Universidad Nacional Autónoma de México
Facultad de Ingeniería
División de Ingeniería Eléctrica

Trazador Digital de Nyquist para auxilio didáctico

Tesis que para obtener el título de
Ingeniero en Computación presentan:

Miguel Angel Montaño Medina
Edgar Valentín López García

Director de tesis:

M.I. Antonio Salvá Calleja.

En memoria de
Esperanza Montaño Esquivel

Gracias por la educación, el amor y el tiempo que me dedicaste

Miguel Angel

A mis padres Alejandra y Valentín

Por toda una vida de esfuerzo y sacrificios.

Edgar

Agradecimientos

Quiero agradecer a mis padres por el apoyo que me han proporcionado a lo largo de mis estudios, de igual manera a mi primo Miguel, a mi tía Rosa y a mi abuelita Esperanza. A todos mis amigos del bloque 6 por su amistad y apoyo, en especial a Butanda y a Edgar por haberme ayudado a terminar la carrera. A Keyla, por llenar mi vida de felicidad.

Gracias al M.I. Antonio Salvá Calleja por el tiempo que nos dedicó al dirigir esta tesis. A la UNAM, por la educación que tan generosamente me brindó.

Miguel Angel

Quiero agradecer

A la Facultad de Ingeniería y a mis maestros por todos los conocimientos que me transmitieron. En especial al Maestro Antonio Salvá, director de esta tesis, por su tiempo y dedicación.

A la Universidad, por la inmensa cantidad de oportunidades que me brindo para desarrollarme y por el orgullo de pertenecer a ella.

A Miguel por su dedicación en la elaboración de esta tesis y por su amistad.

A Sandra y Nallely, por compartir una vida juntos en una gran familia.

A mis Padres por el enorme esfuerzo realizado para sacarnos adelante. Sin ustedes yo no seria nada.

A Nora Rodríguez por su compañía. Gracias por llenar de fantasía mi vida.

Edgar

Índice general

Introducción

En la Facultad de Ingeniería de la U.N.A.M. se imparten diversas materias relacionadas con el análisis de circuitos eléctricos, las cuales se complementan con prácticas en el laboratorio. Algunos temas en esas materias requieren el uso de dispositivos para que los alumnos comprendan algunos conceptos con claridad. Por ejemplo, para entender el concepto de frecuencia es muy útil el uso del osciloscopio y del generador de funciones. Una práctica de laboratorio en donde se empleé un instrumento de medición siempre pondrá en claro un concepto teórico visto en las aulas.

Los cursos de control analógico que se imparten en la Facultad de Ingeniería, cubren algunos de los temas relacionados con el análisis de respuesta en frecuencia de sistemas estables, lineales e invariantes en el tiempo (sistemas ELIT) Sin embargo en el laboratorio de control analógico, no existe una práctica dedicada a ilustrar este concepto. El análisis de respuesta en frecuencia y las trazas de Nyquist se explican, generalmente, mediante gráficas en el pizarrón; pero en ocasiones, estos conceptos no quedan claros. Por ello, el contar con un dispositivo capaz de obtener la traza de Nyquist de un sistema ELIT, sería de gran utilidad en la enseñanza del tema.

Además es importante mencionar que la mayoría de las prácticas de laboratorio en donde se llevan a cabo análisis de sistemas, solo comprenden el análisis en el dominio del tiempo. La idea de construir este dispositivo, llamado Trazador Digital de Nyquist (TDN), surge como una propuesta para auxiliar a los alumnos que cursan la materia de control analógico dentro de la Facultad de Ingeniería; a entender mejor el concepto de análisis de respuesta en frecuencia.

Por todo lo anterior, el presente trabajo describe el diseño y funcionamiento de un dispositivo capaz de realizar experimentalmente el análisis de respuesta en frecuencia de los sistemas ELIT obteniendo una representación del mismo mediante la traza polar de Nyquist.

La presente tesis consta de 5 capítulos. En el capítulo 1 se habla del principio de operación del TDN; se tratan los conceptos básicos, así como el principio matemático sobre el que se sustenta este aparato. En el capítulo 2 se describe de forma detallada cada uno de los componentes que conforman al TDN, describiendo su funcionamiento. En el capítulo 3 se habla del software de comunicación, el cual esta conformado por programas elaborados en lenguaje ensamblador para el microcontrolador HC11 que se utilizaron para

llevar a cabo la comunicación entre el TDN y el software de interfaz de usuario. En el capítulo 4 se trata el diseño de la aplicación de interfaz de usuario y en general del sistema completo. El modelado del sistema se realizó mediante el lenguaje unificado de modelado UML, esto como un esfuerzo por documentar de forma profesional una aplicación. En el capítulo 5 se presentan algunos ejemplos de aplicación, a partir de estos ejemplos se muestran los alcances del TDN. Por último, en la sección de apéndices se presenta el listado de los programas de enlace, la documentación de las clases del software de interfaz de usuario y un manual de operación en donde se describe de manera breve la forma de emplear el TDN.

Capítulo 1

Principio de operación

1.1. Introducción

A continuación se presentan algunas definiciones de términos básicos para entender mejor qué es un sistema ELIT. El concepto de análisis de respuesta en frecuencia.

Un sistema es una combinación de componentes que actúan juntos y realizan un objetivo determinado. Por otra parte, un sistema se denomina lineal si se aplica el principio de superposición. El cual establece que la respuesta producida por la aplicación simultánea de dos funciones de entradas diferentes es la suma de las dos respuestas individuales. Por tanto, para el sistema lineal, la respuesta a varias entradas se calcula tratando una entrada a la vez y sumando los resultados. Este principio permite desarrollar soluciones complicadas a partir de soluciones simples. Si en un análisis experimental de un sistema son proporcionales la causa y el efecto, lo que implica aplicar el principio de superposición, el sistema se considera lineal.

Los sistemas dinámicos formados por componentes de parámetros concentrados lineales invariantes con el tiempo se describen mediante ecuaciones diferenciales lineales invariantes con el tiempo (de coeficientes constantes). Tales sistemas se denominan sistemas lineales invariantes con el tiempo (o lineales de coeficientes constantes).

Un sistema está en equilibrio si, en ausencia de cualquier perturbación o entrada, la salida permanece en el mismo estado. Un sistema lineal, invariante con el tiempo es estable si la salida termina por regresar a su estado de equilibrio cuando el sistema está sujeto a una condición inicial.

La traza de Nyquist es una representación de la respuesta en frecuencia. Con el término *respuesta en frecuencia* nos referimos a la respuesta de un sistema ELIT, en estado estable, a una entrada senoidal. Este concepto es la base del presente proyecto, ya que un trazador de Nyquist mide la respuesta en frecuencia de sistemas ELIT, que en ocasiones son difíciles de modelar de manera experimental. Por lo tanto, resulta muy importante poner en claro el concepto de respuesta en frecuencia.

1.2. La respuesta senoidal

Se desea encontrar la respuesta $y(t)$ de un sistema ELIT a una entrada senoidal $x(t)$ descrita por la siguiente ecuación:

$$x(t) = A \operatorname{sen} \omega t \tag{1.1}$$

Consideremos que $G(s)$ es la función de transferencia del sistema ELIT que se muestra en la figura (1.1); la cual, se puede representar mediante el cociente de dos polinomios en s:

$$G(s) = \frac{P(s)}{Q(s)} = \frac{P(s)}{(s+s_1)(s+s_2)\ldots(s+s_n)} \tag{1.2}$$

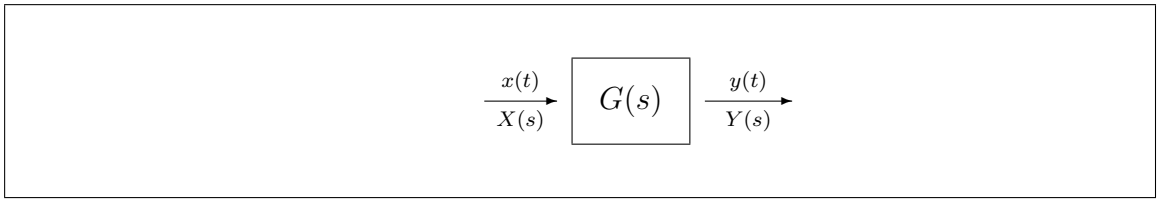


Figura 1.1: Sistema estable, lineal e invariante con el tiempo

Observemos también, que la salida del sistema esta dada por:

$$Y(s) = G(s)X(s) \tag{1.3}$$

donde $X(s)$ es la transformada de Laplace de $x(t)$

$$\begin{aligned} X(s) &= \mathscr{L}\{x(t)\} \\ &= \mathscr{L}\{A\, sen\, \omega t\} \\ &= \frac{A\omega}{s^2+\omega^2} \end{aligned} \tag{1.4}$$

Sustituyendo las ecuaciones (1.2) y (1.4) en la ecuación (1.3) obtenemos:

$$Y(s) = \left(\frac{P(s)}{(s+s_1)(s+s_2)\ldots(s+s_n)}\right)\left(\frac{A\omega}{s^2+\omega^2}\right) \tag{1.5}$$

y al efectuar la expansión en fracciones parciales de la ecuación anterior nos queda:

$$Y(s) = \frac{a}{s+j\omega} + \frac{\overline{a}}{s-j\omega} + \frac{b_1}{s+s_1} + \frac{b_2}{s+s_2} + \ldots + \frac{b_n}{s+s_n} \tag{1.6}$$

donde a es constante y $\overline{a}$ es el complejo conjugado de a, y los coeficientes b_i son constantes. Al aplicar la transformada inversa de Laplace en ambos miembros de la ecuación (1.6), llegamos al siguiente resultado:

$$\begin{aligned} y(t) &= \mathscr{L}^{-1}\{Y(s)\} \\ y(t) &= ae^{-j\omega t} + \overline{a}e^{-j\omega t} + be^{-s_1 t} + be^{-s_2 t} +, \ldots, +be^{-s_n t} \end{aligned} \tag{1.7}$$

Para un sistema estable, los polos $s_1, s_2, \ldots, s_n$ tienen partes reales negativas, es decir, tienen ubicados los polos en el semiplano izquierdo del plano complejo. Por lo tanto, cuando t tiende a infinito los términos $e^{-s_1 t}, e^{-s_1 t}, \ldots, e^{-s_1 t}$ tienden a cero.

