435 606	24767454 691	1306	37504 852	27110376 919	1430	16423 939	30845246 460
1183	15385 287	24791810 978	1307	40502 729	27159879 648	1431	16408 066	30861654 526
1184	15330 302	24807150 780	1308	43879 664	27203759 312	1432	16392 488	30878074 014
1185	15274 102	24822490 482	1309	46971 175	27250676 487	1433	16376 586	30894423 600
1186	15219 108	24837834 190	1310	49796 097	27300574 484	1434	16356 118	30910779 178
1187	15172 011	24853186 201	1311	52362 033	27352892 517	1435	16336 459	30927116 177
1188	15133 727	24868540 929	1312	54548 152	27407427 668	1436	16317 460	30943533 636
1189	15103 000	24883891 228	1313	56265 316	27463692 985	1437	16300 616	30959751 252
1190	15084 800	24899137 929	1314	57593 256	27521286 241	1438	16287 616	30976011 869
1191	15073 341	24914411 570	1315	58660 985	27580947 127	1439	16279 641	30992301 509
1192	15071 266	24929682 825	1316	59480 766	27639427 805	1440	16275 704	31008572 215
1193	15081 129	24945036 984	1317	60062 812	27699490 705	1441	16274 918	31024852 131
1194	15100 445	24960404 200	1318	60374 351	27759665 055	1442	16272 852	31041124 983
1195	15127 176	24975849 575	1319	60578 084	27820243 159	1443	16271 440	31057396 424
1196	15165 177	24988754 752	1320	60906 816	27880330 955	1444	16269 262	31073668 686
1197	15206 496	25003961 248	1321	59575 612	27939915 567	1445	16272 486	31089918 172
1198	15264 029	25019225 277	1322	58893 580	27999809 147	1446	16281 142	31106219 314
1199	15325 051	25034581 228	1323	58106 418	28059615 565	1447	16297 982	31122517 205
1200	15388 238	25049949 460	1324	57247 673	28114163 239	1448	16326 712	31138844 007
1201	15445 118	25065354 581	1325	56335 013	28170498 252	1449	16369 737	31155211 744
1202	15493 480	25080878 063	1326	55395 058	28228930 310	1450	16428 109	31171641 853
1203	15553 958	25096410 901	1327	54430 084	28289323 394	1451	16491 260	31188133 114
1204	15601 891	25111960 641	1328	53439 704	28337367 998	1452	16560 819	31204693 933
1205	15652 052	25127555 943	1329	52408 150	28383791 248	1453	16629 648	31221323 617
1206	15696 252	25143196 195	1330	51341 159	28437512 407	1454	16699 518	31238032 136
1207	15646 465	25158788 658	1331	50241 775	28487754 181	1455	16767 888	31254791 024
1208	15615 388	25174371 046	1332	49171 806	28538727 987	1456	16836 669	31271627 693
1209	15601 548	25189960 641	1333	47996 833	28589947 693	1457	16905 035	31288532 770
1210	15651 415	25205654 054	1334	46884 778	28641754 588	1458	16971 351	31305468 080
1211	15678 945	25221352 909	1335	45777 915	28697532 502	1459	17039 754	31322543 833
1212	15711 186	25237046 185	1336	44663 836	28721936 338	1460	17104 172	31339648 006
1213	15755 215	25252740 474	1337	43545 764	28756542 102	1461	17169 409	31356817 415
1214	15798 739	25268434 763	1338	42430 148	28789102 617	1462	17227 167	31374044 582
1215	15844 118	25284128 231	1339	41324 317	28844966 567	1463	17282 480	31391272 002
1216	15892 783	25300032 104	1340	40232 141	28899728 708	1464	17335 538	31408660 600
1217	15943 353	25316128 477	1341	39162 657	28928891 365	1465	17391 580	31426054 180
1218	15995 781	25332224 946	1342	38115 789	28967005 154	1466	17457 138	31443511 318
1219	16049 980	25348320 218	1343	37097 617	29009097 172	1467	17532 058	31461043 356
1220	16106 182	25364424 400	1344	36100 262	29040918 034	1468	17617 118	31478670 444
1221	16164 015	25380528 414	1345	35141 872	29075339 906	1469	17710 890	31496371 364
1222	16180 527	25396632 604	1346	34234 887	29109564 794	1470	17811 368	31514182 732
1223	16207 577	25412736 818	1347	33356 911	29148672 633	1471	17911 830	31532094 562
1224	16222 902	25428840 218	1348	32548 770	29195470 543	1472	18009 077	31550209 639
1225	16226 639	25444944 149	1349	31805 915	29220726 459	1473	18096 616	31568209 255
1226	16220 406	25461048 554	1350	31127 947	29238404 406	1474	18176 006	31586376 202
1227	16207 093	25477152 648	1351	30504 903	29266909 309	1475	18251 803	31604628 064
1228	16190 739	25493256 187	1352	29930 778	29298071 374	1476	18325 598	31622958 654
1229	16178 365	25509360 751	1353	29386 710	29328226 797	1477	18400 301	31641353 965
1230	16172 688	25525464 149	1354	28867 327	29359704 123	1478	18473 764	31659827 719
1231	16176 543	25541568 582	1355	28369 543	29393463 667	1479	18550 752	31678378 471
1232	16186 949	25557672 932	1356	27888 424	2943351 090	1480	18631 714	31697010 185
1233	16201 810	25573776 242	1357	27430 542	2948072 633	1481	18714 147	31715734 332
1234	16222 149	25590080 900	1358	26960 893	29546733 525	1482	18828 345	31734562 677
1235	16245 720	25606384 630	1359	26509 368	29614242 893	1483	18950 003	31753512 686
1236	16268 457	25622688 400	1360	26061 019	29682030 913	1484	19082 056	31772604 736
1237	16294 513	25639091 370	1361	25619 306	29750523 419	1485	19249 880	31791864 544
1238	16320 410	25655494 384	1362	25181 096	29819819 173	1486	19429 265	31811303 812
1239	16350 814	25671900 594	1363	24754 703	29890859 217	1487	19620 180	31830923 062
1240	16383 424	25688306 017	1364	24344 157	29963020 374	1488	19790 986	31850714 978
1241	16420 053	25704716 070	1365	23959 023	29944162 398	1489	19941 449	31870656 426
1242	16456 992	25721120 648	1366	23603 375	29925304 419	1490	20071 799	31890728 225
1243	16494 073	25737524 158	1367	23278 906	29906449 979	1491	20199 965	31910908 190
1244	16529 678	25753928 813	1368	22984 705	29890489 584	1492	20327 490	31931175 670
1245	16563 341	25770332 400	1369	22712 835	29876742 419	1493	20455 950	31951518 630
1246	16596 197	25786736 016	1370	22457 282	29863999 701	1494	20584 332	31971934 353
1247	16629 072	25803140 229	1371	22210 722	29851816 123	1495	20712 714	31992424 315
1248	16663 174	25819544 203	1372	21974 540	29840234 963	1496	20841 096	32012914 981
1249	16697 169	25835948 272	1373	21744 698	29829129 661	1497	20969 478	32033405 911
1250	16731 293	25852352 216	1374	21520 699	29818024 360	1498	21097 860	32053896 840
1251	16765 398	25868756 096	1375	21299 282	29807949 443	1499	21226 242	32074387 769
1252	16799 549	25885160 016	1376	21079 212	29797904 915	1500	21354 624	32094878 698
1253	16833 689	25901564 016	1377	20860 710	29787860 594	1501	21483 006	32115369 627
1254	16867 830	25917968 016	1378	20640 557	29777816 151	1502	21611 388	32135860 556
1255	16901 970	25934372 016	1379	20421 209	29767771 708	1503	21739 770	32156351 485
1256	16936 111	25950776 016	1380	20201 861	29757727 265	1504	21868 152	32176842 414
1257	16970 252	25967180 016	1381	19982 513	29747682 822	1505	21996 534	32197333 343
1258	17004 393	25983584 016	1382	19763 165	29737638 379	1506	22124 916	32217824 272
1259	17038 534	25999988 016	1383	19543 817	29727593 936	1507	22253 298	32238315 201
1260	17072 675	26016392 016	1384	19324 469	29717549 493	1508	22381 680	32258806 130
1261	17106 816	26032796 016	1385	19105 121	29707505 050	1509	22510 062	32279297 059
1262	17140 957	26049200 016	1386	18885 773	29697460 607	1510	22638 444	32300004 817
1263	17175 098	26065604 016	1387	18666 425	29687416 164	1511	22766 826	32320712 575
1264	17209 239	26082008 016	1388	18447 077	29677371 721	1512	22895 208	32341420 333
1265	17243 380	26098412 016	1389	18227 729	29667327 278	1513	23023 590	32362128 091
1266	17277 521	26114816 016	1390	18008 381	29657282 835	1514	23151 972	32382835 849
1267	17311 662	26131220 016	1391	17789 033	29647238 392	1515	23280 354	32403543 607
1268	17345 803	26147624 016	1392	17569 685	29637193 949	1516	23408 736	32424251 365
1269	17380 044	26164028 016	1393	17350 337	29627149 506	1517	23537 118	32444959 123
1270	17414 285	26180432 016	1394	17130 989	29617105 063	1518	23665 500	32465666 881
1271	17448 526	26196836 016	1395	16911 641	29607060 620	1519	23793 882	32486374 639
1272	17482 767	26213240 016	1396	16692 293	29597016 177	1520	23922 264	32507082 397
1273	17517 008	26229644 016	1397	16472 945	29586971 734	1521	24050 646	32527790 155
1274	17551 249	26246048 016	1398	16253 597	29576927 291	1522	24179 028	32548497 913
1275	17585 490	26262452 016	1399	16034 249	29566882 848	1523	24307 410	32569205 671
1276	17619 731	26278856 016	1400	15814 901	29556838 405	1524	24435 792	32589913 429
1277	17653 972	26295260 016	1401	15595 553	29546793 962	1525	24564 174	32610621 187
1278	17688 213	26311664 016	1402	15376 205	29536749 519	1526	2469	