Debemos notar que los términos $ae^{-j\omega t}$ y $\overline{a}e^{-j\omega t}$ de la ecuación (1.7) son significativos en estado estable (cuando $t \to \infty$). De lo anterior se concluye que, la respuesta en estado estable de un sistema ELIT a una entrada senoidal esta dada por:

$$y_{ss}(t) = ae^{-j\omega t} + \overline{a}e^{j\omega t} \tag{1.8}$$

Por otra parte, obteniendo el valor de a a partir de la ecuación (1.6):

$$a = \left(\underbrace{\frac{b_1}{s+s_1} + \frac{b_2}{s+s_2} + , \ldots , + \frac{b_n}{s+s_n}}_{G(s)} + \underbrace{\frac{a}{s+j\omega} + \frac{\overline{a}}{s-j\omega}}_{\frac{A\omega}{s^2+\omega^2}} \right) (s+j\omega) \Bigg|_{s=-j\omega}$$

$$\begin{aligned} a &= G(s)\frac{A\omega}{s^2+\omega^2}(s+j\omega)\Big|_{s=-j\omega} \\ &= G(s)\frac{A\omega}{(s+j\omega)(s-j\omega)}(s+j\omega)\Big|_{s=-j\omega} \\ &= \frac{A\,G(-j\omega)}{-2j} \end{aligned} \tag{1.9}$$

y realizando el mismo procedimiento para $\overline{a}$ obtenemos:

$$\overline{a} = \frac{A\,G(j\omega)}{2j} \tag{1.10}$$

Ahora, como $G(j\omega)$ es una función compleja se puede escribir como:

$$G(j\omega) = |G(j\omega)|\, e^{j\phi} \tag{1.11}$$

Donde $|G(j\omega)|$ es la magnitud y ϕ es el ángulo de $G(j\omega)$ esto es:

$$\begin{aligned} \phi &= \angle G(j\omega) \\ &= tan^{-1}\left[\frac{parte\ imaginaria\ de\ G(j\omega)}{parte\ real\ de\ G(j\omega)}\right] \end{aligned}$$

Sustituyendo la ecuación (1.11) en las expresiones (1.9) y (1.10) tenemos:

$$a = -\frac{A\,|G(j\omega)|\, e^{-j\phi}}{2j} \tag{1.12}$$

$$\overline{a} = \frac{A\,|G(j\omega)|\, e^{j\phi}}{2j} \tag{1.13}$$

Finalmente, tomando en cuenta las ecuaciones (1.12) y (1.13); la ecuación (1.8) se puede escribir de la siguiente manera:

$$\begin{aligned} y_{ss}(t) &= -\frac{A\,|G(j\omega)|\,e^{-j\phi}e^{-j\omega t}}{2j} + \frac{A\,|G(j\omega)|\,e^{j\phi}e^{j\omega t}}{2j} \\ &= A\,|G(j\omega)|\,\frac{e^{-j(\omega t+\phi)} + e^{j(\omega t+\phi)}}{2j} \\ &= A\,|G(j\omega)|\,\mathrm{sen}(\omega t+\phi) \end{aligned} \tag{1.14}$$

De acuerdo a la ecuación (1.14) observamos que un sistema ELIT sujeto a una entrada senoidal tendrá en estado estable, una salida senoidal de la misma frecuencia que la entrada; pero, la amplitud y la fase serán, por lo regular, diferentes a la entrada. Al observar la ecuación (1.14) podemos darnos cuenta, que la amplitud de la salida se obtiene mediante el producto de la entrada por la magnitud $|G(j\omega)|$, y el ángulo de fase difiere del de la entrada en la cantidad $\phi = \angle G(j\omega)$.

Lo anterior nos lleva a un importante resultado, que es la base teórica de la presente tesis, y que se explica mediante las siguientes ecuaciones:

$$|G(j\omega)| = \left|\frac{Y(j\omega)}{X(j\omega)}\right| \tag{1.15}$$

$$\angle G(j\omega) = \angle\frac{Y(j\omega)}{X(j\omega)} \tag{1.16}$$

La ecuación (1.15) nos dice que el cociente de las amplitudes de la señal senoidal de salida entre la señal senoidal de entrada es igual a la magnitud de la función de transferencia evaluada para $s = j\omega$, mientras que la ecuación (1.16) nos dice que el ángulo de la función de transferencia evaluada para $s = j\omega$ es igual al ángulo de desfasamiento entre la señal de salida y la entrada. Por lo tanto, las características de la respuesta de un sistema para una entrada senoidal se obtienen directamente de $G(j\omega)$

Es importante tener en cuenta que si tenemos una función de transferencia $G(s)$ de un sistema ELIT y queremos conocer su respuesta en frecuencia solo debemos sustituir s por $j\omega$. Teniendo a $G(j\omega)$ podemos obtener $|G(j\omega)|$ y $\angle G(j\omega)$.

1.3. La traza polar de Nyquist

La traza polar de Nyquist de una función de transferencia $G(j\omega)$, es una gráfica en el plano complejo de los valores de $G(j\omega)$ conforme ω varía de cero a infinito y, por lo tanto, es el lugar geométrico de los vectores $|G(j\omega)|\,\angle G(j\omega)$. Las proyecciones de dichos vectores sobre los ejes real e imaginario son las respectivas componentes de $G(j\omega)$. Conforme a esto, podemos comenzar a discutir el método que nos ayudará a construir un dispositivo que nos permitirá obtener la traza de Nyquist de un sistema cuya función de transferencia desconocemos.

1.4. Obtención de las componentes de $G(j\omega)$

De acuerdo a la ecuación (1.14), si excitamos a un sistema ELIT con una señal de amplitud unitaria y frecuencia ω_0, la respuesta en estado estable del sistema sería:

$$y_{ss}(t) = A\,|G(j\omega)|\,\mathrm{sen}(\omega t + \phi) \tag{1.17}$$

y conforme a la ecuación (1.11) el vector asociado a esta respuesta estaría dado por:

$$\begin{aligned} G(j\omega_0) &= |G(j\omega_0)|\,\angle G(j\omega_0) \\ &= M(\omega_0)\angle\phi(\omega_0) \end{aligned} \tag{1.18}$$

Observamos que en la ecuación (1.18) obtenemos un valor para la magnitud y otro para el ángulo de fase dependiendo del valor de frecuencia que estemos evaluando. Estos valores representan un punto de la traza polar de Nyquist; que para poder representarse en el plano complejo, es necesario contar con su componentes real e imaginaria.

Si tenemos una cantidad compleja representada en forma polar $z = re^{j\theta}$, y queremos representarla en la forma rectangular $z = a + jb$, donde a es la parte real y b es la parte imaginaria; debemos aplicar la siguiente regla de conversión:

$$\begin{aligned} a &= rcos(\theta) \\ b &= rsen(\theta) \end{aligned}$$

Por lo tanto, para el caso de $G(j\omega)$, sus componentes real e imaginaria están dadas por:

$$\begin{aligned} Re\{G(j\omega_0)\} &= |G(j\omega_0)|\,cos(\phi(\omega_0)) \\ Im\{G(j\omega_0)\} &= |G(j\omega_0)|\,sen(\phi(\omega_0)) \end{aligned}$$

Observamos que, para obtener las componentes anteriores, es necesario multiplicar la ecuación (1.17) por $cos(\phi(\omega_0))$ y por $sen(\phi(\omega_0))$ respectivamente. Para lo cual, se requerirá contar físicamente con las señales $cos(\omega t)$ y $sen(\omega t)$ para toda frecuencia en un intervalo de interés. Esto requiere de un oscilador que proporcione dos señales senoidales que estén en cuadratura para cualquier frecuencia dentro del intervalo. En la práctica, es complicado el realizar el oscilador en cuadratura requerido.

Para el dispositivo, objeto de esta tesis, las señales $cos(\omega t)$ y $sen(\omega t)$ se sustituyeron por señales cuadradas con un desfasaje de 90^o; para después, mediante el empleo de filtros paso bajas adecuadamente diseñados, obtener niveles de corriente directa (CD) proporcionales a $Re\{G(j\omega_0)\}$ y $Im\{G(j\omega_0)\}$. La figura (1.2) ilustra lo anterior.

De la figura (1.2) se observa lo siguiente:

$$\begin{aligned} S_a(t) &= |G(j\omega_0)|\,sen(\omega_0 t + \phi(\omega_0))cos(\omega_0 t) \\ S_a(t) &= \frac{1}{2}\,|G(j\omega_0)|\,[sen(\phi(\omega_0)) + sen(2\omega_0 t + \phi(\omega_0))] \end{aligned} \tag{1.19}$$

$$\begin{aligned} S_b(t) &= |G(j\omega_0)|\,sen(\omega_0 t + \phi(\omega_0))sen(\omega_0 t) \\ S_b(t) &= \frac{1}{2}\,|G(j\omega_0)|\,[cos(\phi(\omega_0)) - cos(2\omega_0 t + \phi(\omega_0))] \end{aligned} \tag{1.20}$$

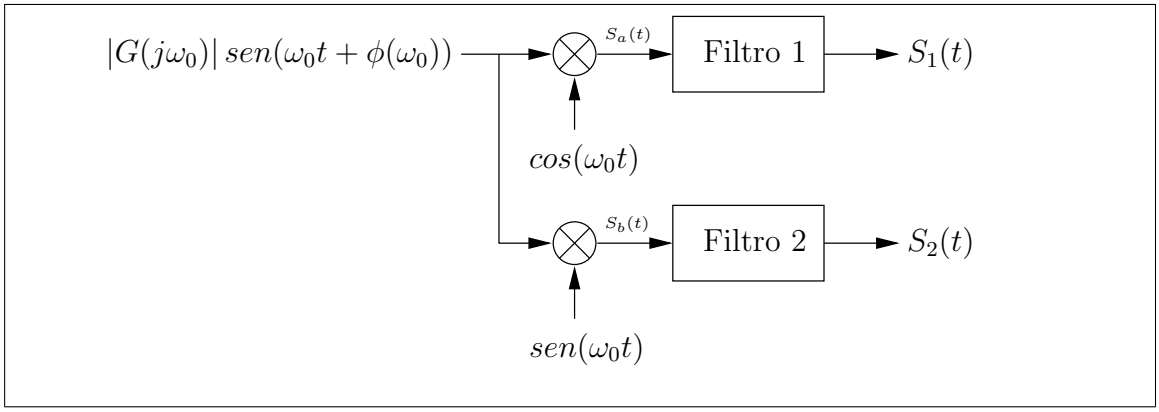


Figura 1.2: Multiplicación por las señales $cos(\omega t)$ y $sen(\omega t)$

Al analizar las ecuaciones (1.19) y (1.20) observamos que éstas contienen los valores buscados; sin embargo, se encuentran acompañados de una componente de alta frecuencia. Para obtener la componente real e imaginaria del vector, es necesario eliminar la componente de alta frecuencia con un filtro paso-bajas. De esta forma, es posible obtener un valor proporcional de las proyecciones de $G(j\omega_0)$ sobre los ejes real e imaginario; como se muestra a continuación.