1924	31846 447	44705893 641	"	"	2048	20152 502	48392654 847	"	"	2172	28003 246	52951530 046	"	"
1925	32164 070	44138057 716	"	"	2049	20121 778	48412776 626	"	"	2173	27570 719	52979100 764	"	"
1926	32811 134	44170868 850	"	"	2050	20081 934	48432858 560	"	"	2174	27164 464	53006265 228	"	"
1927	33769 609	44221450 541	"	"	2051	20043 701	48452895 571	"	"	2175	26780 404	53033045 632	"	"
1928	34361 694	44240000 234	"	"	2052	19995 503	48472891 071	"	"	2176	26418 871	53059448 503	"	"
1929	36967 270	44276967 604	"	"	2053	19965 627	48492856 701	"	"	2177	26083 260	53085557 765	"	"
1930	38588 564	44315655 068	"	"	2054	19932 581	48512809 281	"	"	2178	25770 383	53111318 147	"	"
1931	40446 845	44356002 813	"	"	2055	19895 028	48532760 309	"	"	2179	25480 719	53136798 866	"	"
1932	43501 916	44396504 729	"	"	2056	19859 959	48553718 268	"	"	2180	25211 534	53162010 590	"	"
1933	47380 395	44446885 124	"	"	2057	19827 924	48572691 193	"	"	2181	24955 138	53186965 338	"	"
1934	51624 219	44498509 143	"	"	2058	19807 377	48592688 569	"	"	2182	24706 777	53211672 514	"	"
1935	54743 314	44553981 657	"	"	2059	20030 489	48612719 059	"	"	2183	24454 426	53236126 740	"	"
1936	58630 198	44612613 856	"	"	2060	20076 926	48632795 985	"	"	2184	24198 188	53260324 928	"	"
1937	61082 842	44673696 698	"	"	2061	20137 780	48652933 765	"	"	2185	23919 183	53284264 111	"	"
1938	62402 375	44736509 073	"	"	2062	20211 914	48673145 679	"	"	2186	23677 237	53307941 347	"	"
1939	64210 063	44800815 136	"	"	2063	20297 264	48693442 943	"	"	2187	23415 768	53331357 116	"	"
1940	65175 980	44865901 117	"	"	2064	20383 384	48713826 327	"	"	2188	23159 411	53354816 527	"	"
1941	67383 633	44931774 749	"	"	2065	20465 418	48734289 754	"	"	2189	22918 068	53377434 595	"	"
1942	66078 195	44997852 945	"	"	2066	20536 551	48754826 296	"	"	2190	22701 799	53400136 594	"	"
1943	66162 509	45064015 454	"	"	2067	20605 933	48775432 230	"	"	2191	22520 513	53422656 907	"	"
1944	66067 418	45130082 871	"	"	2068	20681 734	48796113 963	"	"	2192	22386 137	53445043 044	"	"
1945	67995 200	45195878 071	"	"	2069	20767 015	48816880 978	"	"	2193	22301 400	53467344 444	"	"
1946	65391 101	45261269 173	"	"	2070	20856 892	48837137 560	"	"	2194	22257 330	53489601 774	"	"
1947	64867 370	45326136 543	"	"	2071	20950 356	48858687 916	"	"	2195	22252 869	53511854 643	"	"
1948	64187 755	45390324 298	"	"	2072	21048 959	48879716 875	"	"	2196	22283 488	53534138 131	"	"
1949	63325 678	45453649 976	"	"	2073	21145 578	48900882 453	"	"	2197	22334 184	53556472 285	"	"
1950	62594 320	45518943 606	"	"	2074	21244 727	48922124 677	"	"	2198	22370 522	53578916 457	"	"
1951	61140 806	45577081 562	"	"	2075	21351 144	48943475 822	"	"	2199	22393 615	53601236 421	"	"
1952	60001 696	45636981 258	"	"	2076	21474 889	48964950 711	"	"	2200	22387 064	53623623 485	"	"
1953	58904 441	45695982 700	"	"	2077	21613 585	48986564 293	"	"	2201	22346 427	53645969 911	"	"
1954	57247 401	45752350 100	"	"	2078	21757 039	49008321 323	"	"	2202	22327 026	53668241 938	"	"
1955	55879 664	45808700 764	"	"	2079	21899 240	49030250 564	"	"	2203	22284 061	53690018 900	"	"
1956	54525 090	45863254 854	"	"	2080	22056 709	49052277 272	"	"	2204	22047 930	53711458 930	"	"
1957	53255 614	45916470 768	"	"	2081	22222 822	49074499 094	"	"	2205	21917 564	53733476 494	"	"
1958	52007 393	45968478 161	"	"	2082	22410 908	49096909 998	"	"	2206	21784 762	53755611 236	"	"
1959	50823 367	46019049 826	"	"	2083	22605 903	49120701 011	"	"	2207	21655 525	53777616 781	"	"
1960	49650 722	46068951 150	"	"	2084	22814 362	49142490 263	"	"	2208	21542 537	53799319 518	"	"
1961	48476 973	46117428 123	"	"	2085	23046 121	49165756 584	"	"	2209	21452 072	53820811 390	"	"
1962	47309 527	46164737 650	"	"	2086	23262 324	49189376 708	"	"	2210	21393 024	53842204 415	"	"
1963	46138 391	46210886 041	"	"	2087	23500 655	49213367 364	"	"	2211	21359 846	53863564 260	"	"
1964	45002 066	46258885 027	"	"	2088	23762 647	49237010 111	"	"	2212	21335 782	53884933 047	"	"
1965	43863 003	46299741 109	"	"	2089	24042 244	49262454 255	"	"	2213	21323 823	53906136 326	"	"
1966	42735 901	46342487 010	"	"	2090	24336 965	49287518 220	"	"	2214	21324 306	53927730 632	"	"
1967	41637 160	46384124 170	"	"	2091	24645 400	49312877 620	"	"	2215	21342 927	53949203 324	"	"
1968	40575 858	46424700 027	"	"	2092	25061 068	49339391 688	"	"	2216	21400 967	53970718 921	"	"
1969	39585 744	46464551 772	"	"	2093	25587 238	49366928 916	"	"	2217	21480 218	53992248 218	"	"
1970	38599 685	46502855 457	"	"	2094	26237 585	49394016 501	"	"	2218	21580 225	54013811 927	"	"
1971	37700 401	46540555 958	"	"	2095	26998 969	49416745 470	"	"	2219	21744 788	54035356 715	"	"
1972	36857 076	46577415 033	"	"	2096	27883 710	49431319 181	"	"	2220	21951 001	54056887 806	"	"
1973	36047 311	46613460 345	"	"	2097	28899 134	49447038 314	"	"	2221	22209 876	54078407 682	"	"
1974	35264 319	46648872 603	"	"	2098	29954 728	49464076 729	"	"	2222	22588 033	54099916 815	"	"
1975	34508 611	46684233 475	"	"	2099	31068 879	49482491 608	"	"	2223	23090 653	54121477 498	"	"
1976	33778 701	46717012 175	"	"	2100	32342 146	49502883 754	"	"	2224	23726 404	54143083 903	"	"
1977	33089 176	46750101 581	"	"	2101	33782 501	49518116 255	"	"	2225	24598 998	54164765 901	"	"
1978	32435 789	46782552 120	"	"	2102	35397 173	49533381 916	"	"	2226	25677 750	54186551 611	"	"
1979	31820 623	46814537 811	"	"	2103	37200 246	49548687 651	"	"	2227	27001 941	54208403 926	"	"
1980	31241 534	46845990 286	"	"	2104	39090 077	49564030 630	"	"	2228	28595 854	54230352 956	"	"
1981	30697 940	46876296 825	"	"	2105	40225 685	49579351 315	"	"	2229	29949 232	54252402 183	"	"
1982	30190 999	46906487 324	"	"	2106	41870 182	49594726 497	"	"	2230	31322 106	54274424 594	"	"
1983	29723 131	46936720 435	"	"	2107	43840 602	49610107 099	"	"	2231	32848 033	54296962 327	"	"
1984	29336 707	46966555 157	"	"	2108	46089 497	49625496 596	"	"	2232	34519 354	54318882 610	"	"
1985	28991 137	46994548 293	"	"	2109	48675 227	49640971 822	"	"	2233	36350 800	54340306 960	"	"
1986	28674 488	47023222 781	"	"	2110	51693 455	49657408 367	"	"	2234	38357 477	54362323 437	"	"
1987	28375 557	47051598 139	"	"	2111	55193 708	49674909 076	"	"	2235	40549 742	54384893 179	"	"
1988	28099 198	47078797 431	"	"	2112	59442 554	49693439 908	"	"	2236	42998 009	54407110 634	"	"
1989	27799 937	47107457 374	"	"	2113	64697 328	49717787 326	"	"	2237	45726 606	54429492 194	"	"
1990	27474 595	47134931 969	"	"	2114	71234 822	49743122 148	"	"	2238	48910 479	54451576 672	"	"
1991	27163 093	47162095 061	"	"	2115	79188 435	49768511 583	"	"	2239	52143 601	54473720 273	"	"
1992	26841 644	47189956 706	"	"	2116	88632 456	49793914 040	"	"	2240	55999 754	54495960 027	"	"
1993	26521 648	47218446 701	"	"	2117	99720 960	49819243 999	"	"	2241	60933 770	54518333 968	"	"
1994	26216 049	47247617 749	"	"	2118	112417 317	49844635 317	"	"	2242	67366 656	54540910 484	"	"
1995	25816 422	47276734 172	"	"	2119	128411 154	49869084 471	"	"	2243	74897 027	54563689 451	"	"
1996	25457 960	47306892 132	"	"	2120	147948 330	49894441 301	"	"	2244	83902 277	54586691 758	"	"
1997	25064 726	47337986 658	"	"	2121	170982 020	49919781 721	"	"	2245	94690 567	54609491 032	"	"
1998	24734 460	47369121 118	"	"	2122	196787 833	49946236 554	"	"	2246	108398 366	54633131 591	"	"
1999	24383 271	47400264 389	"	"	2123	227085 089	49973021 643	"	"	2247	124958 549	54656860 240	"	"
2000	24050 656	47431155 045	"	"	2124	263193 706	500022953 350	"	"	2248	145965 163	54680525 403	"	"
2001	23739 691	47461494 734	"	"	2125	306288 169	50035541 518	"	"	2249	174688 588	54704293 990	"	"
2002	23439 616	47492155 278	"	"	2126	358287 212	50068782 318	"	"	2250	208699 976	54728093 987	"	"
2003	23210 636	47523053 924	"	"	2127	419521 474	5010213							

2296	71729 840	56386356 570	2420	261725 984	60594957 556	2544	23831 563	65592171 808
2297	72282 395	56458738 965	2421	26442 431	60622309 987	2545	23869 358	65616601 160
2298	72828 480	56531567 445	2422	26703 489	60649103 477	2546	23926 488	65636064 654
2299	72989 574	56605457 020	2423	26961 062	60676064 538	2547	23981 269	65653958 023
2300	73281 656	56677378 676	2424	27205 405	60703269 943	2548	24051 999	65671101 921
2301	73375 854	56749754 530	2425	27439 568	60730709 511	2549	24099 290	65711210 211
2302	73746 987	56821501 517	2426	27701 015	60758410 527	2550	24142 862	65736253 077
2303	70958 214	56894249 761	2427	28010 477	60786421 604	2551	24193 218	65760446 391
2304	70636 560	56966249 522	2428	28402 760	60814217 780	2552	24255 274	65784701 765
2305	69026 741	57031523 065	2429	28841 491	60843665 271	2553	24344 854	65809046 619
2306	67471 830	57099444 892	2430	29290 495	60872955 767	2554	24479 346	65833255 964
2307	66730 294	57166171 185	2431	29727 943	60902083 710	2555	24638 060	65858164 025
2308	65569 886	57231541 071	2432	30162 267	60931584 077	2556	24801 990	65882966 023
2309	63910 257	57296451 508	2433	30577 757	60961423 824	2557	24951 709	65907917 332
2310	62373 899	57357225 207	2434	30966 335	60991360 168	2558	25085 287	65933003 018
2311	60788 609	57418615 873	2435	31344 119	61024734 288	2559	25203 822	65958206 840
2312	59165 761	57477282 657	2436	31725 344	61057489 632	2560	25301 337	65983308 378
2313	57547 720	57535300 354	2437	32137 432	61090277 064	2561	25370 562	66008378 940
2314	55953 410	575941283 763	2438	32733 850	61123460 914	2562	25425 446	66034304 385
2315	54404 964	57646888 728	2439	33241 859	61155702 773	2563	25463 377	66059767 762
2316	52943 224	57698631 951	2440	33836 228	61189439 001	2564	25502 948	66085270 710
2317	51581 412	57750213 365	2441	34624 275	61222461 276	2565	25535 961	66110846 671
2318	50310 680	57800253 445	2442	35778 220	61259441 496	2566	25720 611	66136567 282
2319	49091 500	57849614 944	2443	37247 083	61297188 579	2567	25919 562	66162486 844
2320	47909 526	57897524 270	2444	38672 252	61335600 831	2568	26168 407	66188555 252
2321	46769 211	57944287 481	2445	40087 586	61373948 416	2569	26439 207	66215094 459
2322	45658 606	57990901 828	2446	41549 318	61412466 727	2570	26724 133	66241808 592
2323	44596 098	58034254 206	2447	43150 864	61450947 591	2571	27020 332	66268839 124
2324	43578 485	58078120 691	2448	44875 374	61489622 965	2572	27314 630	66296153 754
2325	42590 680	58120711 378	2449	46741 644	61528457 609	2573	27618 843	66323742 597
2326	41637 582	58162748 966	2450	48756 796	61567399 504	2574	27944 737	66351583 333
2327	40740 382	58203079 522	2451	50908 361	61606492 765	2575	28282 163	66380549 497
2328	39868 986	58242948 298	2452	53205 035	61645700 800	2576	28730 117	66407995 614
2329	39051 641	58281999 659	2453	55658 050	61685278 850	2577	28602 169	66436597 783
2330	38261 380	58320201 349	2454	58270 333	61725139 285	2578	28901 799	66465499 582
2331	37494 245	58357493 281	2455	61062 900	61765614 000	2579	29183 211	66494839 215
2332	36708 941	58394449 515	2456	64088 194	62030711 384	2580	29450 953	66524133 747
2333	35954 787	58430404 301	2457	67465 283	62106636 607	2581	29735 125	66553868 872
2334	35223 794	58466288 056	2458	71272 413	62183909 080	2582	30034 152	66583903 024
2335	34520 180	58500148 236	2459	75610 064	62261583 144	2583	30345 173	66614248 197
2336	33849 064	58530079 500	2460	80520 150	62341371 904	2584	30679 848	66645208 036
2337	33230 419	58567246 719	2461	86059 076	62421648 373	2585	31111 637	66676079 682
2338	32650 442	58604906 271	2462	92345 970	62503299 344	2586	31730 389	66707770 071
2339	32118 364	58642024 655	2463	99410 753	62586860 096	2587	32435 994	66740206 066
2340	31625 199	58679638 053	2464	107423 687	62673432 385	2588	33316 445	66773104 511
2341	31174 049	58716719 585	2465	117478 950	62764061 736	2589	33790 895	66807142 406
2342	30754 162	58754573 746	2466	129038 372	62858710 108	2590	34425 031	66841567 437
2343	30354 374	58793288 080	2467	142515 314	62957486 149	2591	35000 671	66876568 108
2344	29961 359	58831892 309	2468	158114 816	63061882 110	2592	35586 222	66912154 331
2345	29586 974	58870495 813	2469	175919 848	63170946 328	2593	36317 043	66948786 381
2346	29167 026	58908446 946	2470	196198 533	63286238 462	2594	37237 300	66985729 048
2347	28760 832	58947386 741	2471	209439 857	63407831 319	2595	38197 133	67023926 181
2348	28348 716	58987175 457	2472	234623 271	63549024 591	2596	39234 546	67063160 727
2349	27933 766	59029669 223	2473	264186 354	63704322 948	2597	40520 945	67103681 672
2350	27511 658	59074991 281	2474	298232 837	63874218 782	2598	42440 799	67146123 381
2351	27117 714	59124907 995	2475	348787 711	64068844 493	2599	44827 790	67190410 171
2352	26720 579	59171028 574	2476	416298 191	64281421 683	2600	46307 779	67236717 950
2353	26327 303	59220735 877	2477	501579 718	64519902 211	2601	49820 311	67286538 661
2354	25942 327	59273298 707	2478	605375 497	64783317 898	2602	54647 270	67341485 931
2355	25569 654	59328884 946	2479	738599 688	65085917 482	2603	60665 080	67401551 041
2356	25205 367	59387400 513	2480	90457 018	65420374 600	2604	66469 475	67466200 485
2357	24865 894	59448936 207	2481	111127 479	65844597 079	2605	68319 240	67534519 725
2358	24544 762	59513490 969	2482	138671 587	66373248 666	2606	71276 247	67605795 972
2359	24227 218	59581378 187	2483	169999 999	66937150 665	2607	73803 901	67679596 873
2360	23914 614	59651704 501	2484	206434 463	67641781 128	2608	75972 171	67755577 044
2361	23603 150	59723559 651	2485	246499 583	68469434 711	2609	77565 248	67833137 292
2362	23308 545	59798206 196	2486	290708 215	69413142 927	2610	78559 648	67911696 941
2363	23024 249	59874264 445	2487	349898 845	70493012 772	2611	79090 456	67990787 397
2364	22752 657	59961988 065	2488	426723 567	71761324 316	2612	79426 326	68071088 511
2365	22497 716	60049656 798	2489	519008 266	73293831 705	2613	79031 665	68149057 897
2366	22258 672	60140285 470	2490	638012 059	75067961 764	2614	78531 780	68222589 677
2367	22034 739	60235030 209	2491	784233 751	77111895 545	2615	77786 075	68305375 752
2368	21827 967	60334903 176	2492	958443 783	79455198 298	2616	76809 613	68382185 365
2369	21639 054	60439209 045	2493	116067 011	821092 309	2617	75210 953	68464102 317
2370	21467 443	60544456 410	2494	136611 276	851585 585	2618	74289 819	68532096 137
2371	21319 173	60661647 583	2495	158121 608	881426 192	2619	72874 342	68604970 479
2372	21183 414	60784878 998	2496	181695 383	9149195 383	2620	71411 682	68676382 162
2373	21059 776	60911376 775	2497	208581 111	9533883 543	2621	69900 001	68746290 163
2374	20948 571	61043090 344	2498	239232 344	10000638 157	2622	68344 290	68816317 917
2375	20849 745	61180470 203	2499	274167 256	10542060 044	2623	66757 410	68881391 863
2376	20764 624	61323877 826	2500	304616 234	11144522 275	2624	65334 334	68946549 296
2377	20694 887	61469141 715	2501	329977 885	11809200 060	2625	63154 498	69010083 799
2378	20639 492	61623604 644	2502	359414 365	12592914 165	2626	61895 376	69071848 680
2379	20593 928	61785474 872	2503	393105 253	13503219 618	2627	60261 093	69132240 463
2380	20557 455	61956567 325	2504	430670 659	14560850 277	2628	58651 981	69190892 444
2381	20529 944	62136202 278	2505	473071 001	15804024 278	2629	57095 060	69247987 504
2382	20510 680	6232031 958	2506	52136 628	16207050 907	2630	55666 444	69303597 847
2383	20499 581	62512726 519	2507	57695 280	16850064 186	2631	54166 840	69357660 917
2384	20497 259	62712996 758	2508	64096 475	1768942 661	2632	52773 476	69410534 394
2385	20504 096	62929493 854	2509	71451 158	18707483 819	2633	51432 332	69461966 726
2386	20519 289	63149475 113	2510	79933 700	19947478 519	2634	50146 512	69512113 237
2387	20539 249	63385554 362	2511	89607 122	21376685 640	2635	48918 632	69561031 890
2388	20569 828	63634651 190	2512	100431 062	22971416 203	2636	47750 389	69608783 678
2389	20609 889	63898781 079	2513	113499 633	24818916 336	2637	46623 223	69655405 902
2390	20660 220	64180058 318	2514	129272 880	26848618 215	2638	45533 921	69700939 823
2391	20721 441	64483369 761	2515	147601 690	2913230 906	2639	44499 958	69745439 781
2392	20794 787	64804836 556	2516	169804 620	4090095 476	2640	43532 898	69788972 680
2393	20878 162	65149754 718	2517	206664 167	5492599 753	2641	42634 301	69831606 981
2394	20972 164	654997941 882	2518	26121				

Table with multiple columns of numbers. The first column contains numbers from 1700 to 3700. Subsequent columns contain various numerical data points, likely representing a sequence or specific values for each row.

3040	54675 094	80830321 834	3164	53907 253	92812894 478	3288	43741 760	90233503 432
3041	54231 358	80884553 392	3165	52653 350	92865347 828	3289	42941 051	90276534 484
3042	53839 038	80931627 410	3166	51439 066	92916988 893	3290	42183 627	90318718 637
3043	51173 857	80988856 287	3167	50259 140	92967345 998	3291	41422 237	90360190 857
3044	50173 611	87039029 898	3168	49125 404	93016371 402	3292	40790 526	90400981 183
3045	48944 636	87087971 534	3169	48092 264	93064463 666	3293	40124 634	90441105 818
3046	47850 900	87135805 334	3170	47066 537	93111670 203	3294	39492 586	90482058 403
3047	46827 295	87182663 728	3171	46057 613	93158127 816	3295	38901 810	90519906 214
3048	45899 071	87238531 800	3172	45776 308	93203904 124	3296	38356 474	90557856 608
3049	45026 930	87293558 730	3173	45108 109	93249012 233	3297	37857 627	90595714 315
3050	44215 486	87341774 186	3174	44399 946	93293452 178	3298	37411 115	90633125 543
3051	43460 508	87391291 693	3175	43746 904	93337998 173	3299	37020 409	90670145 830
3052	42738 884	87440393 578	3176	43018 274	93382017 446	3300	36698 003	90706843 842
3053	42039 670	87496011 248	3177	42255 447	93422473 394	3301	36444 684	90743289 526
3054	41344 681	87547357 929	3178	41466 883	93463940 277	3302	36247 509	90779537 035
3055	40662 472	87603820 401	3179	40675 253	93504615 330	3303	36075 382	90815590 417
3056	40006 538	87662026 949	3180	39907 560	93545523 089	3304	35937 505	90851447 923
3057	39375 325	87667402 284	3181	39198 502	93583721 592	3305	35664 200	90887112 123
3058	38767 258	87664616 542	3182	38558 291	93622279 983	3306	35478 651	90922590 773
3059	38182 364	87684551 906	3183	37995 717	93662025 700	3307	35307 419	90957898 192
3060	37613 696	87721965 602	3184	37491 444	93697767 143	3308	35138 288	90993036 480
3061	37060 388	87759025 089	3185	37033 508	93734800 651	3309	35002 788	900028039 269
3062	36528 314	87795564 303	3186	36605 845	93771406 497	3310	34933 196	900062972 465
3063	36032 421	87831616 725	3187	36195 650	93807602 146	3311	34960 124	900099732 589
3064	35589 654	87867206 379	3188	35778 435	93843380 381	3312	35033 335	900132668 923
3065	35132 216	87902344 905	3189	35323 006	93879132 617	3313	35125 522	90018460 734
3066	34719 319	87937068 914	3190	34926 211	93913658 888	3314	35216 288	900203703 734
3067	34329 799	87971398 713	3191	34523 962	93948182 851	3315	35321 166	900238289 900
3068	33988 670	88005387 384	3192	34181 351	93982364 202	3316	35456 236	900274085 136
3069	33710 559	88039097 943	3193	33911 862	94016276 064	3317	35604 290	900309689 425
3070	33471 394	88072418 477	3194	33717 428	94049993 492	3318	35765 750	900336682 956
3071	33238 715	88105807 952	3195	33527 202	94083570 694	3319	35969 486	900381424 662
3072	33018 040	88138826 352	3196	33340 036	94117050 730	3320	36199 207	900417623 869
3073	32807 041	88171633 393	3197	33144 942	94150455 672	3321	36417 379	900454041 248
3074	32605 698	88204454 490	3198	32922 867	94183878 666	3322	36580 635	900490671 262
3075	32426 895	88236665 887	3199	32733 721	94217002 380	3323	36731 625	900527353 908
3076	32251 196	88268757 083	3200	32511 570	94250113 959	3324	36897 461	900564250 707
3077	32093 848	88300670 931	3201	32307 612	94283211 571	3325	37068 034	90060119 004
3078	31947 878	88332346 739	3202	32096 230	94316677 801	3326	37236 838	90063875 844
3079	31814 847	88364139 526	3203	31900 988	94350385 790	3327	37408 430	90067640 748
3080	31693 962	88395963 188	3204	31749 403	94384233 192	3328	37577 066	900713951 278
3081	31584 998	88426179 186	3205	31618 424	94415581 517	3329	37848 142	900752299 419
3082	31495 666	88457182 882	3206	31504 478	94449145 904	3330	38099 531	900791208 590
3083	31428 405	88488106 287	3207	31403 667	94482907 291	3331	38566 141	90083075 091
3084	31373 915	88519802 002	3208	31314 297	94516818 979	3332	39048 793	90087144 884
3085	31331 659	88549842 862	3209	31236 037	94550822 016	3333	39599 944	900912003 824
3086	31299 906	88580734 767	3210	31168 840	94584866 856	3334	40266 676	900953690 504
3087	31281 806	88611688 633	3211	31112 622	94618952 478	3335	42242 633	90099113 146
3088	31276 355	88642724 988	3212	31067 609	94653692 287	3336	44143 199	901039266 346
3089	31281 358	88673812 746	3213	31032 482	94688047 769	3337	46101 228	901083327 574
3090	31232 226	88705205 472	3214	31006 223	94722579 993	3338	48174 159	901128447 733
3091	31225 139	88736730 611	3215	31000 863	94755905 863	3339	46625 002	901175072 734
3092	31221 246	88768442 857	3216	31005 307	94790259 170	3340	47923 354	901222996 088
3093	31203 161	88800305 018	3217	31019 089	94824423 817	3341	49037 356	901272033 444
3094	31187 120	88832270 138	3218	31044 992	94859058 830	3342	50197 384	901322230 828
3095	31235 866	88864706 004	3219	31086 665	94893521 494	3343	52231 523	901374462 351
3096	32063 821	88896369 825	3220	31150 614	94928112 108	3344	54614 160	901429076 512
3097	32089 807	88928429 633	3221	31239 906	94963022 014	3345	58575 975	901487952 487
3098	32054 768	88960464 400	3222	31341 654	94998378 866	3346	62818 623	90154670 362
3099	32003 648	88992468 048	3223	31465 624	95034402 969	3347	69161 545	901620833 107
3100	32004 104	89024472 152	3224	31615 624	95070918 593	3348	74409 811	901695242 918
3101	32094 913	89056567 065	3225	31709 990	95107923 683	3349	78712 168	901773955 086
3102	32287 629	89088554 694	3226	31846 926	95145402 609	3350	82264 476	901853616 982
3103	32340 847	89119139 316	3227	32005 958	95183308 593	3351	85467 600	90193219 620
3104	32814 347	89154209 883	3228	32180 841	95221829 434	3352	87986 095	902030175 256
3105	33086 182	89187296 066	3229	32367 177	95261191 611	3353	89636 584	902119811 840
3106	33317 217	89220613 283	3230	32570 473	95301466 068	3354	91399 855	902210971 696
3107	33490 793	89253104 076	3231	32794 618	95343497 078	3355	92352 616	90230324 311
3108	33614 532	89282718 608	3232	33031 427	95384619 505	3356	92813 116	902396037 427
3109	33724 975	89312453 582	3233	33280 361	95427439 867	3357	92854 290	902488891 817
3110	33914 314	89354367 896	3234	33542 704	95471332 571	3358	92460 528	902581352 245
3111	34221 345	89395989 241	3235	33927 745	95516309 815	3359	91875 415	902673227 661
3112	34688 467	89437279 708	3236	34340 160	95563272 920	3360	91323 969	902765067 582
3113	35203 861	89484958 170	3237	34781 800	95609155 775	3361	90712 812	902854533 432
3114	35987 562	89532699 132	3238	35240 370	95656876 146	3362	89029 453	902943562 885
3115	36684 734	89583225 866	3239	35736 679	95707412 824	3363	87700 378	903031263 264
3116	37460 746	89636714 612	3240	36289 059	95754722 646	3364	86245 824	903117509 083
3117	38448 593	89690152 310	3241	36904 058	95804127 706	3365	84784 907	903203462 523
3118	39441 703	89744609 913	3242	37584 461	95859477 167	3366	83251 395	903285515 389
3119	40393 810	89799303 723	3243	38338 712	95916058 879	3367	81643 266	90336158 655
3120	41225 269	89854228 992	3244	39178 929	96003784 808	3368	79910 879	903447069 534
3121	41930 998	89911759 090	3245	38635 774	96140150 583	3369	78080 617	903525150 151
3122	42899 539	89974546 628	3246	39068 810	96273722 172	3370	76154 678	90360404 713
3123	43400 600	89985715 328	3247	39518 536	96322377 320	3371	74174 681	903675479 510
3124	44531 674	89991686 902	3248	40002 002	96371009 222	3372	72186 237	903747665 747
3125	46073 997	89947760 890	3249	40530 400	96412939 722	3373	70292 224	903817957 971
3126	47772 339	89909833 218	3250	41107 418	96460917 240	3374	68623 982	90388680 953
3127	49703 312	90045236 550	3251	41741 884	96509578 727	3375	67134 226	90395351 179
3128	51684 087	90098700 637	3252	42445 608	96558204 335	3376	65709 281	904019424 461
3129	53762 655	90158373 292	3253	43204 799	96606979 134	3377	64289 983	904083714 444
3130	56091 938	90224765 320	3254	44027 336	96656386 470	3378	62844 322	904146538 767
3131	57801 361	90296761 177	3255	44915 838	96706322 308	3379	61480 913	904207939 540
3132	59649 886	90372211 066	3256	45870 496	96756788 727	3380	60090 161	904267849 701
3133	61625 607	90453636 671	3257	46904 814	96807806 617	3381	58849 124	904326340 825
3134	63810 456	90537447 126	3258	48000 223	96859206 840	3382	57792 092	904383532 917
3135	66251 278	90623708 404	3259	49268 522	96910634 362	3383	56900 815	904439533 732
3136	68801 361	90711509 766	3260	50708 810	96963087 981	3384	56190 703	904494099 915
3137	71464 185	90800133 951	3261	52234 634	97016480 806	3385	55781 475	904548191 411
3138	74297 401	90889						