A partir de la figura (1.2), la salida del filtro 1 será:

$$S_1(t) = \frac{1}{2}\,|G(j\omega_0)|\, sen(\phi(\omega_0)) \approx Im\{G(j\omega_0)\} \tag{1.21}$$

y análogamente la salida del filtro 2 será:

$$S_2(t) = \frac{1}{2}\,|G(j\omega_0)|\, cos(\phi(\omega_0)) \approx Re\{G(j\omega_0)\} \tag{1.22}$$

1.5. La multiplicación por señales cuadradas

Para justificar el uso de señales cuadradas defasadas 90^o en lugar de las señales $cos(\omega t)$ y $sen(\omega t)$ emplearemos el teorema del valor promedio y demostraremos que el resultado de la multiplicación por estas señales y su filtrado posterior, es proporcional a las componentes real e imaginaria de $G(j\omega)$.

El valor promedio de una señal periódica $v(t)$ esta dado por:

$$V_p = \frac{1}{T}\int_0^T v(t)dt \tag{1.23}$$

en donde T es el periodo de la señal.

De acuerdo a la ecuación (1.23), el valor que entregan los filtros paso bajas α_1 y α_2, de la figura (1.3), se pueden calcular de la siguiente forma:

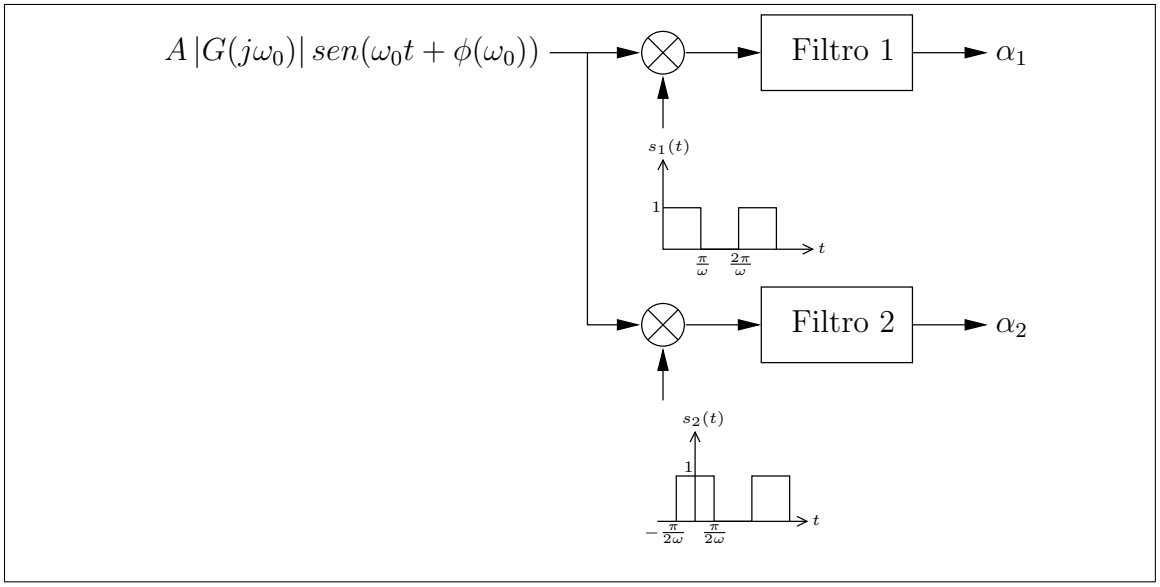


Figura 1.3: Multiplicación por señales cuadradas

Para el filtro 1:

$$\alpha_1 = \frac{1}{T}\int_0^T A\,|G(j\omega)|\,sen(\omega t + \phi)s_1(t)dt$$

donde A representa la amplitud de la señal que entra al filtro y considerando que:

- La señal cuadrada $s_1(t)$ tiene un periodo que va de 0 a $\frac{2\pi}{\omega}$
- La señal cuadrada $s_1(t)$ vale 1 de 0 a $\frac{\pi}{\omega}$
- La señal cuadrada $s_1(t)$ vale 0 de $\frac{\pi}{\omega}$ a $\frac{2\pi}{\omega}$

Entonces:

$$\begin{aligned}
\alpha_1 &= \frac{1}{T}\int_0^{\frac{\pi}{\omega}} A\,|G(j\omega)|\,sen(\omega t + \phi)dt \\
&= \frac{-1}{\omega T} A\,|G(j\omega)|\,[cos(\omega t + \phi)]_0^{\frac{\pi}{\omega}} \\
&= \frac{-A\,|G(j\omega)|}{2\pi}(cos(\pi + \phi) - cos(\phi))
\end{aligned}$$

y tomando en cuenta que: $cos(\pi+\phi) = cos\pi cos\phi - sen\pi sen\phi = -cos\phi$, α_1 queda finalmente como:

$$\alpha_1 = \frac{A\,|G(j\omega)|}{\pi}\cos(\phi) = X(j\omega) \tag{1.24}$$

Análogamente, para α_2:

$$\alpha_2 = \frac{1}{T}\int_{\frac{-\pi}{2\omega}}^{\frac{\pi}{2\omega}} A\,|G(j\omega)|\,sen(\omega t+\phi)s_2(t)dt$$

y considerando que:

- La señal cuadrada $s_2(t)$ tiene un periodo que va de $\frac{-\pi}{2\omega}$ a $\frac{3\pi}{2\omega}$
- La señal cuadrada $s_2(t)$ vale 1 de $\frac{-\pi}{2\omega}$ a $\frac{\pi}{2\omega}$
- La señal cuadrada $s_2(t)$ vale 0 de $\frac{\pi}{2\omega}$ a $\frac{3\pi}{2\omega}$

entonces:

$$\begin{aligned}\alpha_2 &= \frac{1}{T}\int_{\frac{-\pi}{2\omega}}^{\frac{\pi}{2\omega}} A\,|G(j\omega)|\,sen(\omega t+\phi)dt \\ &= \frac{-1}{\omega T}A\,|G(j\omega)|\,[cos(\omega t+\phi)]_{\frac{-\pi}{2\omega}}^{\frac{\pi}{2\omega}} \\ &= \frac{-A\,|G(j\omega)|}{2\pi}(cos(\frac{\pi}{2}+\phi)-cos(\phi-\frac{\pi}{2}))\end{aligned}$$

y utilizando las identidades: $cos(\frac{\pi}{2}+\phi) = -sen\pi$ y $cos(\phi-\frac{\pi}{2}) = sen\phi$, α_2 queda finalmente como:

$$\alpha_2 = \frac{A\,|G(j\omega)|}{\pi}\,\mathrm{sen}(\phi) = Y(j\omega) \tag{1.25}$$

Las ecuaciones (1.24) y (1.25) son proporcionales a las componentes real e imaginaria de $G(j\omega)$ buscadas. Por lo tanto, es válido sustituir las señales $cos(\omega t)$ y $sen(\omega t)$ por señales cuadradas desfasadas 90^o.

1.6. Representación en el plano complejo

Se ha demostrado que al multiplicar a la respuesta del sistema por una señal cuadrada se puede obtener su componente de valor promedio, seguido de una serie de componentes de alta frecuencia. Mediante el empleo de filtros paso-bajas es posible eliminar las componentes de alta frecuencia de las ecuaciones (1.19) y (1.20) para obtener, únicamente, los términos $X(\omega_0)$ y $Y(\omega_0)$ descritos en las ecuaciones (1.24) y (1.25). Estos términos son proporcionales a las componentes real e imaginaria de la señal de salida del sistema bajo análisis; por lo tanto, obtener sus componentes se reduce a una simple relación.

$$Re\{G(j\omega)\} = \frac{\pi}{A}X(j\omega) = |G(j\omega)|\,cos(\phi(\omega)) \tag{1.26}$$

$$Im\{G(j\omega)\} = \frac{\pi}{A}Y(j\omega) = |G(j\omega)|\,sen(\phi(\omega)) \tag{1.27}$$

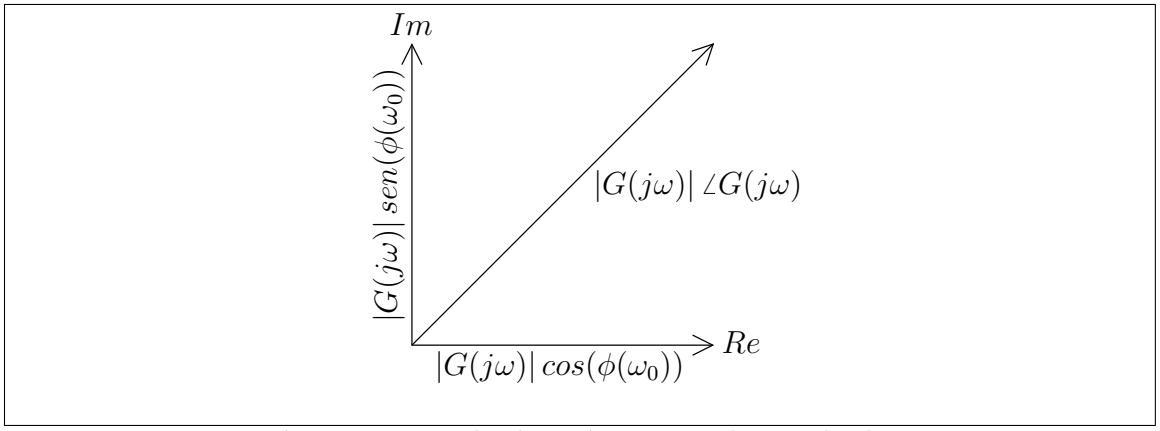


Figura 1.4: Multiplicación por señales cuadradas

Una vez que se obtienen los términos $Re\{G(j\omega)\}$ e $Im\{G(j\omega)\}$ es posible representar en el plano complejo al vector correspondiente a la respuesta en frecuencia de $G(j\omega)$ para una cierta frecuencia ω_0. La representación gráfica se muestra en la figura (1.4)

Si se dibujan todos los vectores del sistema de prueba dentro del rango $0 < \omega < \infty$, se obtiene la traza de Nyquist del sistema analizado. Un ejemplo de traza de Nyquist generada con esta idea se muestra en la figura (1.5). En donde todos los puntos de la traza polar de $G(j\omega)$ representan el punto terminal de un vector en un valor determinado de ω.