3412	58887 495	10771234 964	3660	64272 717	121835005 435
3413	38829 678	105770064 642	3661	62190 386	121898295 822
3414	38274 932	104085789 478	3662	61008 925	121960204 746
3415	38580 840	105873770 415	3663	60612 242	1220208 16 988
3416	38412 226	105887882 641	3664	60444 646	12208260 634
3417	58272 717	105924055 358	3665	58278 927	122138639 561
3418	38242 897	105962298 256	3666	57371 555	122196011 317
3419	38447 254	106003745 510	3667	56444 279	122252454 195
3420	38775 701	106040461 664	3668	55635 362	122308090 737
3421	38948 286	106078415 896	3669	54820 206	122363010 963
3422	39094 132	106117510 029	3670	54269 017	122417279 980
3423	39170 237	106150680 266	3671	53668 644	122470948 624
3424	39198 589	106195879 235	3672	53065 946	122524014 570
3425	39209 869	106235089 234	3673	52444 260	122576428 830
3426	39277 493	106274306 217	3674	51720 483	122628149 312
3427	39586 032	106313922 749	3675	51026 128	12269175 440
3428	39411 708	106353434 457	3676	50293 499	122759568 940
3429	39248 589	106393453 647	3677	49843 305	122779412 545
3430	39409 907	106434522 954	3678	49410 005	12282823 249
3431	40694 938	106475222 889	3679	49090 261	122877912 511
3432	40937 669	106516100 458	3680	48830 297	122926742 807
3433	41316 626	106554776 984	3681	48574 492	122975317 309
3434	42108 100	106599585 090	3682	48325 116	123024342 416
3435	42966 585	106642541 475	3683	48159 570	123073180 985
3436	43770 677	106686312 153	3684	48062 762	123119864 747
3437	44552 387	106730864 540	3685	48001 770	123167866 517
3438	45307 218	106776171 921	3686	47981 488	123215848 005
3439	46149 478	106823438 358	3687	47982 873	123263881 011
3440	47035 125	106870519 463	3688	47986 990	123311871 871
3441	49105 027	106919261 490	3689	48090 814	123359908 685
3442	50803 627	106967065 117	3690	48222 942	123408331 627
3443	53882 747	107029046 464	3691	48970 577	123456789 204
3444	54886 506	107078333 970	3692	49720 436	123505023 640
3445	59065 001	107136898 971	3693	50448 856	123553741 496
3446	67445 721	107204474 692	3694	51151 449	123602622 945
3447	70276 618	107280601 310	3695	51883 544	123666906 480
3448	83361 683	107367290 964	3696	52652 706	123731349 085
3449	88905 747	107452868 745	3697	53596 260	123796755 455
3450	93569 548	107546438 203	3698	54606 856	123863362 312
3451	107490 990	107643939 703	3699	55843 144	123937205 452
3452	107443 037	107734039 619	3700	57186 159	124019191 312
3453	107462 247	107824693 386	3701	58735 403	124091217 015
3454	104318 517	107915127 904	3702	60722 847	124163849 862
3455	105295 150	108005607 054	3703	65339 007	124236919 469
3456	105547 621	108102114 685	3704	7136 428	124190325 897
3457	105196 215	108202701 060	3705	78993 158	124291915 054
3458	104233 805	108317104 803	3706	86184 778	124355503 832
3459	102988 804	108424494 610	3707	92125 842	12442629 674
3460	101708 100	108576063 015	3708	96959 306	124494589 040
3461	99977 059	108676040 974	3709	101007 606	124564596 646
3462	98343 837	108774384 689	3710	104649 723	124634582 681
3463	96360 720	108870645 531	3711	107383 814	124704580 184
3464	94307 728	108964953 259	3712	109664 988	124774055 172
3465	92142 877	109076966 136	3713	111036 090	124843991 180
3466	89900 909	109186907 046	3714	111737 775	12491825 456
3467	87609 637	109293606 682	3715	111988 268	124993262 787
3468	85252 269	109391985 951	3716	111852 486	125068699 710
3469	82862 428	109490272 379	3717	111411 287	125145080 997
3470	80497 203	109588321 582	3718	110499 432	125221980 429
3471	78191 449	109686411 131	3719	109188 656	125300769 086
3472	75958 933	109784970 264	3720	107859 135	125380683 883
3473	73826 016	109883196 080	3721	105729 160	125461992 047
3474	71862 132	109981558 212	3722	103657 890	125545493 937
3475	70059 255	109855117 447	3723	101456 822	125631206 749
3476	68352 206	109952469 653	3724	99142 838	125717957 877
3477	66695 680	10981855 333	3725	96761 903	125805111 091
3478	65066 508	110053231 640	3726	94451 309	125893562 399
3479	63499 933	110116691 473	3727	92362 426	125983194 826
3480	61858 640	110278590 213	3728	90361 787	126073638 613
3481	60365 726	110358815 986	3729	88450 082	126164736 695
3482	58712 397	110439556 247	3730	86642 188	126256811 218 834
3483	57247 888	110544774 135	3731	84422 295	126350564 179
3484	55822 018	11064006 153	3732	82315 607	126445977 146
3485	54625 601	110736281 347	3733	80248 902	126543020 348
3486	53444 237	110832776 570	3734	78237 970	126640101 818
3487	52347 660	110929107 230	3735	76318 536	126737258 554
3488	51227 432	111025210 662	3736	74516 719	126834975 272
3489	50150 494	111121752 156	3737	72840 579	126932615 852
3490	49082 278	111220160 664	3738	71181 681	127030820 404
3491	48028 311	111328600 746	3739	69530 156	127129066 171
3492	47111 515	111437052 260	3740	67921 080	127227408 769
3493	47007 424	111545499 685	3741	66381 337	127335310 106
3494	46775 057	111653947 174	3742	64888 812	127443809 918
3495	46751 909	111762394 244	3743	63446 909	127552948 827
3496	46539 290	111870795 540	3744	62084 188	127662553 014
3497	46225 629	111979241 169	3745	60774 762	127771324 576
3498	46241 658	112087682 827	3746	59516 820	127880647 996
3499	45768 395	112196131 322										

1784	95337	945	130199550	628	#	#	3889	71732	577	139207417	768	#	#	3994	12631	220	148009133	645	#	#
1785	99819	410	130299370	038	#	#	3890	70426	490	13927844	258	#	#	3995	126880	172	148135813	878	#	#
1786	103321	482	130402691	520	#	#	3891	69113	274	139346957	532	#	#	3996	126421	361	148262235	176	#	#
1787	106406	625	130509992	445	#	#	3892	67765	726	1394278	722	#	#	3997	125519	721	148387754	908	#	#
1788	10848	265	130617945	811	#	#	3893	66455	382	139481163	111	#	#	3998	124591	075	148512045	975	#	#
1789	110566	687	130728511	898	#	#	3894	65198	889	139546361	000	#	#	3999	123534	542	148634880	917	#	#
1790	111850	302	130840362	200	#	#	3895	64118	000	139610478	999	#	#	4000	122060	934	148755941	450	#	#
1791	112482	273	130952844	473	#	#	3896	63194	210	139673673	210	#	#	4001	119074	772	148875016	222	#	#
1792	112481	447	131052462	654	#	#	3897	62491	688	1397464	898	#	#	4002	116901	203	148991917	426	#	#
1793	111898	123	13117224	642	#	#	3898	61907	139	139798072	036	#	#	4003	114718	091	149106635	377	#	#
1794	110897	239	131288121	281	#	#	3899	61467	662	139859339	698	#	#	4004	112526	859	149219623	575	#	#
1795	109822	461	131397943	743	#	#	3900	61197	209	139920736	907	#	#	4005	110266	732	149329429	107	#	#
1796	108627	156	131506570	899	#	#	3901	60996	791	139981733	698	#	#	4006	107954	942	149437384	049	#	#
1797	107248	588	131613819	487	#	#	3902	60852	364	14004586	062	#	#	4007	105567	593	149542951	642	#	#
1798	105060	002	131719425	489	#	#	3903	60658	244	140103244	305	#	#	4008	103170	274	149646121	978	#	#
1799	103711	517	131823137	005	#	#	3904	60380	485	140163624	790	#	#	4009	100906	663	149747028	515	#	#
1800	101725	989	131924862	994	#	#	3905	60140	245	140223735	035	#	#	4010	98689	131	149845717	709	#	#
1801	99789	471	132024625	654	#	#	3906	60017	258	140283852	293	#	#	4011	96530	149	149942247	859	#	#
1802	97850	407	132124502	492	#	#	3907	60567	106	140344419	399	#	#	4012	94440	423	150036688	281	#	#
1803	95899	240	132218402	212	#	#	3908	61076	832	140405496	231	#	#	4013	92409	450	150129097	371	#	#
1804	93988	803	132312301	014	#	#	3909	61425	252	140466921	583	#	#	4014	90493	657	150219591	988	#	#
1805	91865	040	132404166	055	#	#	3910	61736	849	140528558	433	#	#	4015	88597	008	150308188	397	#	#
1806	90009	442	132494175	497	#	#	3911	62100	764	14059018	196	#	#	4016	86633	381	15039821	780	#	#
1807	88259	291	132582434	790	#	#	3912	62880	197	140653648	393	#	#	4017	84618	723	150479440	502	#	#
1808	86472	020	132668906	810	#	#	3913	63966	266	140717614	659	#	#	4018	82596	040	150562036	543	#	#
1809	84581	648	132753488	458	#	#	3914	65526	779	140783141	378	#	#	4019	80596	097	150642632	640	#	#
1810	82591	810	132836080	268	#	#	3915	67286	897	140850428	334	#	#	4020	78676	493	150721399	153	#	#
1811	80522	634	132916602	902	#	#	3916	68850	921	140919278	517	#	#	4021	77056	368	150798365	351	#	#
1812	78424	625	132994027	527	#	#	3917	70669	273	14098947	791	#	#	4022	75667	853	150874033	504	#	#
1813	76437	751	133071465	258	#	#	3918	73517	384	141063465	174	#	#	4023	74499	448	150948443	307	#	#
1814	74703	509	133146168	767	#	#	3919	77306	019	141140771	194	#	#	4024	73252	255	151021695	856	#	#
1815	73170	265	133223932	332	#	#	3920	81763	294	141222348	488	#	#	4025	72108	571	151093804	427	#	#
1816	7149	205	133301088	237	#	#	3921	83389	904	141310924	391	#	#	4026	71007	058	151164811	485	#	#
1817	70382	973	133361471	210	#	#	3922	85389	618	141406314	010	#	#	4027	70019	622	151234831	107	#	#
1818	68989	966	133430461	177	#	#	3923	101499	111	141507811	321	#	#	4028	69139	613	151303970	721	#	#
1819	67632	341	133498095	338	#	#	3924	107177	229	141619935	550	#	#	4029	68227	893	151372496	280	#	#
1820	66310	261	133566027	779	#	#	3925	111648	921	141726359	471	#	#	4030	68282	093	151444078	063	#	#
1821	65140	884	133629586	163	#	#	3926	114845	471	141841484	942	#	#	4031	68249	509	151509030	171	#	#
1822	63971	844	133693558	007	#	#	3927	117063	315	141958548	255	#	#	4032	68241	080	151577271	252	#	#
1823	62873	484	133756431	491	#	#	3928	119066	181	142077614	436	#	#	4033	68175	283	151643446	534	#	#
1824	61860	387	133819579	779	#	#	3929	120295	945	142199539	79	#	#	4034	68147	492	151713549	363	#	#
1825	60953	365	133879245	244	#	#	3930	120815	600	142318225	779	#	#	4035	68016	252	151781565	940	#	#
1826	60234	723	133939469	967	#	#	3931	121136	829	142439862	608	#	#	4036	68381	104	151849947	748	#	#
1827	59700	536	133999170	503	#	#	3932	121331	155	142561193	763	#	#	4037	69339	704	151919286	044	#	#
1828	59203	208	134058373	713	#	#	3933	120848	608	142682042	372	#	#	4038	70263	763	151989650	511	#	#
1829	58610	773	134117044	485	#	#	3934	119775	458	142801874	441	#	#	4039	71619	705	152061270	211	#	#
1830	58103	938	134175148	416	#	#	3935	118258	382	142920075	823	#	#	4040	72988	407	15213258	621	#	#
1831	57495	190	134232643	607	#	#	3936	116556	002	143036631	825	#	#	4041	74853	064	152209111	685	#	#
1832	56870	408	134289514	015	#	#	3937	114953	293	143151585	118	#	#	4042	76073	465	152287185	157	#	#
1833	56349	900	134348303	915	#	#	3938	113095	997	143264678	214	#	#	4043	80914	366	152360999	516	#	#
1834	55975	410	134406204	325	#	#	3939	110954	236	143386245	51	#	#	4044	84379	052	152436278	568	#	#
1835	55670	269	134457899	594	#	#	3940	108594	217	143482266	668	#	#	4045	105538	627	152508017	194	#	#
1836	54262	102	134512935	697	#	#	3941	106094	639	143590321	607	#	#	4046	114856	766	15262873	961	#	#
1837	53287	196	134563222	893	#	#	3942	103650	425	143693972	032	#	#	4047	122580	722	152805454	682	#	#
1838	52538	510	134623561	202	#	#	3943	101347	484	143795716	416	#	#	4048	129137	128	152934911	817	#	#
1839	51850	908	134684202	112	#	#	3944	99032	047	14394351	563	#	#	4049	13401	976	153080605	766	#	#
1840	50904	445	134745246	557	#	#	3945	96758	893	143991110	456	#	#	4050	137575	573	1532061	360	#	#
1841	50230	335	134791476	892	#	#	3946	94616	839	144085727	295	#	#	4051	140302	211	153346484	577	#	#
1842	49446	483	134847923	375	#	#	3947	92885	477	144178610	791	#	#	4052	141961	116	153488445	681	#	#
1843	48727	327	134904650	702	#	#	3948	91293	588	144269903	660	#	#	4053	142627	767	153631088	394	#	#
1844	48104	218	134962040	921	#	#	3949	89545	103	144359448	853	#	#	4054	143056	669	15377444	465	#	#
1845	47351	188	135020312	108	#	#	3950	87629	558	144447078	411	#	#	4055	142842	792	153916987	255	#	#
1846	46714	061	135079459	170	#	#	3951	85609	428	144532687	839	#	#	4056	141742	429	154058729	684	#	#
1847	46039	598	135139498	767	#	#	3952	83528	109	144616215	948	#	#	4057	139901	181	154198330	865	#	#
1848	45416	357	135209615	124	#	#	3953	81452	239	144697661	206	#	#	4058	137562	818	154316393	884	#	#
1849	44732	868	135266297	983	#	#	3954	79407	187	144777158	394	#	#	4059	135064	316	154471238	000	#	#
1850	44059	923	135327497	916	#	#	3955	77698	238	144854856	632	#	#	4060	132504	536	154603762	536	#	#
1851	43435	969	135395151	884	#	#	3956	75956	237	144930812	9									