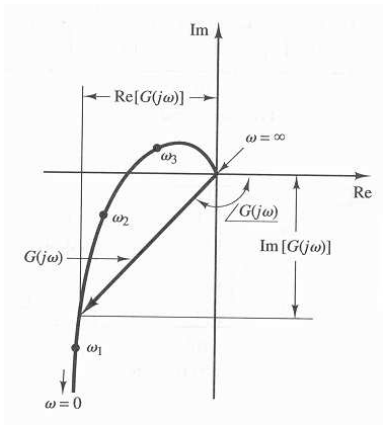


Figura 1.5: Ejemplo de Traza Polar de Nyquist

Por ello es importante, que el dispositivo a construir sea capaz de realizar un barrido en un rango de frecuencias determinado, por ejemplo de ω_i a ω_f, y que para cada frecuencia dentro de ese rango, se pueda obtener un vector característico de la respuesta en frecuencia.

En este capítulo se han discutido las bases teóricas sobre las cuales, se ha construido el trazador digital de Nyquist. En el siguiente capítulo se explicará, de manera detallada, la forma en que fue construido el hardware que se encargará de realizar el análisis de respuesta en frecuencia de un sistema puesto a prueba.

Capítulo 2

Hardware analógico

2.1. Introducción

En el capítulo anterior se habló de la base teórica, a partir de la cual, es posible construir un dispositivo que nos permita obtener la traza de Nyquist de un sistema lineal e invariante en el tiempo, de manera experimental. En este capítulo se hablará acerca del diseño del trazador digital de Nyquist. Se explicarán cada una de las etapas de las que consta el dispositivo y se describirá de manera detallada su funcionamiento.

2.2. Etapas del Trazador de Nyquist

La figura (2.1) muestra el esquema general del Trazador Digital de Nyquist. El usuario, mediante la computadora, solicita un análisis de un sistema cualquiera. El programa de interfaz de usuario, que se ejecuta en la computadora, manda mediante al puerto serie los programas de control que son recibidos por el TDN a través del HC11, quien carga en la memoria los programas y los ejecuta. Mediante estos programas y a partir del Hardware del TDN, se lleva a cabo el análisis del sistema. Los resultados del análisis son leídos por el HC11 y enviados, vía puerto serie, a la computadora del usuario en donde son procesados y presentados en forma gráfica. El presente capítulo trata sobre el bloque que llamamos

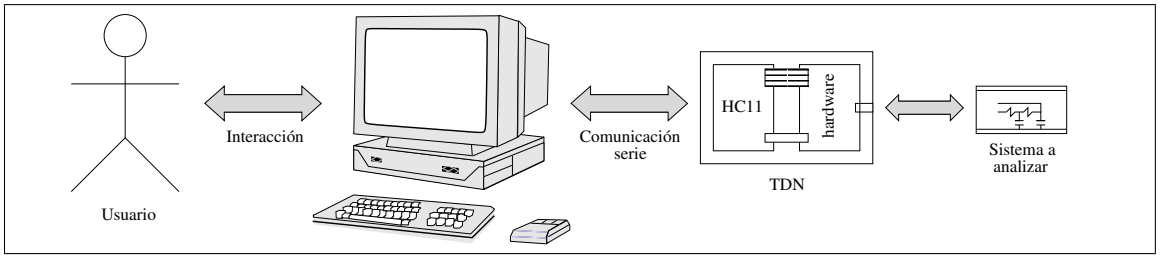


Figura 2.1: Diagrama General del Análisis de respuesta en frecuencia mediante el TDN

TDN. En el capítulo anterior se habló de la base teórica sobre la cual se fundamenta y se determina que es posible su construcción. A continuación se describirá el funcionamiento de cada una de sus partes.

La figura (2.2) nos muestra un diagrama conceptual del dispositivo a construir. El generador de funciones nos proporciona dos señales: una senoidal y la otra cuadrada, las cuales están desfasadas 90^o y con amplitudes iguales. La señal senoidal es la entrada del sistema por analizar. A su vez, esta señal senoidal se *cuadratiza*. De esta forma se tienen dos señales cuadradas desfasadas 90^o. La salida del sistema analizado (que es una señal senoidal) se multiplica por cada una de las señales cuadradas, mediante los switches analógicos y posteriormente el resultado de la multiplicación se filtra para obtener únicamente la componente de directa, la cual se adecúa a un voltaje que el convertidor analógico-digital del HC11 puede leer (esto es, un voltaje entre 0 y 5 [V]). También se obtiene la amplitud de la señal senoidal de entrada mediante un circuito detector de pico máximo y se lee por medio del convertidor analógico digital del HC11. El proceso anteriormente descrito se tiene que efectuar en un rango de frecuencias por lo que la frecuencia de operación del generador de señales se modifica a partir del bloque de control de frecuencia. Por último, el frecuencímetro se encarga de tomar la lectura de la frecuencia real a la que esta trabajando el generador de funciones, esta frecuencia es leída por el microntrolador HC11.

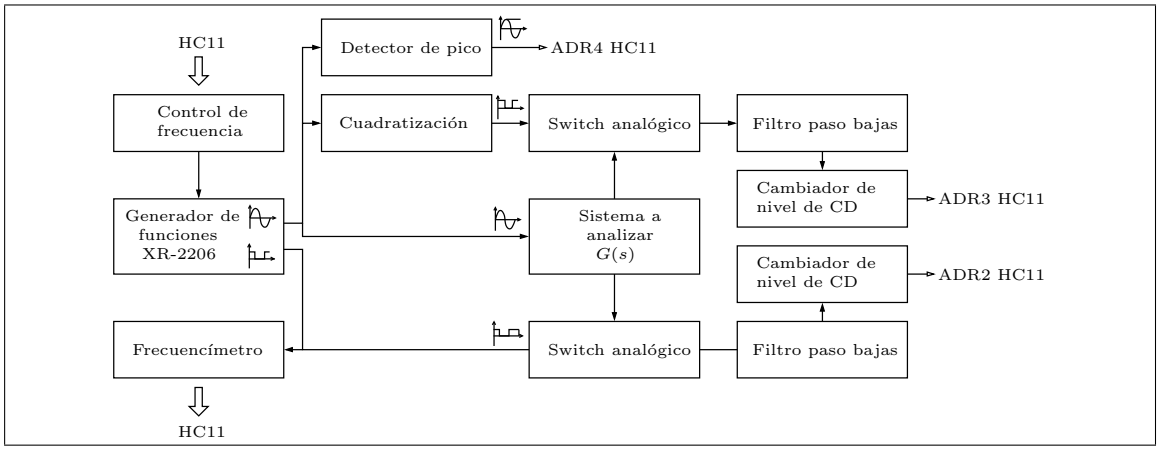


Figura 2.2: Diagrama de bloques del TDN

2.3. El Generador de funciones

Para el desarrollo del trazador digital de Nyquist se decidió emplear el circuito integrado XR2206. El circuito XR2206 es un generador de funciones monolítico capaz de producir señales senoidales, cuadradas, triangulares, rampas y algunas otras formas de onda de gran estabilidad. Las formas de onda de salida pueden ser modificadas de dos

diferentes maneras, tanto en frecuencia, como en amplitud mediante un voltaje externo. La frecuencia de operación puede ser seleccionada externamente sobre un rango que va de 0.01 Hz hasta más de 1 MHz.

La decisión de emplear el generador de funciones XR2206 de Exar se debió básicamente a dos razones:

- El generador de funciones puede producir dos señales de la misma frecuencia pero de forma diferente (senoidal y cuadrada ó triangular y cuadrada), desfasadas 90^o una con respecto a otra

- Es posible modificar externamente la frecuencia de las señales de salida, en un rango de frecuencias de operación muy amplio.

El circuito básico de operación del XR2206 se tomó directamente de su manual de operación, disponible en internet [3] y se muestra en la figura (2.3). A partir de este circuito es posible modificar tanto la amplitud de las señales de salida como la frecuencia a la que operan. El bloque que aparece en la figura (2.3), etiquetado *Circuitos de control de barrido de frecuencia*, es el responsable de modificar la frecuencia de operación del generador en un rango que va de 10 Hz a 100 KHz. Los análisis de sistemas que se realicen mediante este dispositivo, solo podrán realizarse en el rango anteriormente descrito. Esta es una de las limitaciones de nuestro dispositivo. Una descripción más detallada de los circuitos que nos sirven para llevar a cabo el barrido en frecuencia, se ofrece en la sección(2.8).