APENDICE 5

A continuación se incluye el código fuente del programa auxiliar para presentación de las gráficas:

```
*****
NOMBRE      grafdata.c
AUTOR       Brenda Eugenia Olivos Fuentes
DESCRIPCION Este programa es auxiliar para la presentacion y/o impresion de
             las graficas asociadas tanto con los datos de adquisicion como
             con los datos de proceso. Necesita que los archivos de entrada
             lleven los nombres adecuados y contengan los datos del tipo
             requerido y en el orden en que los arrojan los programas de
             adquisicion y proceso.
*****/
#include <stdio.h>
main(argc,argv)
int argc;
char *argv[];

{
  FILE *fpgnu, *fptex1, *fptex2;
  int i, j, equis[9], yes[9];
  char *titulo[9];

  titulo[0]=" Periodo Vs Numero de pulsos ";
  titulo[1]=" Periodo Vs tiempo ";
  titulo[2]=" Inverso del periodo Vs numero de pulsos ";
  titulo[3]=" Inverso del periodo Vs tiempo ";
  titulo[4]=" Numero de pulsos Vs Tiempo ";
  titulo[5]=" Periodo suavizado Vs pulsos ";
  titulo[6]=" Periodo suavizado Vs tiempo ";
  titulo[7]=" Numero de pulsos ";
  titulo[8]=" Tiempo entre pulsos ";

  equis[0]=equis[2]=equis[5]=equis[7]=equis[8]=1;
  equis[1]=equis[3]=equis[6]=equis[4]=3;
  yes[0]=yes[1]=yes[5]=yes[6]=2;
  yes[2]=yes[3]=yes[7]=4;
  yes[4]=1;
  yes[8]=5;

  if(argc!=2){
    fprintf(stderr,"USE: grafdata.exe nomarch \n");
    exit(1);
  }
}
```

```

fpgnu=fopen("grafdata.gnu","w");
if(fpgnu==NULL){
    fprintf(stderr,"ERROR AL ABRIR EL ARCHIVO grafdata.gnu!!!!\n");
    exit(1);
}

/** se genera grafdata.gnu con las directrices de graficacion **/
fprintf(fpgnu,"set term latex\nset nogrid\n");
for(i=0;i<9;i++){
    fprintf(fpgnu,"set output 'GRAFICAS/graf0%d.tex"\n",i+1);
    fprintf(fpgnu,"set title '%s'\n",titulo[i]);
    if(i<5)
        fprintf(fpgnu,"plot 'adq%s.dat' using %d:%d with lines\n",argv[1],equis[i],yes[i]);
    else
        fprintf(fpgnu,"plot 'proc%s.dat' using %d:%d with lines\n",argv[1],equis[i],yes[i]);
}
fclose(fpgnu);

/**se llama al gnuplot para que ejecute las directrices de grafdata.gnu**/
system("gnuplot grafdata.gnu");

/** para cada una de las graficas se genera grafdata.tex y se llama a td150**/
for(j=1;j<=9;j++){
    fptex1=fopen("grafdata.tex","w");
    if(fptex1==NULL){
        fprintf(stderr,"ERROR AL ABRIR EL ARCHIVO grafdata.tex!!!!\n");
        exit(1);
    }
    fprintf(fptex1,"\\documentstyle[12pt]{article}\n");
    fprintf(fptex1,"\\pagestyle{empty}\n");
    fprintf(fptex1,"\\topmargin=0in\n");
    fprintf(fptex1,"\\headheight=0in\n");
    fprintf(fptex1,"\\headsep=0in\n");
    fprintf(fptex1,"\\oddsidemargin=0in\n");
    fprintf(fptex1,"\\textwidth=15.5cm\n");
    fprintf(fptex1,"\\textheight=22.5cm\n");
    fprintf(fptex1,"\\begin{document}\n");
    fprintf(fptex1,"\\setcounter{figure}{0}\n");
    fprintf(fptex1,"\\begin{figure}[h] \\input{./GRAFICAS/graf0%d.tex} \n",j);
    fprintf(fptex1,"\\end{figure}\n");
    fprintf(fptex1,"\\end{document}\n");
    system("pwd");
    fclose(fptex1);

    system("td150 grafdata");
}
}

```

APENDICE 6

Este apéndice se ha dividido en tres partes, de la siguiente manera:

A. Señal Simulada, B. Transición al Ruido y C. Ruido.

A. Señal Simulada.

A partir de dos generadores de señales, se obtuvo una señal eléctrica de frecuencia modulada por voltaje; capaz de ser descrita por la ecuación (4.2). Así pues, se logró simular la señal interferométrica de un movimiento armónico, lo que facilitó la evaluación de los programas para la adquisición y proceso de datos. En este apéndice se presentan las gráficas correspondientes a esa señal simulada.

Obsérvese el parecido de estas gráficas (figuras A1, A2, A3, A4, A5, A6, A7, A8 y A9) con las correspondientes a la señal proveniente del interferómetro (figuras 4.23, 4.24, 4.25, 4.26, 4.27, 4.29, 4.30, 4.31 y 4.32). Nótese sin embargo, que las gráficas que a continuación se presentan, son más limpias y simulan un movimiento armónico no amortiguado.

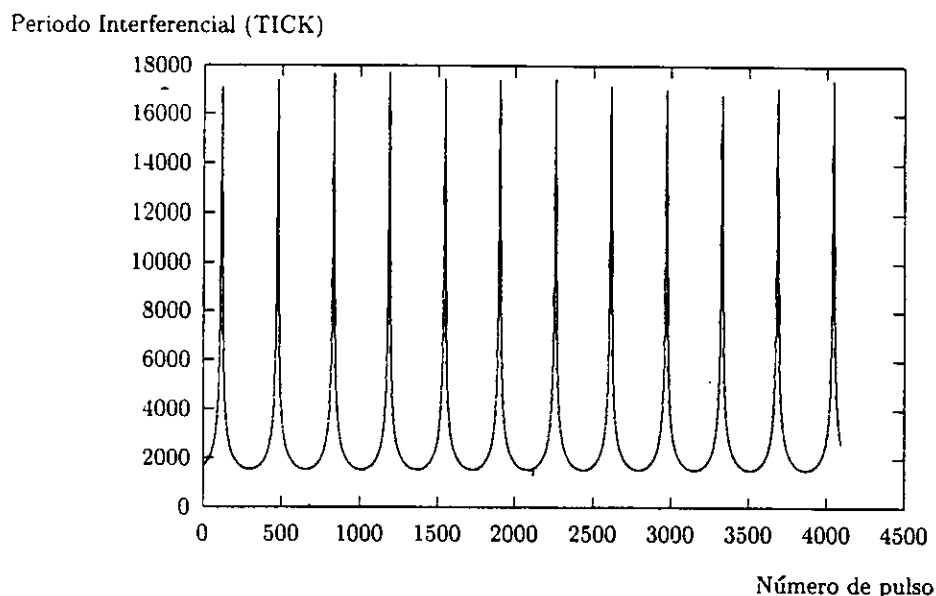


Fig. A1. Gráfica del periodo interferencial simulado contra el número de pulso.

Periodo Interferencial (TICK)

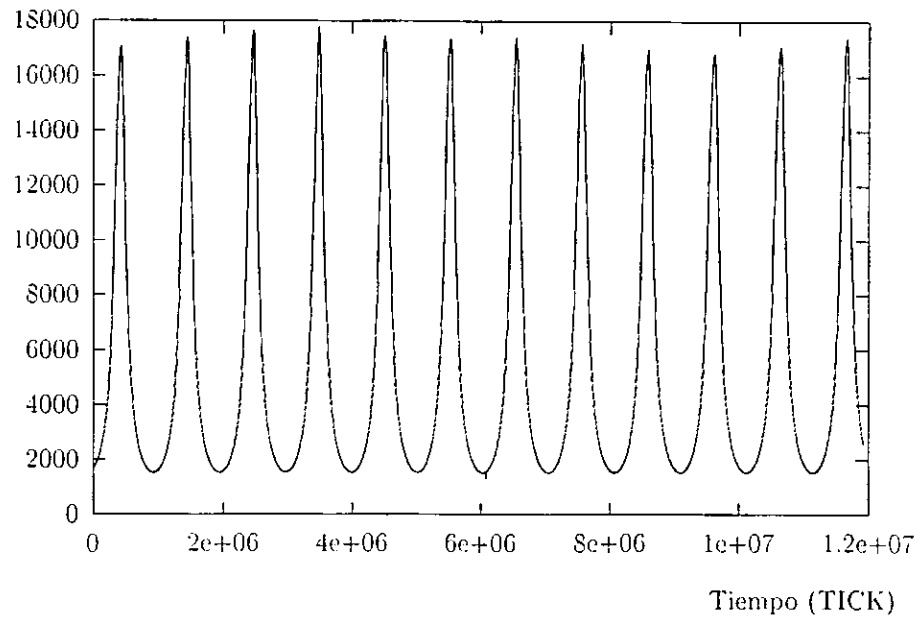


Fig. A2. Gráfica del periodo interferencial simulado contra el tiempo.

Frecuencia Interferencial (1/TICK)

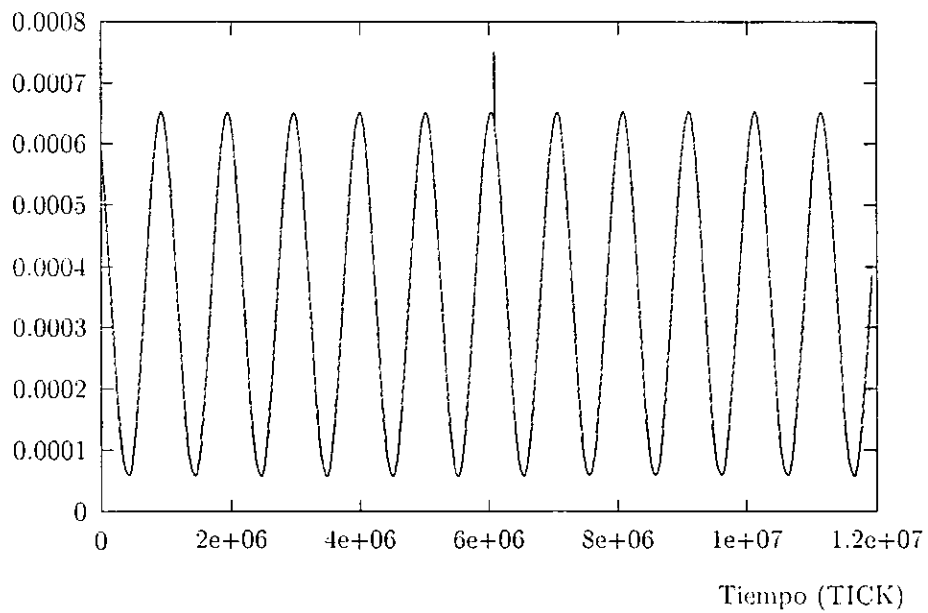


Fig. A3. Gráfica de la frecuencia interferencial simulada contra el tiempo.

Frecuencia Interferencial (1/TICK)

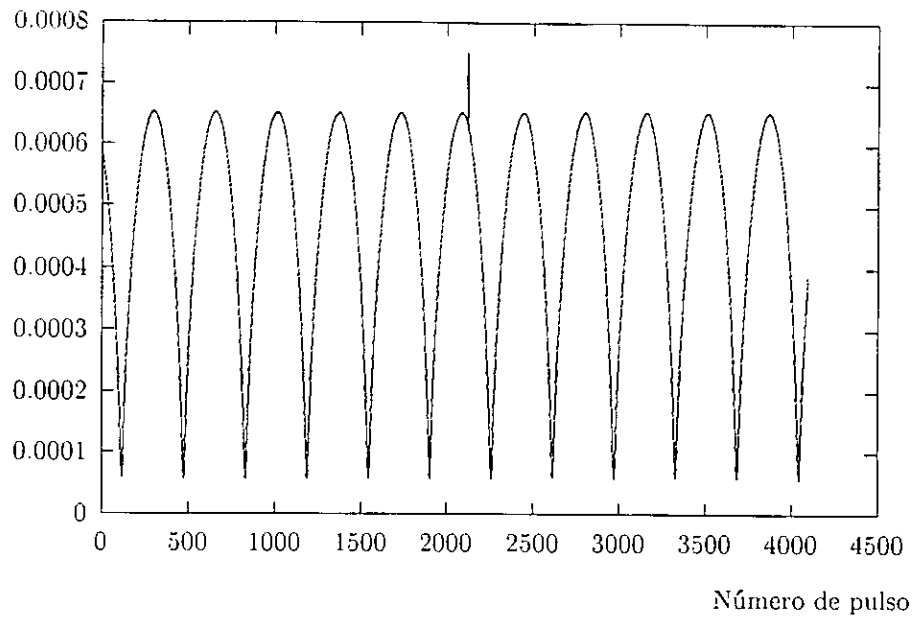


Fig. A4. Gráfica de la frecuencia interferencial simulada contra el número de pulso.

Desplazamiento ($\lambda/2$)

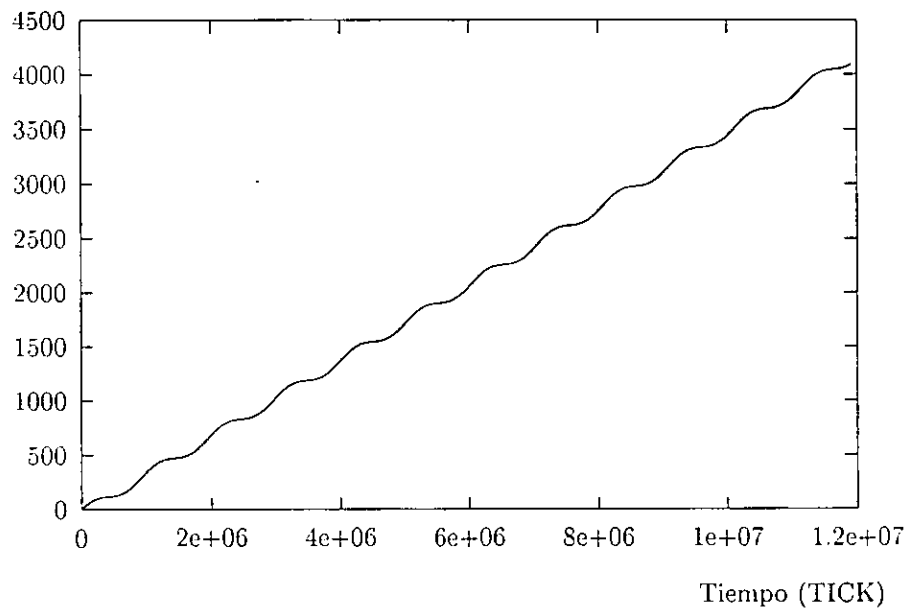


Fig. A5. Gráfica del desplazamiento del oscilador mecánico simulado contra el tiempo.