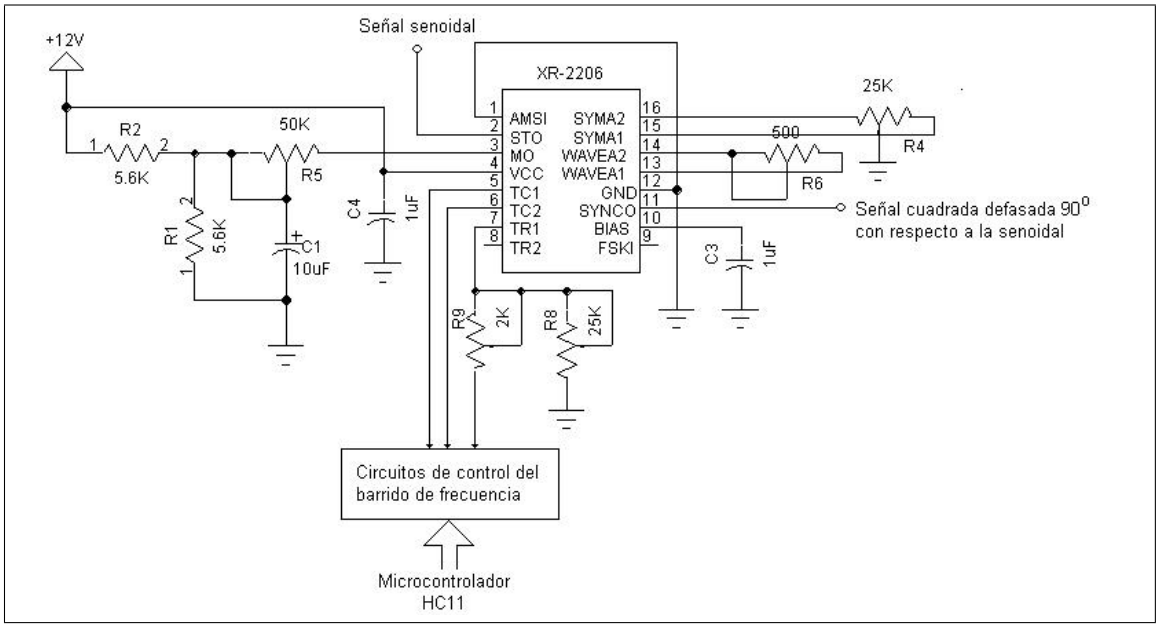


Figura 2.3: Circuito de prueba del generador de señales XR2206

La señal cuadrada que nos entrega el circuito XR2206, esta desfasada 90^o con respecto a la señal senoidal. A partir de la señal senoidal que nos entrega el circuito directamente, se obtuvo otra señal cuadrada mediante un circuito comparador LM339 en configuración de detector de cruce por cero.

La señal senoidal a la salida del circuito XR2206 cuenta con una componente de DC, la cual se eliminó, utilizando para ello un circuito RC que funciona como filtro paso-altas. Esta señal (con offset cero) pasa por un acoplador y posteriormente se provee al detector de cruce por cero. Por último, la amplitud de la señal se ajusta a 5[V], de esta manera la señal que se obtiene es una señal cuadrada TTL con la misma fase que la señal senoidal. La figura (2.4)muestra el circuito empleado para llevar a cabo la cuadratización de la señal.

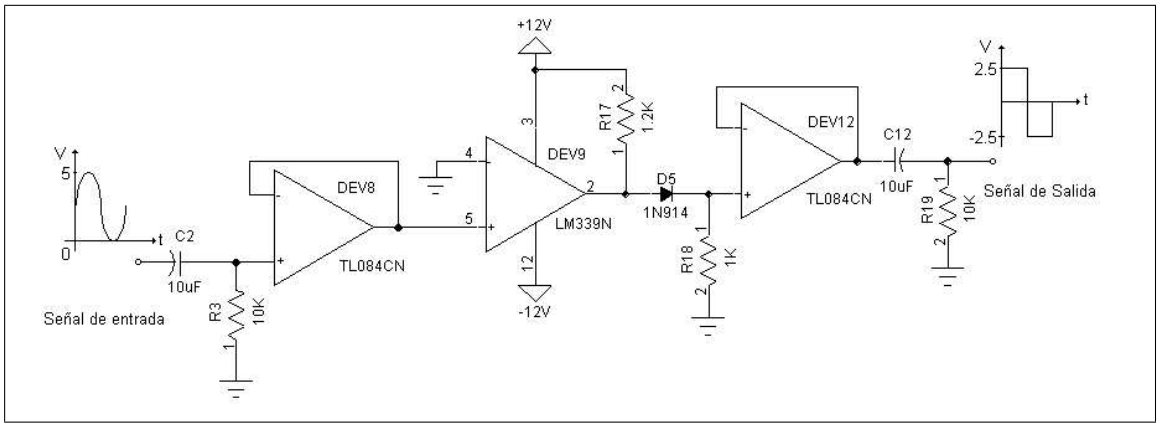


Figura 2.4: Circuito empleado para cuadratizar una señal senoidal

El switch analógico CD4016 emplea señales de control de AC puras (con offset 0), por esta razón es necesario eliminar la componente de DC de la señal cuadrada TTL (mediante un circuito RC, que funciona como filtro paso-altas), de esta forma obtenemos una señal cuadrada de 5[V] de amplitud pero con offset 0.

La segunda señal cuadrada se obtiene directamente del circuito XR2206. Esta señal esta desfasada 90^o con respecto a las señales senoidal y cuadrada por lo que únicamente se eliminó el nivel de DC de la señal y se ajustó su amplitud a 5[V] como se muestra en la figura(2.5). La señal senoidal obtenida directamente del generador de señales, debe ser una señal pura de AC. Ésta señal es la que se utiliza como entrada para nuestro circuito de prueba.

2.4. Implantación física de la multiplicación por onda cuadrada

Para multiplicar las señales cuadradas por la salida del sistema a analizar, se empleó el circuito CD4016, que es un switch analógico y que está controlado por las misma señales cuadradas generadas. El switch analógico CD4016 se alimenta con 5[V] y -5[V]. Las señales

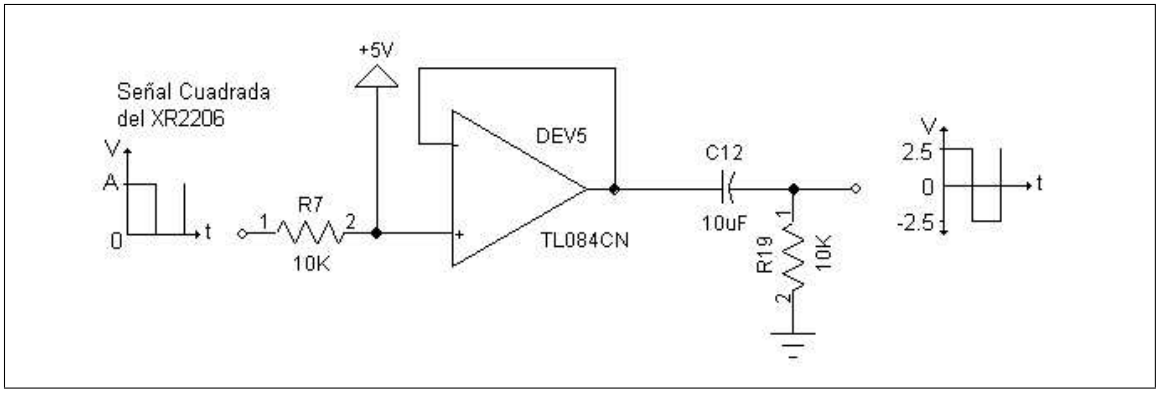


Figura 2.5: Circuito empleado para adaptar una señal cuadrada de control

de control (señales cuadradas desfasadas 90^o entre sí) deben ser señales de AC puras (sin componente de DC) y con 5[V] de amplitud.

La operación de multiplicación por una señal cuadrada es, en general, muy simple. La señal cuadrada rige al switch analógico de tal forma que cuando la señal cuadrada está en su nivel alto (2.5[V]) lo que se observa a la salida del switch es la entrada, mientras que cuando la señal cuadrada esta en su nivel bajo (-2.5[V])lo que observamos a la salida del switch es un cero. Esto equivale a multiplicar por 1 o por 0 a la entrada del switch. Cuando multiplicamos a la respuesta del sistema de prueba a una entrada senoidal $sen(\omega_o t)$ con la señal cuadrada que tiene un ángulo de desfasamiento de 90^o, obtenemos la componente real del vector asociado a la frecuencia ω_o, con sus respectivas componentes de alta frecuencia, que es necesario eliminar. De forma análoga, cuando multiplicamos a la respuesta del sistema de prueba a una entrada senoidal por una señal cuadrada con el mismo ángulo de fase que el de la señal de entrada del circuito de prueba, entonces obtenemos la componente imaginaria del vector asociado a la frecuencia ω_o, con sus componentes de alta frecuencia. El proceso anteriormente descrito se muestra en la figura(2.6).

2.5. Eliminación de componentes de alta frecuencia

Como se explicó en la sección anterior, las salidas del switch CD4016, contienen componentes de alta frecuencia que es necesario eliminar para poder obtener los términos descritos por las ecuaciones (1.24) y (1.25). Lo anterior se realizó con el empleo de filtros paso-bajas. El diseño de estos filtros se basó en las siguientes consideraciones:

- A través de los filtros sólo deben pasar señales de corriente directa.
- El trazador digital de Nyquist, debe funcionar para un rango de frecuencias de 10 Hz a 100 KHz. Por lo tanto, la frecuencia mínima de operación del trazador debe ser de 10 Hz. Es por ello que la frecuencia de corte debe ser menor que 10 Hz. Y la frecuencia de supresión se debe ajustar a 10 Hz.

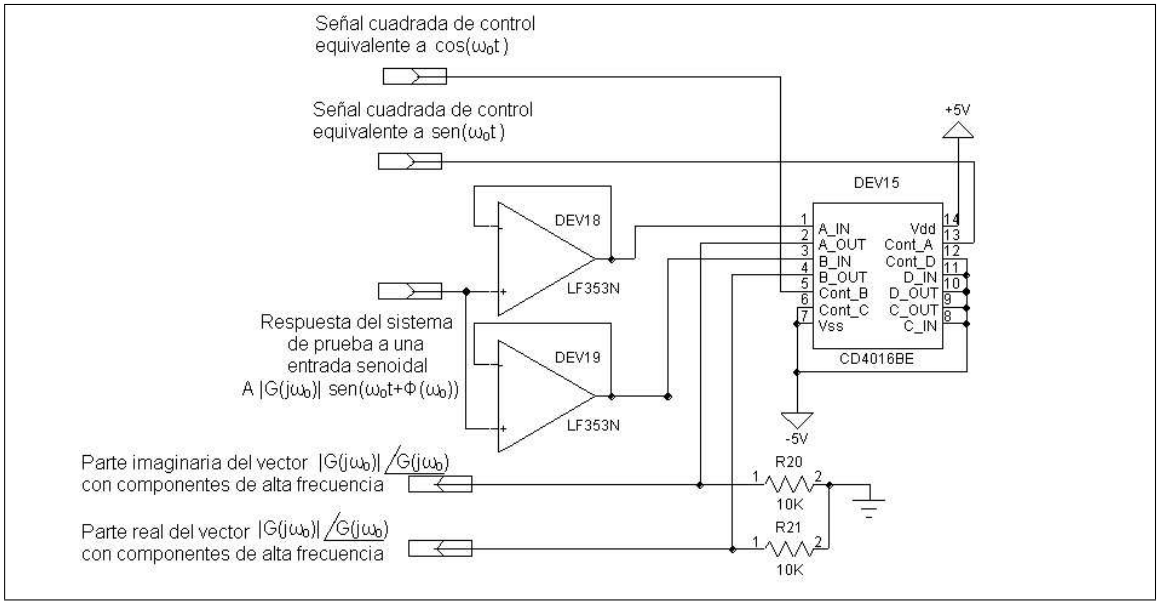


Figura 2.6: Switch analógico CD4016 empleado para llevar a cabo la multiplicación por señales cuadradas

- La ganancia total del filtro (K) debe ser igual a 1
- Debe existir una pérdida de 60[dB] entre la frecuencia de corte y la frecuencia de supresión.

Es por ello que se escogió una plantilla de diseño como la que aparece en la figura (2.7). Las características de los filtros son:

- Frecuencia de corte $f_c = 5[Hz]$
- Frecuencia de supresión $f_s = 10[Hz]$
- Perdida de 60[dB] entre f_c y f_s.

Los filtros se diseñaron utilizando una aproximación Butterworth. El diseño de los filtros se realizó con el auxilio de un programa de computadora [8]. Dicho programa acepta como entradas los parámetros de diseño del filtro y arroja como salidas los valores de resistencias y capacitores del filtro que se quiere implementar. Mediante este programa se implementó físicamente el filtro que se muestra en la figura (2.8)

2.6. Cambiador de nivel de Corriente Directa

La salida de cada filtro debe ser leída por el convertidor analógico digital del HC11. Sin embargo, debido a que el microprocesador HC11 poseé un convertidor analógico digital que

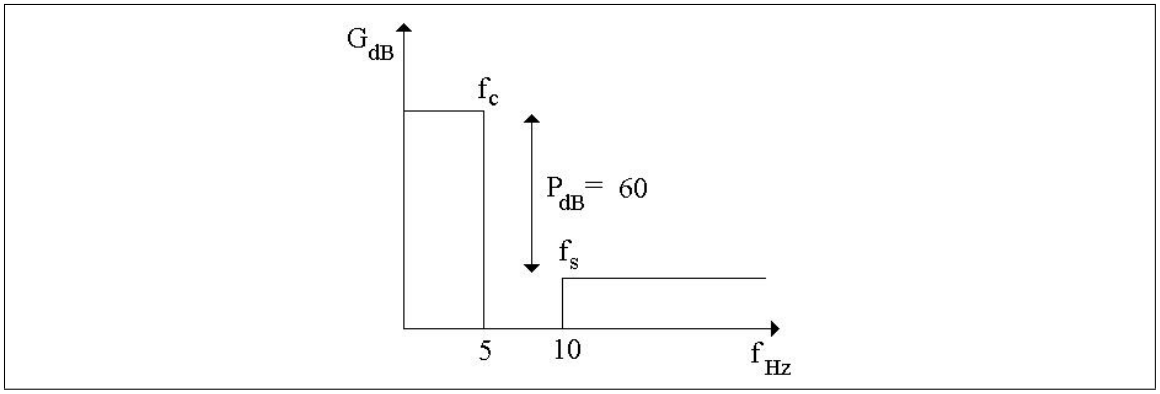


Figura 2.7: Plantilla de diseño de los filtros paso-bajas

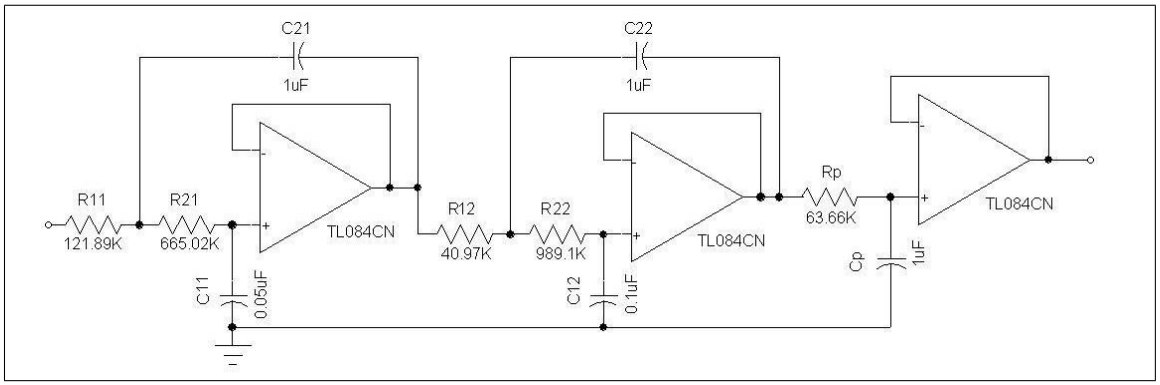


Figura 2.8: Filtro paso-bajas

está implementado con una fuente de voltaje de referencia de 5.12[V], entonces el rango para la señal analógica de entrada debe ser de 0 a 5.12[V]. Además debemos considerar que el voltaje de DC de salida de los filtros es bipolar en la mayoría de los casos. Por lo anterior, para poder registrar voltajes bipolares debe llevarse a cabo un acondicionamiento de las señales que entrarán al convertidor. Esto se llevó a cabo mediante la implementación de un circuito analógico que transforma un rango de -2 a 2[V] en un rango de 0 a 5[V]. El cambio de nivel de voltaje se puede ver más claramente en la figura (2.9). Por ejemplo, cuando el valor del voltaje de entrada V_E es igual a $-2[V]$ el voltaje de salida del cambiador de nivel V_S es $0[V]$, a su vez cuando $V_E = 2[V]$, $V_S = 5[V]$. El circuito cambiador de nivel responde a la ecuación (2.1), por lo que el cambiador de nivel se puede calibrar manualmente a partir de los potenciómetros que se muestran en la figura (2.10). Las salidas de los cambiadores de nivel se conectan directamente al convertidor analógico digital del HC11.

$$V_S = \frac{R_f}{R_1} V_E - \frac{R_f}{R_2} V_0 \tag{2.1}$$

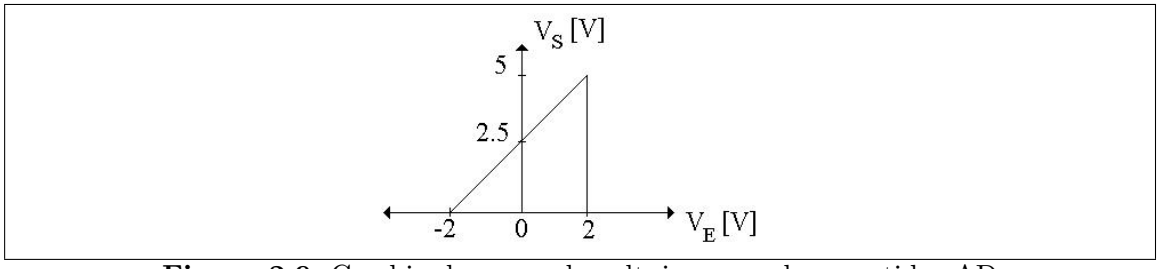


Figura 2.9: Cambio de rango de voltajes para el convertidor AD

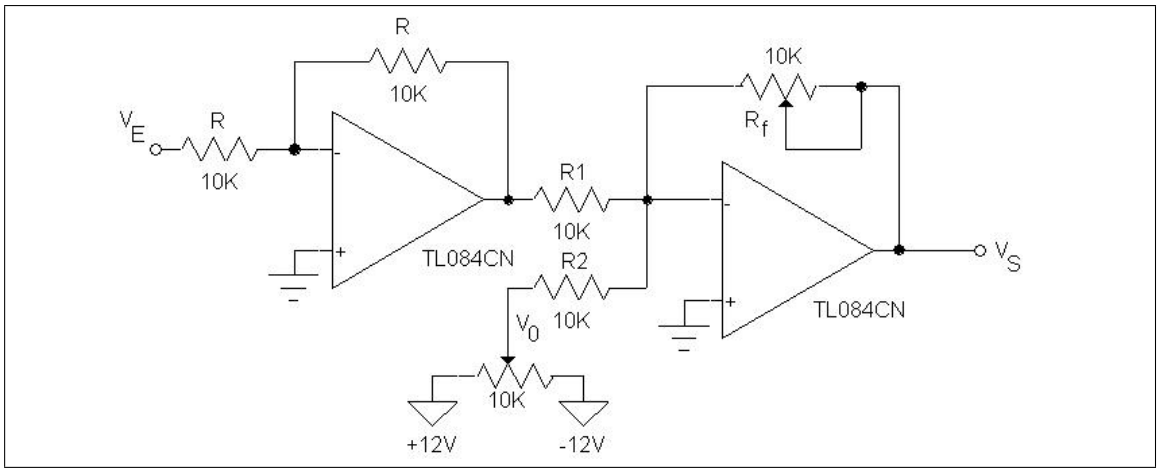


Figura 2.10: Circuito que acondiciona el voltaje para el convertidor AD

2.7. Circuito detector de pico máximo

En la sección (1.6) se mencionó que para obtener la parte real y la parte imaginaria de $G(j\omega)$ es necesario multiplicar la salida de los filtros por la cantidad $\frac{\pi}{A}$, en donde A representa el voltaje pico a pico de la señal de entrada al sistema que se está analizando. Por ello es necesario obtener la amplitud A de la señal de entrada, para cada una de las frecuencias que se analicen.