Periodo Interferencial Suavizado (TICK)

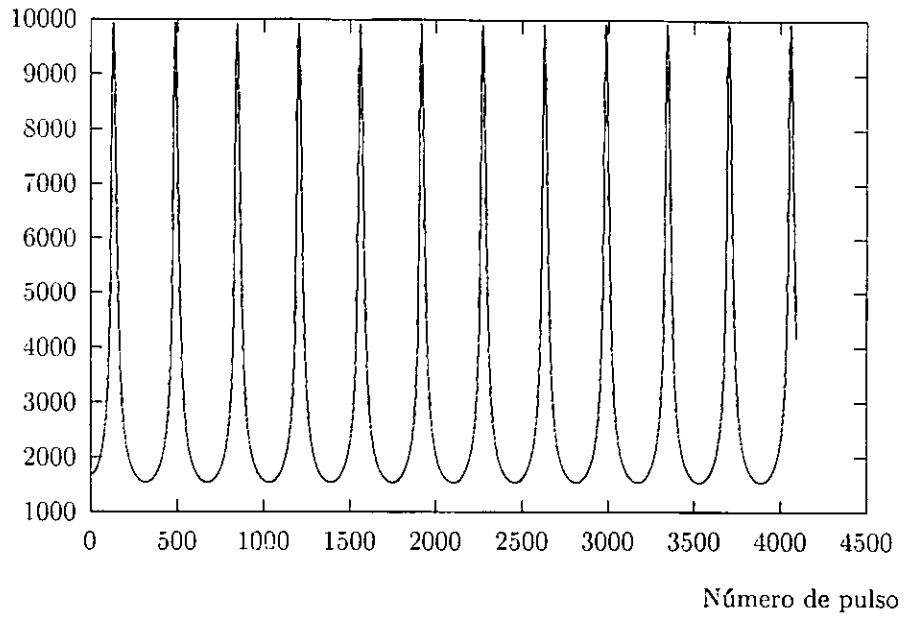


Fig. A6. Gráfica del periodo interferencial simulado, una vez suavizado, contra el número de pulso.

Periodo Interferencial Suavizado (TICK)

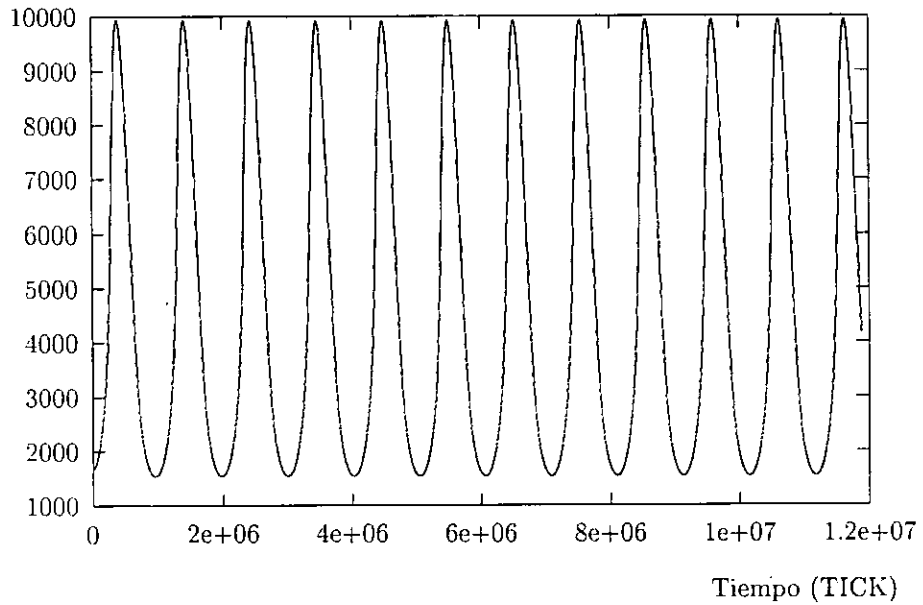


Fig. A7. Gráfica del periodo interferencial simulado, una vez suavizado, contra el tiempo.

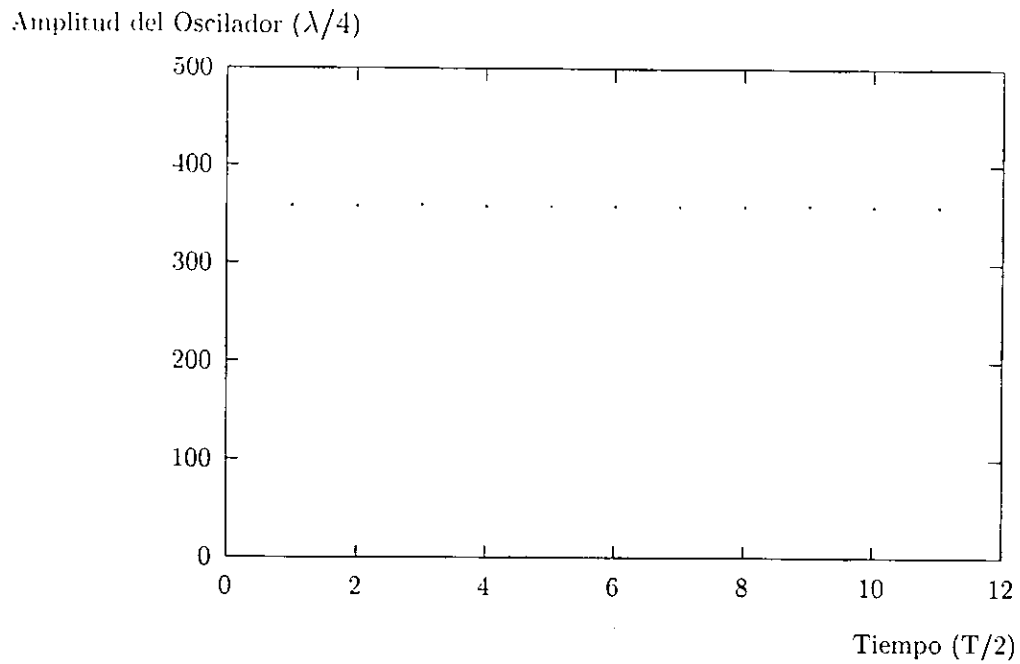


Fig. A8. Gráfica de la amplitud del oscilador simulado contra el tiempo.

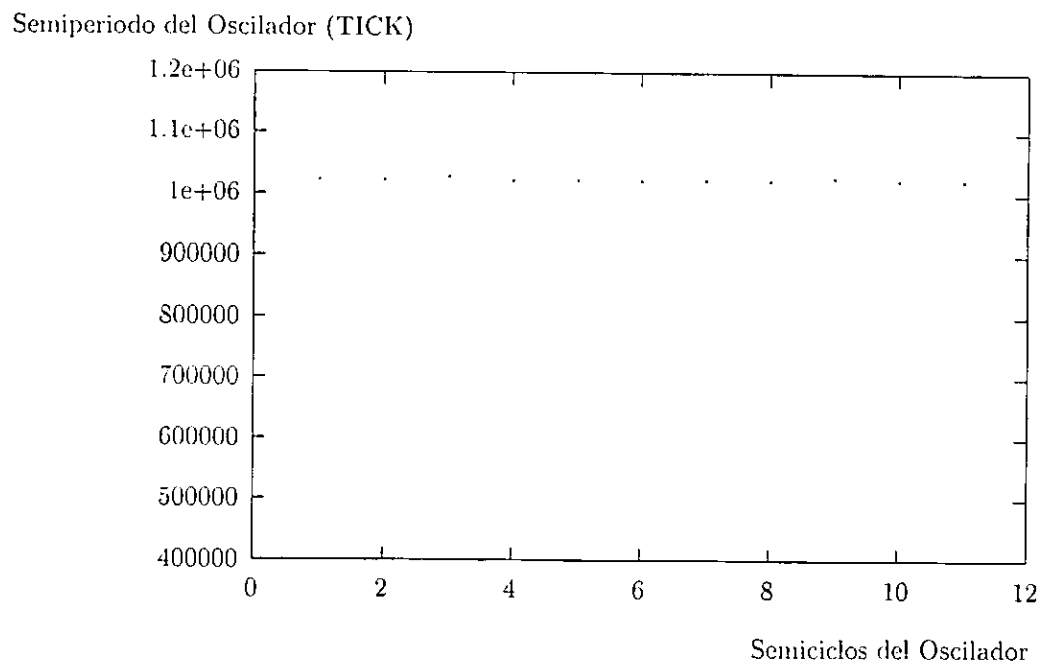


Fig. A9. Gráfica del semiperiodo ($T/2$) del oscilador simulado.

En las figuras A1 y A2 es posible observar máximos correspondientes a la velocidad igual a cero del oscilador simulado. La distancia entre máximos de la fig. A1 es constante; ello muestra que se está simulando un movimiento armónico no amortiguado. Por esa misma razón, el valor de los mínimos también aparece como constante (tampoco se amortigua la velocidad). Ambas gráficas muestran sus bondades para identificar las “regiones de mínima densidad” las cuales se manifiestan como máximos pronunciados.

La figura A2 constituye una prueba del éxito de la simulación. En ella se observa la isocronía del oscilador simulado.

Las figuras A3 y A4 muestran la señal senoidal rectificada de la velocidad del oscilador armónico que está simulándose. La velocidad del movimiento armónico, no se amortigua ni respecto al tiempo ni respecto a la distancia recorrida. Así pues, en la fig. A5 se pueden notar las ondulaciones de la curva, las cuales muestran las variaciones en velocidad del oscilador. El desplazamiento total del oscilador simulado tiene una envolvente lineal lo que es también claro indicador de que el movimiento no es amortiguado.

Ahora bien, las figuras A6 y A7, son las gráficas de los datos suavizados. Puede comprobarse que la suavización no altera las propiedades de la curva de interés; únicamente elimina el ruido pequeño, lo cual es determinante para el cálculo de la amplitud y frecuencia del oscilador.

Finalmente, las figuras A8 y A9 muestran la amplitud del oscilador y su semiperíodo respectivamente, valores que son resultado del proceso de los datos. Obsérvese que la dispersión es mínima.

B. Transición al ruido.

En este apartado se presentan las gráficas correspondientes a la señal interferométrica en el momento de la transición de movimiento armónico a ruido.

Obsérvese en las figuras B1, B2, B3 y B4, los efectos del amortiguamiento en las zonas donde es claro el movimiento armónico. Nótese en la figura B5 que las variaciones en la velocidad son cada vez menores y la envolvente de la curva pasa de ser una exponencial a una línea recta. Observense también los efectos de la suavización en las gráficas B6 y B7, especialmente en la

zona de ruido, así como sus implicaciones en el cálculo de la amplitud y semiperiodo del oscilador mostrados en las figuras B8 y B9.

Periodo Interferencial (TICK)

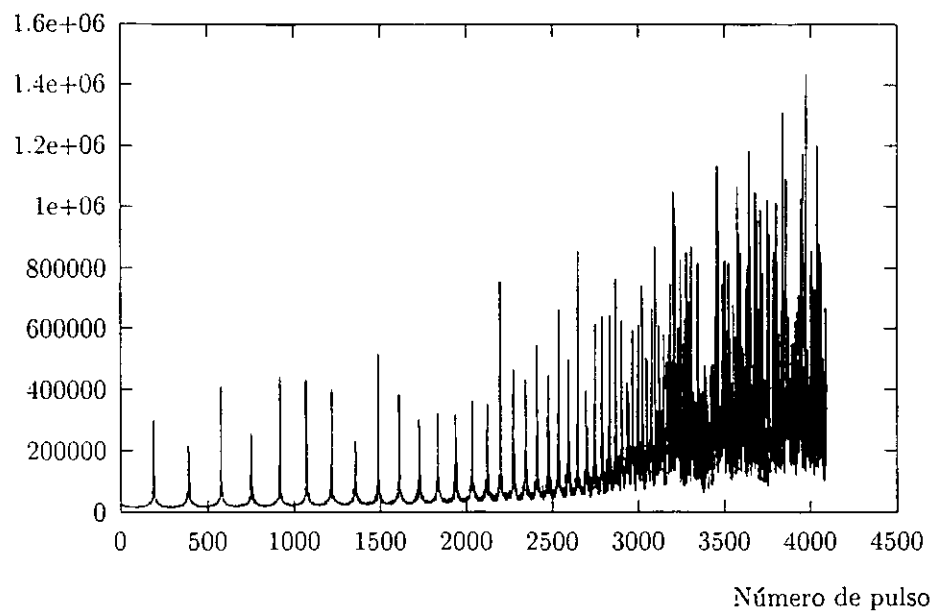


Fig. B1. Gráfica del periodo interferencial contra el número de pulso.

Periodo Interferencial (TICK)

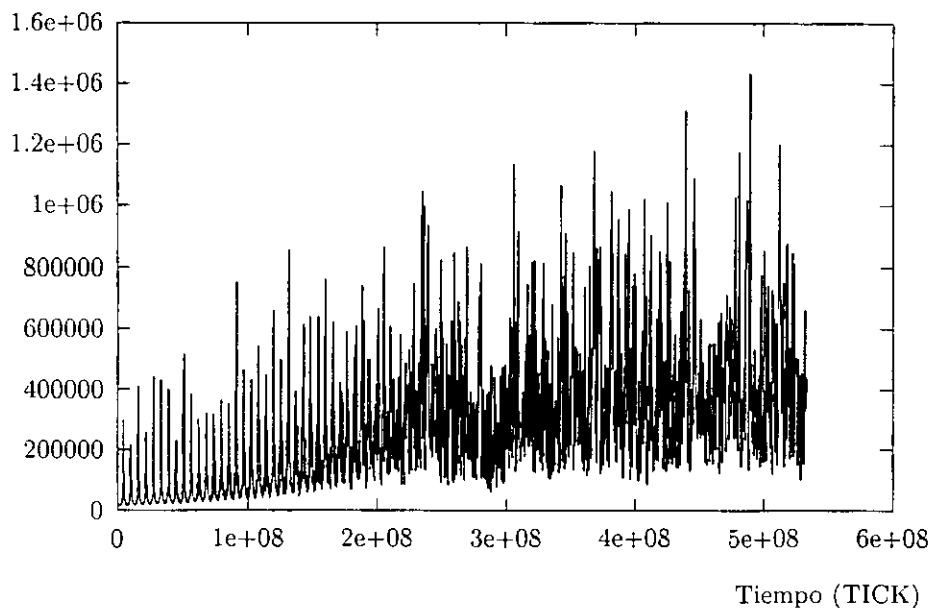


Fig. B2. Gráfica del periodo interferencial contra el tiempo.