Lo anterior se realizó mediante la implementación del circuito detector de pico que se muestra en la figura (2.11). La entrada de este circuito es la señal senoidal que se aplica al sistema de prueba. La salida del circuito representa una fracción de la amplitud de la señal de entrada ($\frac{A}{2}$), y está conectada directamente al convertidor analógico digital del HC11.

Es necesario medir la amplitud de la señal de entrada para cada una de las frecuencias que se estén analizando. El capacitor C14 de la figura (2.11), retiene el voltaje más alto de la señal de entrada del detector. Debido a que se deben llevar a cabo varias mediciones consecutivas con el detector de pico, se emplea un circuito auxiliar de descarga como el que aparece en la figura (2.11). La base del transistor Q5, está conectada al pin PF7 del

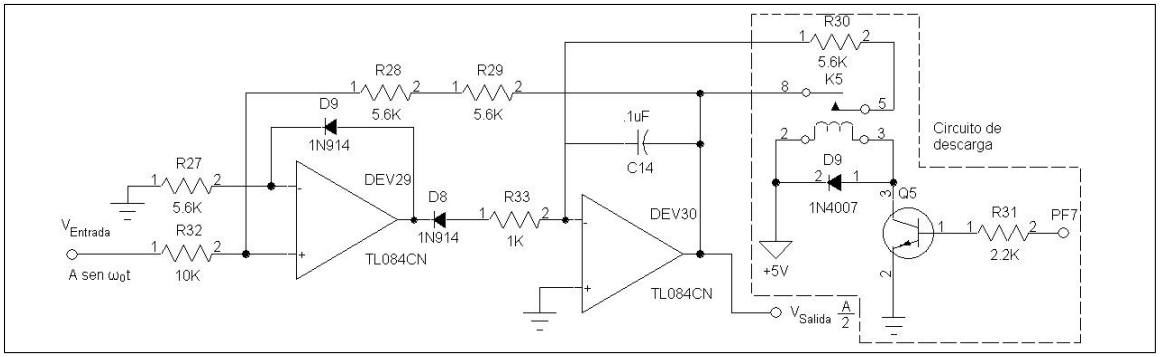


Figura 2.11: Circuito detector de pico

HC11. Cuando hay 5[V] en la base, el relay D9 se cierra y el capacitor se descarga con la resistencia R30.

2.8. Circuitos de control de frecuencia

Para llevar a cabo un análisis de respuesta en frecuencia es necesario aplicar una señal senoidal de entrada, al sistema que se quiera analizar. La frecuencia de dicha señal debe variar en un cierto en un rango $\omega_0 < \omega < \omega_f$. El rango de operación depende, en gran parte, del circuito generador de señales empleado para generar la señal senoidal. En nuestro caso, se utilizó el circuito XR2206 de Exar, el cual puede generar señales que van de 0.01 Hz a 1 MHz. Sin embargo no se utilizó completamente este rango ya que la distorsión de las señales con frecuencias de más de 100 KHz es muy considerable, como se puede observar en la figura (2.12). Además se consideró que para fines prácticos, los análisis podrían empezar en 10 Hz, ya que es poco común que se lleven a cabo análisis de respuesta en frecuencia para frecuencias tan bajas (de 0.01 Hz a 10 Hz). El circuito XR2206 permite modificar la frecuencia de las señales que entrega si se modifican los valores de los capacitores y las resistencias de tiempo que se conectan a sus pines 5, 6, 7 y 8 de acuerdo su diagrama de bloques (2.13). El Oscilador Controlado por Voltaje (VCO por sus siglas en inglés) produce una frecuencia de salida proporcional a la corriente de entrada que se ajusta mediante las resistencias conectadas a las terminales de tiempo. Es posible conectar 2 resistencias para cada uno de los pines de tiempo del XR2206 (pines 7 y 8), estas resistencias se seleccionan mediante el pin FSK (pin 9). Para el TDN únicamente se conectó una resistencia en el pin 7 y no se utilizó el pin 9 como se observa en la figura (2.3).

El VCO tiene un voltaje interno de 3.12[V]. Si se aplica un voltaje igual en el pin 7 del XR2206 el voltaje en el VCO es igual a 0, de esta forma la frecuencia de operación del XR2206 es la mínima dentro de un cierto rango de operación establecido por el capacitor de tiempo. Si el voltaje que se aplica al pin 7 es de 0[V], entonces, la frecuencia de operación es la máxima dentro de ese rango. Debido a que la relación entre el voltaje en el VCO y

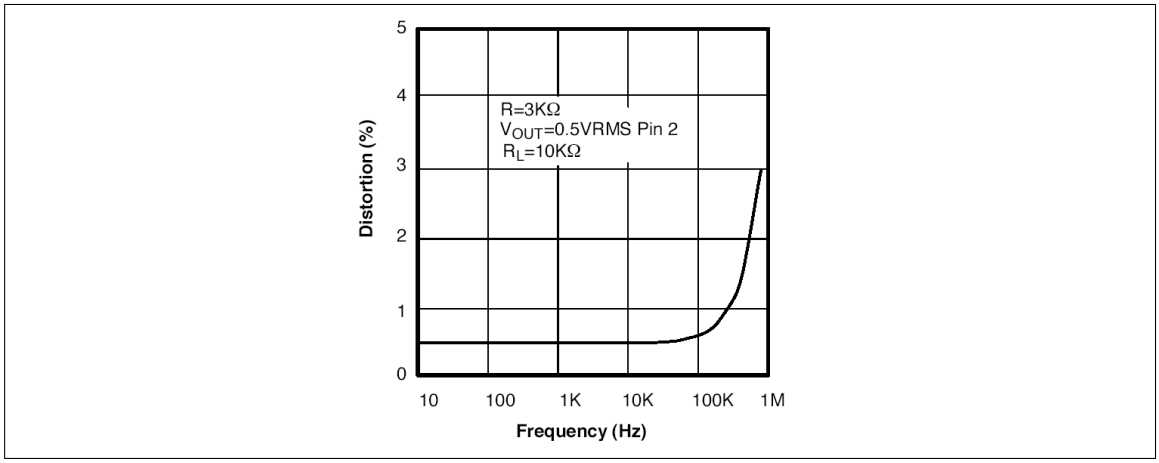


Figura 2.12: Frecuencia contra distorsión del XR2206

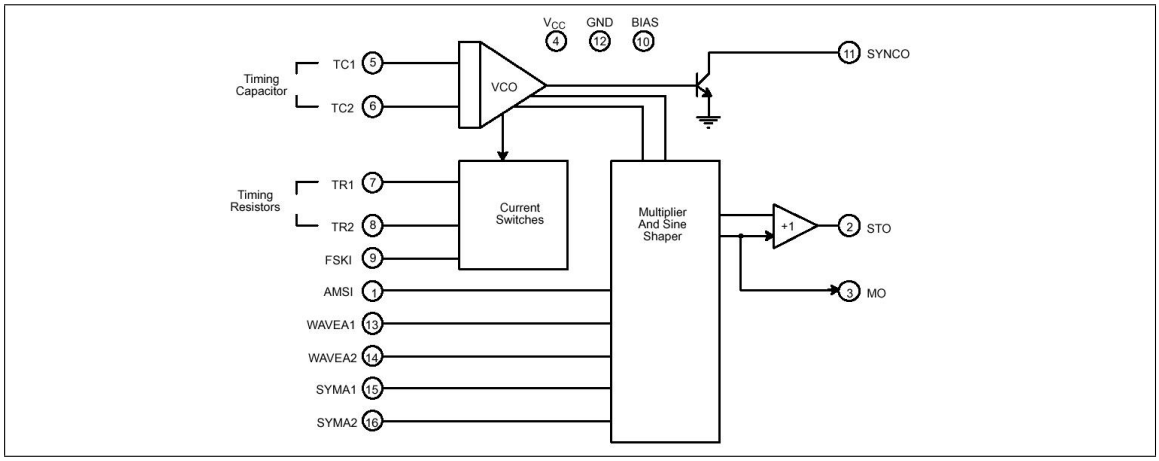


Figura 2.13: Diagrama de bloques del XR2206

la frecuencia de salida es lineal, el XR2206 se puede ajustar a una frecuencia de operación deseada, conociendo el valor del voltaje que se debe aplicar en el pin 7.

Los rangos de operación se establecen a partir del valor del capacitor de tiempo. La frecuencia de operación f_0 del XR2206 está dada por:

$$f_0 = \frac{1}{RC} \tag{2.2}$$

Donde C es el capacitor de tiempo externo conectado entre los pines 5 y 6, y R es la resistencia de tiempo conectada, en nuestro caso, en el pin 7. Como puede observarse, la frecuencia es inversamente proporcional al valor de la capacitancia, por lo tanto, es posible variar la frecuencia en décadas cambiando el valor del capacitor de tiempo en múltiplos de 10. La tabla (2.1) nos muestra los rangos de frecuencia que se utilizaron para el TDN

con sus respectivos valores de capacitancia.