Frecuencia Interferencial (1/TICK)

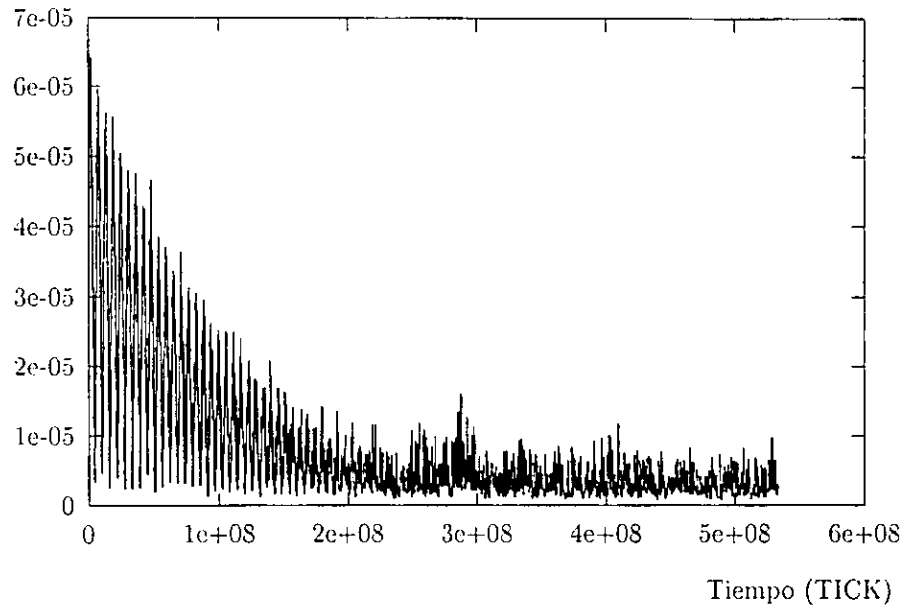


Fig. B3. Gráfica de la frecuencia interferencial contra el tiempo.

Frecuencia Interferencial (1/TICK)

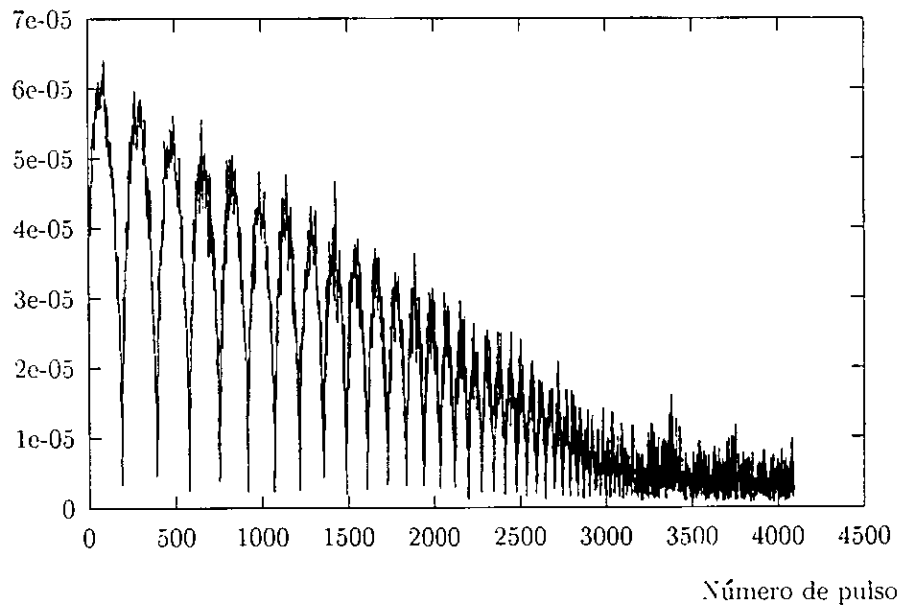


Fig. B4. Gráfica de la frecuencia interferencial contra el número de pulso.

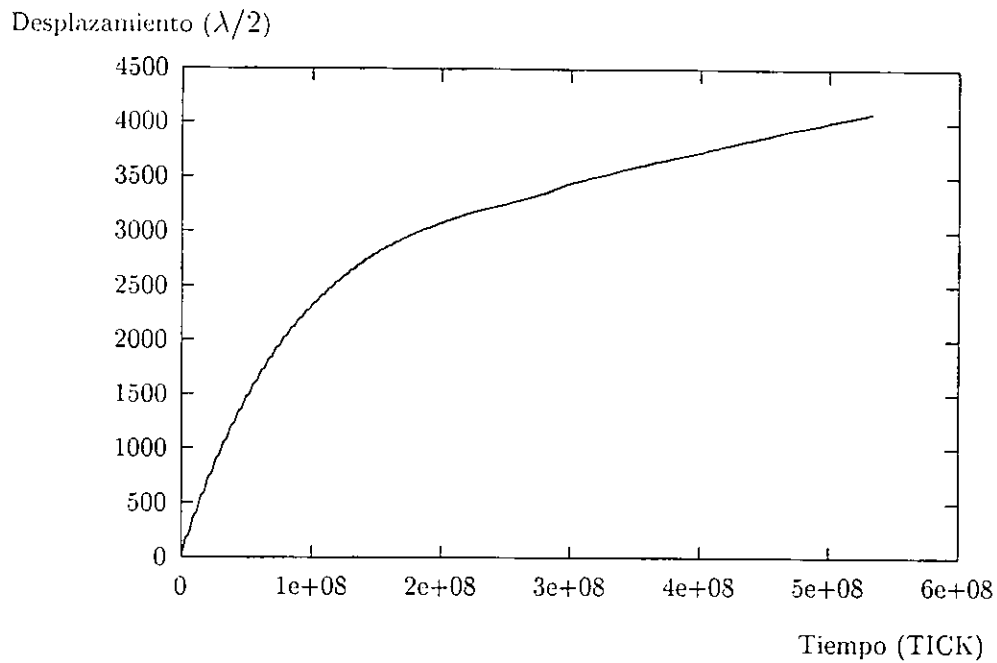


Fig. B5. Gráfica del desplazamiento del oscilador mecánico contra el tiempo.

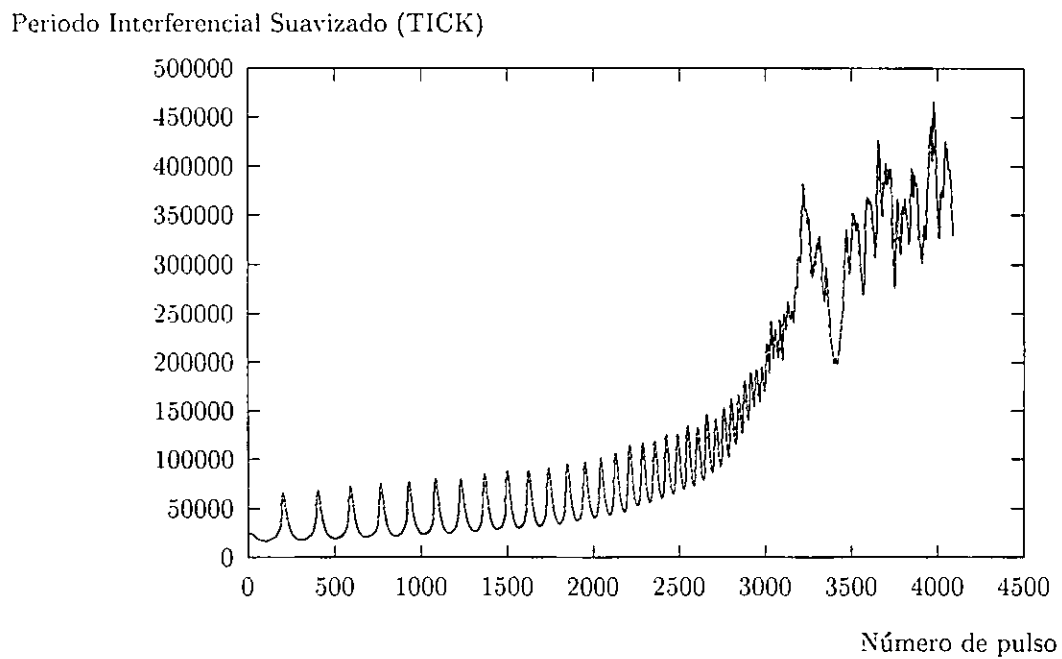


Fig. B6. Gráfica del periodo interferencial suavizado contra el número de pulso.

Periodo Interferencial Suavizado (TICK)

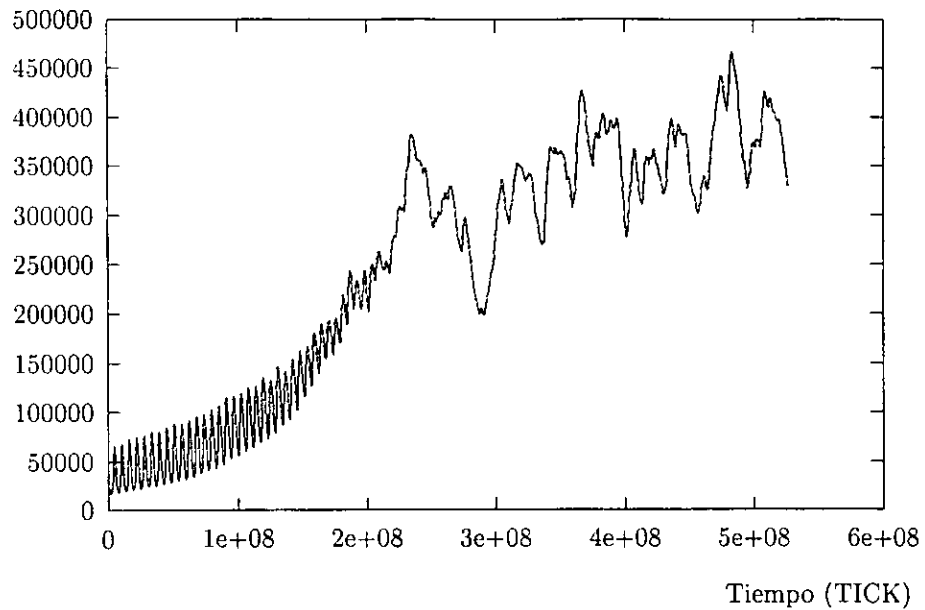


Fig. B7. Gráfica del periodo interferencial suavizado contra el tiempo.

Amplitud del Oscilador ($\lambda/4$)

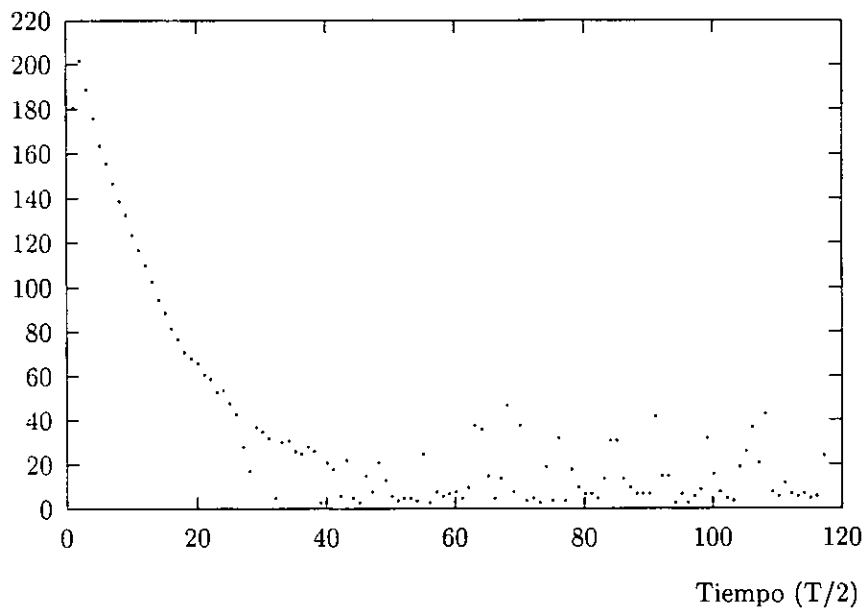


Fig. B8. Gráfica de la amplitud del oscilador contra el tiempo.

Semiperiodo del Oscilador (TICK)

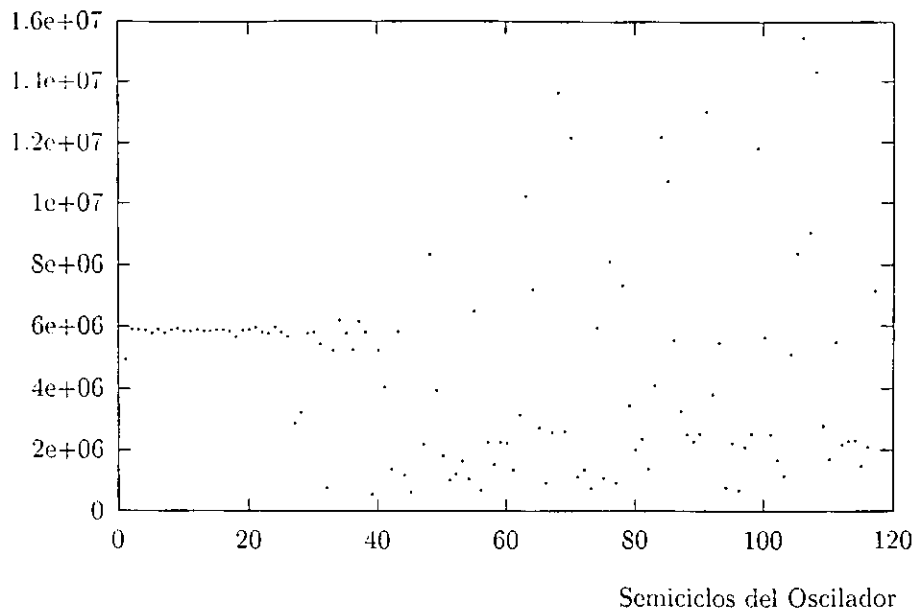


Fig. B9. Gráfica del semiperiodo (T/2) del oscilador.

C. Ruido.

En este apartado se presentan las gráficas correspondientes a una señal de ruido interferométrico, el cual, como se dijo en su momento, corresponde al registro de la microsismicidad presente en el laboratorio. Dado que los programas para la adquisición y proceso de datos se diseñaron especialmente para el análisis de un movimiento armónico, aplicarlos a ruido interferométrico carece de sentido; sin embargo se presentan a continuación la gráficas del periodo interferencial contra número de pulso (fig. C1), frecuencia interferencial contra tiempo (fig. C2) y la del desplazamiento total (fig. C3).

La figura C1 es una gráfica de los datos tal como son arrojados por el interferómetro: es la gráfica del tiempo que le lleva a la esquina de cubo desplazarse distancias iguales a $\lambda/2$.

La figura C2 es de particular interés debido a que es proporcional a la velocidad del móvil (en este caso el móvil es la esquina de cubo); obsérvese el rango en que se mueve esta variable.

Por otro lado, en la figura C3 es posible observar que el desplazamiento total del móvil describe una línea recta casi perfecta y que las variaciones de velocidad que se harían presentes como ondulaciones de la curva, son casi imperceptibles.

Periodo Interferencial (TICK)

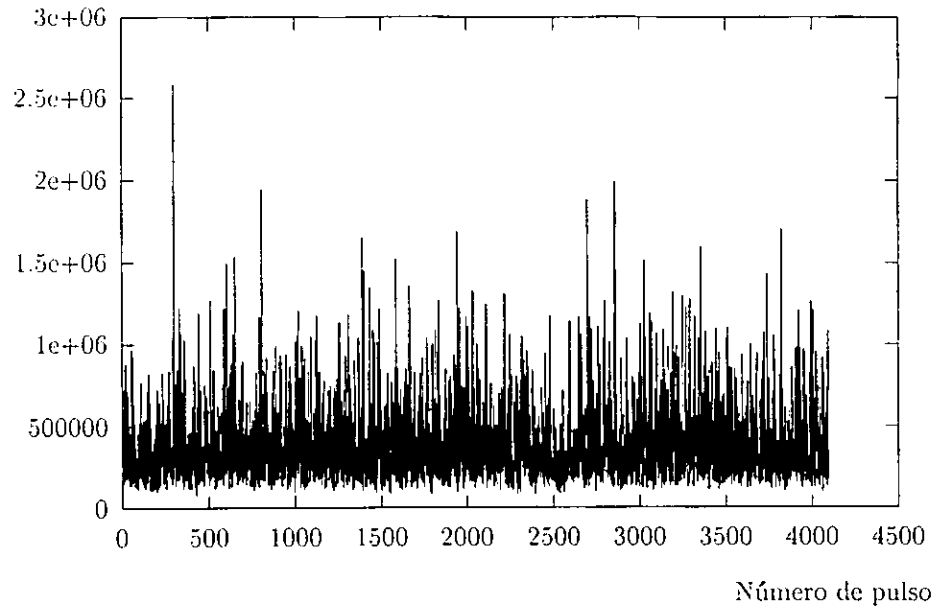


Fig. C1. Gráfica del periodo interferencial contra el número de pulso. Los puntos experimentales se han unido con rectas para mayor claridad.

Frecuencia Interferencial (1/TICK)

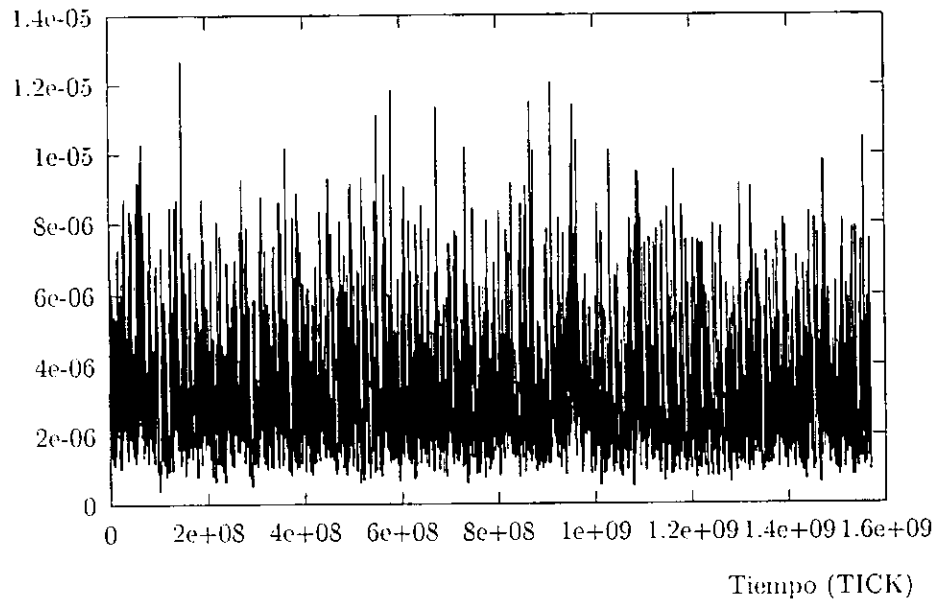


Fig. C2. Gráfica de la frecuencia interferencial contra el tiempo. Los puntos experimentales se han unido con rectas para mayor claridad.

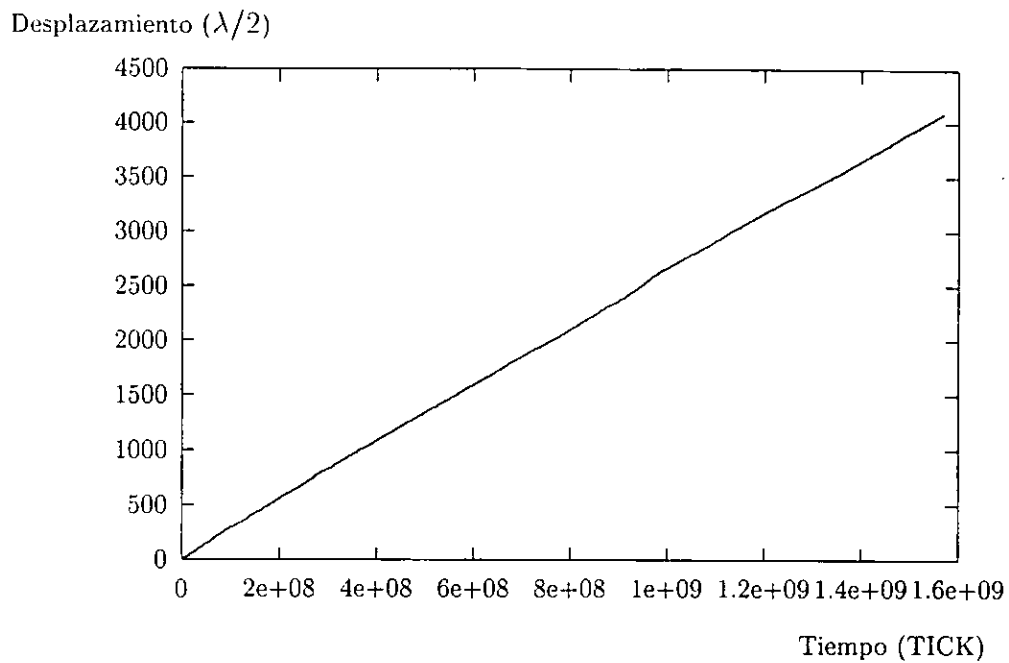


Fig. C3. Gráfica del desplazamiento total del móvil contra el tiempo. Los puntos experimentales se han unido con rectas para mayor claridad.

REFERENCIAS

- [1.1] Morales, A; De Urquijo, J; Mendoza, A. Vertical high-precision Michelson wavemeter. Review of Scientifics Instruments. USA, January 1993. pp 76-81.
- [2.1] The Institute of Electrical and Electronics Engineers. CAMAC Instrumentation and Interface Standards. IEEE, Ed. John Wiley & Sons. USA, 1982.
- [2.2] The Institute of Electrical and Electronics Engineers. Standard Modular Instrumentation and Digital Interface System (CAMAC). IEEE, Ed. John Wiley & Sons. USA, 1982.
- [2.3] Kinetic Systems Corporation. CAMAC 1996-1997 Catalog. Committed to Meeting your Needs. USA, 1997.
- [2.4] Kinetic Systems Corporation. VXI Bus 1995 Product Catalog. VXI Solutions as Easy as 1 2 3. USA, 1995.
También Cf.
Kinetic Systems Corporation. VXI Bus Innovations!. USA, 1996.
- [2.5] Kinetic Systems Corporation. Model 1570. Powered CAMAC Crate. Instruction Manual. USA, 1982.
- [2.6] Kinetic Systems Corporation. Model 3989. RS-232 Crate Controller. Instruction Manual. USA, 1986.

- [2.7] Kinetic Systems Corporation. Model 3982-Z1B/Z2B/Z3B. List Sequencing Crate Controller. Instruction Manual. USA, 1991.
- [2.8] Kinetic Systems Corporation. Model 3664-L1A. Time Interval Counter. Instruction Manual. USA, 1987.
- [3.1] Hewlett Packard. HP 9000, Series 800. Installing Peripherals. USA, 1992.
- [3.2] Hewlett Packard. HP 9000 Computers. How HP-UX Works: Concepts for the System Administrator. USA, 1992.
- [3.3] Hewlett Packard. HP 9000 Computers. HP_UX Reference. USA, 1992. Véase “termio(7)”.
- [3.4] Hewlett Packard. HP 9000 Computers. HP_UX Reference. USA, 1992. Véase “master(4)”.
- [3.5] Hewlett Packard. HP 9000 Computers. HP_UX Reference. USA, 1992. Véase “mknod(1M)”.
- [3.6] Hewlett Packard. HP 9000 Computers. HP_UX Reference. USA, 1992. Véase “ioscan(1M)”.
- [3.7] The Institute of Electrical and Electronics Engineers. Standard Subroutines for CAMAC. IEEE. Ed. John Wiley & Sons. USA, 1982.
- [4.1] László, T; László, M. A Simple and Accurate Contactless Frequency and Amplitude Meter. Review of Scientific Instruments. USA, February 1997. pp 1303-1306.
- [4.2] Hillier, F; Lieberman, G. Introducción a la Investigación de Operaciones. Ed. Mc. Graw Hill. México, 1997. Véase Capítulo 18. Pronósticos.

BIBLIOGRAFIA

1. Abrahams, P; Larson, B. UNIX for the Impatient. Ed. Addison Wesley. USA, 1996.
2. Barrett, M; Wagner, C. C and UNIX. Tools for Software Design. Ed. John Wiley & Sons. USA, 1996.
3. Deitel, H. Sistemas Operativos. Ed. Addison-Wesley Iberoamericana. USA, 1993.
4. Giancoli, D. Física General. Ed. Prentice Hall Hispanoamericana, S.A. México, 1988, Vol. II.
5. Hewlett Packard. HP 9000 Computers. How HP-UX Works: Concepts for the System Administrator. USA, 1992.
6. Hewlett Packard. HP 9000 Computers, HP-UX Reference. USA, 1992.
7. Hewlett Packard. HP 9000 Computers, Programming on HP-UX. USA, 1992.
8. Hewlett Packard. HP 9000, Series 700. Model 712 Owner's Guide. USA, 1993.
9. Hewlett Packard. HP 9000, Series 800 Computers. System Administration Tasks. USA, 1992.
10. Hewlett Packard. HP 9000, Series 800. Installing Peripherals. USA, 1992.
11. Hillier, F; Lieberman, G. Introducción a la Investigación de Operaciones. Ed. Mc. Graw Hill. México, 1997. Véase Capítulo 18. Pronósticos.
12. Jaffe, B. Michelson y la Velocidad de la Luz. Editorial Universitaria de Buenos Aires. Argentina, 1963.

13. Kernighan, B; Ritchie, D. El Lenguaje de Programación C. Ed. Prentice Hall Hispanoamericana. México, 1991.
14. Kinetic Systems Corporation. CAMAC 1996-1997 Catalog. Committed to Meeting your Needs. USA, 1997.
15. Kinetic Systems Corporation. Model 1570. Powered CAMAC Crate. Instruction Manual. USA. 1982.
16. Kinetic Systems Corporation. Model 3664-L1A. Time Interval Counter. Instruction Manual. USA. 1987.
17. Kinetic Systems Corporation. Model 3982-Z1B/Z2B/Z3B. List Sequencing Crate Controller. Instruction Manual. USA, 1991.
18. Kinetic Systems Corporation. Model 3989. RS-232 Crate Controller. Instruction Manual. USA. 1986.
19. Kinetic Systems Corporation. VXI Bus 1995 Product Catalog. VXI Solutions as Easy as 1 2 3. USA, 1995.
20. Kinetic Systems Corporation. VXI Bus Innovations!. USA, 1996.
21. László, T; László, M. A Simple and Accurate Contactless Frequency and Amplitude Meter. Review of Scientific Instruments. USA, February 1997. pp 1303-1306.
22. Morales, A; De Urquijo, J; Mendoza, A. Vertical high-precision Michelson wavemeter. Review of Scientifics Instruments. USA, January 1993. pp 76-81.
23. Murray, R; Spiegel. Estadística. Ed. Mc. Graw Hill. España, 1991.
24. Pajari. G. Writing UNIX Device Drivers. Ed. Addison-Wesley. USA, 1992.

25. Prawda, J. Métodos y Modelos de Investigación de Operaciones. Ed. LIMUSA. México, 1993. Véase Vol. II. Modelos Estocásticos.
26. Resnick, Halliday and Krane. FISICA. Ed. CECSA. USA, 1994.
27. Rochkind, M. Advanced UNIX Programming. Ed. Prentice Hall. USA, 1985.
28. Steel, W. Interferometry. (Cambridge Studies in Modern Optics). Cambridge University Press. Gran Bretaña, 1987.
29. Stollberg, R; Hill, F. FISICA. Fundamentos y Fronteras. Ed. Publicaciones Cultural S.A. México, 1968.
30. The Institute of Electrical and Electronics Engineers. CAMAC Instrumentation and Interface Standards. IEEE, Ed. John Wiley & Sons. USA, 1982.
31. The Institute of Electrical and Electronics Engineers. IEEE Recommended Practice for Block Transfers in CAMAC Systems. IEEE, Ed. John Wiley & Sons. USA, 1982.
32. The Institute of Electrical and Electronics Engineers. IEEE Standard Real Time BASIC for CAMAC. IEEE, Ed. John Wiley & Sons. USA, 1982.
33. The Institute of Electrical and Electronics Engineers. IEEE Standard Serial Highway Interface System (CAMAC). IEEE, Ed. John Wiley & Sons. USA, 1982.
34. The Institute of Electrical and Electronics Engineers. IEEE Standard Subroutines for CAMAC. IEEE, Ed. John Wiley & Sons. USA, 1982.
35. The Institute of Electrical and Electronics Engineers. Standard Modular Instrumentation and Digital Interface System (CAMAC). IEEE, Ed. John Wiley & Sons. USA, 1982.