Capacitancia [μF]	Rango de Frecuencia [Hz]
10	10 - 100
1	100 - 1K
0.1	1 - 10K
0.01	10 - 100K

Cuadro 2.1: Rangos de frecuencia del TDN

2.8.1. Los capacitores de rango

Es posible seleccionar el rango de frecuencia de operación del TDN a partir del capacitor de tiempo que se conecta a los pines 5 y 6 del XR226. Los capacitores y los rangos de operación empleados para el TDN son los de la tabla ((2.1). Para cambiar el rango de frecuencia de operación basta con cambiar el capacitor de tiempo. Esta función la lleva a cabo el circuito que aparece en la figura (2.14). Cada capacitor de tiempo está conectado

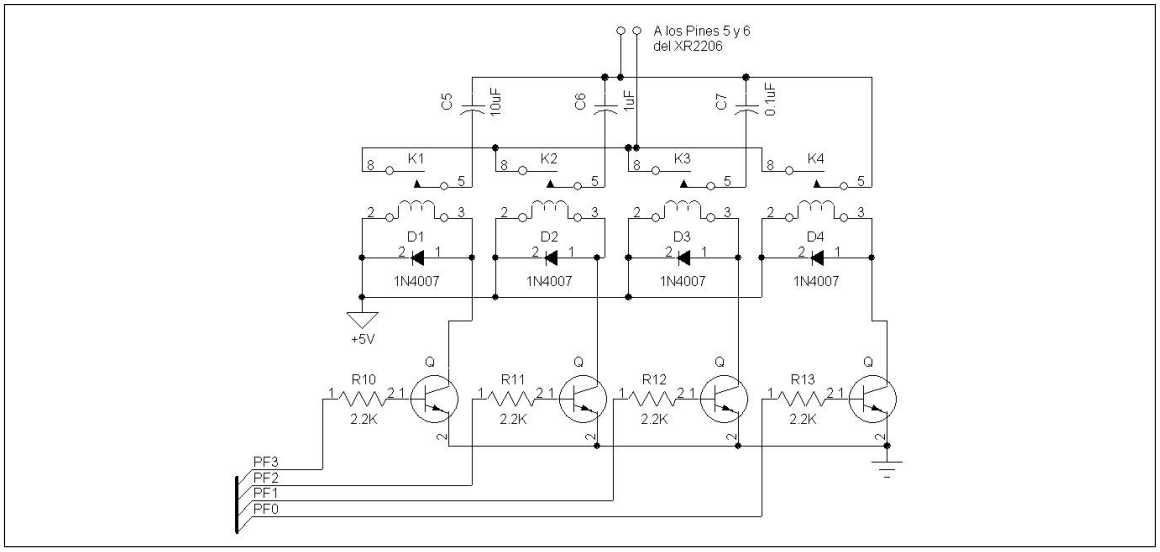


Figura 2.14: Circuito de switcheo de los capacitores de tiempo

a un relevador que a su vez está conectado al puerto F (pines PF0, PF1, PF2 y PF3) del microcontrolador HC11. La función del relevador es simple. Funciona como un *switch* que se cierra cuando a la base del transistor se le aplica un voltaje de aproximadamente 5[V], (es decir cuando se escribe un 1 en el bit del puerto F que controla a dicho reelevador). Los programas de control (ver capítulo 3) nunca cierran dos relevadores a la vez, por lo que los rangos de operación son los establecidos en la tabla (2.1).

2.8.2. El Convertidor Digital a Analógico

Una vez que se ha seleccionado un rango de frecuencia, el barrido se realiza al modificar el valor del voltaje que se aplica a la resistencia conectada al pin 7 del XR2206. Este voltaje debe variar dentro de un rango de 0 a 3.12[V] aproximadamente. Para lo anterior se implementó el circuito de la figura (2.15). El circuito está compuesto de un convertidor

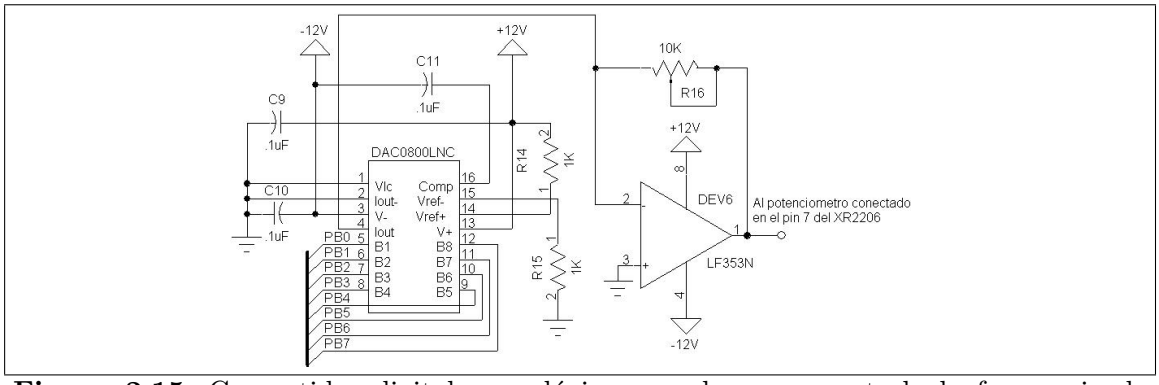


Figura 2.15: Convertidor digital a analógico con el que se controla la frecuencia de operación del XR2206

digital a analógico (DAC por sus siglas en inglés) cuyas entradas están conectadas al puerto B del microntrolador HC11. El rango del voltaje de salida del DAC va de 0 - 3.12[V]. Con los 8 bits de entrada, el DAC puede darnos $\frac{3,12}{2^8} = 12[mV]$ aproximadamente por cada paso o cambio de bit. Cabe destacar que, para cada rango de frecuencias, se podrán obtener 256 puntos, por lo que la resolución del aparato no es la misma en cada rango, siendo mayor para el rango de frecuencias más bajas (de 10 a 100 [Hz]). La relación entre el voltaje del VCO y la frecuencia de operación del XR2206 es lineal, como lo es también la relación entre el byte de entrada en el DAC y el voltaje de salida del mismo. Gracias a este hecho, se obtuvo una relación lineal entre el byte que se coloca en el DAC y la frecuencia de operación del XR2206. Cuando el byte de entrada es \$00, el voltaje de salida del DAC es de 3.12[V] y la frecuencia de operación es la máxima dentro del rango seleccionado , mientras que cuando el byte de entrada es \$FF, el voltaje de salida del DAC es de 0[V] y la frecuencia de operación es la mínima del rango. Con estos dos puntos obtenemos una relación lineal como se muestra en la figura (2.16).

Sin embargo esta relación puede ser más exacta si se toman más puntos de la recta mostrada en la figura (2.16). El proceso para obtener la relación entre el byte de entrada (b)y la frecuencia de operación (f) consiste en obtener 16 puntos de la recta. Estos puntos se obtienen colocando en el DAC los bytes \$00, \$10, \$20, \$30... \$F0 y leyendo la frecuencia correspondiente a cada byte con el frecuencímetro descrito en la sección (2.9). Con estos puntos se realiza un ajuste por mínimos cuadrados para obtener los valores de m y b_0 de

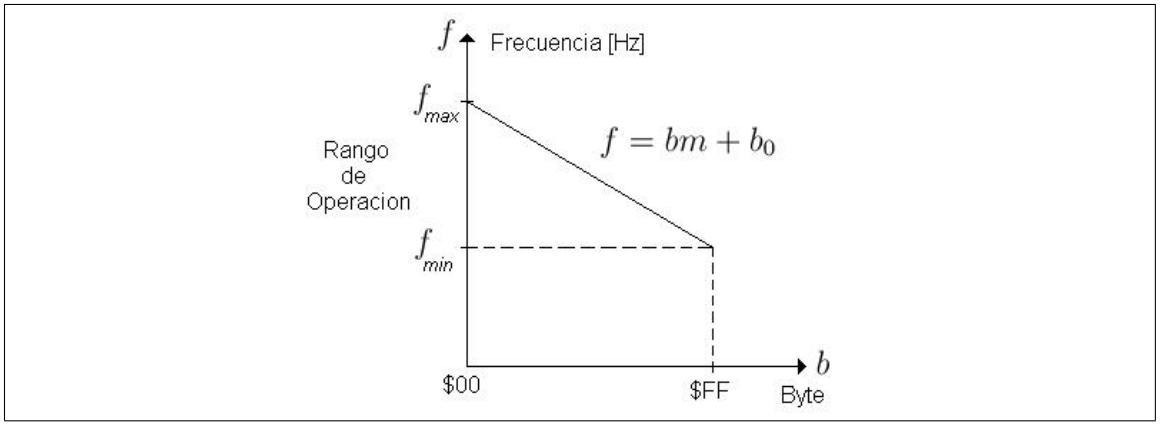


Figura 2.16: Relación entre el byte de entrada en el DAC y la frecuencia de operación

la ecuación (2.3). Estos pasos forman parte de la *calibración*. del TDN.

$$f = bm + b_0 \tag{2.3}$$

2.9. Frecuencímetro

Para cada punto del análisis de la respuesta en frecuencia que se efectúa con el TDN se obtiene una medición de su frecuencia. Para llevar a cabo esta medición, se construyó el frecuencímetro mostrado en la figura (2.17). El circuito esta compuesto de 3 contadores 74HC590 conectados en cascada, por lo que la frecuencia medida se entrega en 3 bytes. El frecuencímetro funciona de la siguiente forma: la señal cuadrada que sale del X2206 se conecta como el reloj del primer contador. El arreglo de contadores cuenta los flancos de subida de la señal durante 1 segundo exactamente. Las salidas de los 3 contadores están conectadas al puerto C del microcontrolador HC11. Controlando el multiplexor 74LS138, el HC11 lee cada una de las salidas de los 3 contadores (3 bytes) mediante el puerto C. El valor decimal representado por los 3 bytes corresponden a la frecuencia de operación del TDN. Nos auxiliamos del buffer 74HC14 para acondicionar la señal cuadrada a la entrada del arreglo de contadores.

2.10. La tarjeta FACIL11B

Se ha mencionado en muchas ocasiones que el control de las diversas funciones del TDN se llevan a cabo con el microcontrolador HC11. Este componente esta montado sobre una tarjeta de desarrollo llamada FACIL11B desarrollada por el M.I Antonio Salvá Calleja en la Facultad de Ingeniería. Dicha tarjeta explota al máximo las capacidades del microcontrolador modelo MCU68HC11F1 empleado en la presente tesis. Entre las características más importantes de este modelo están:

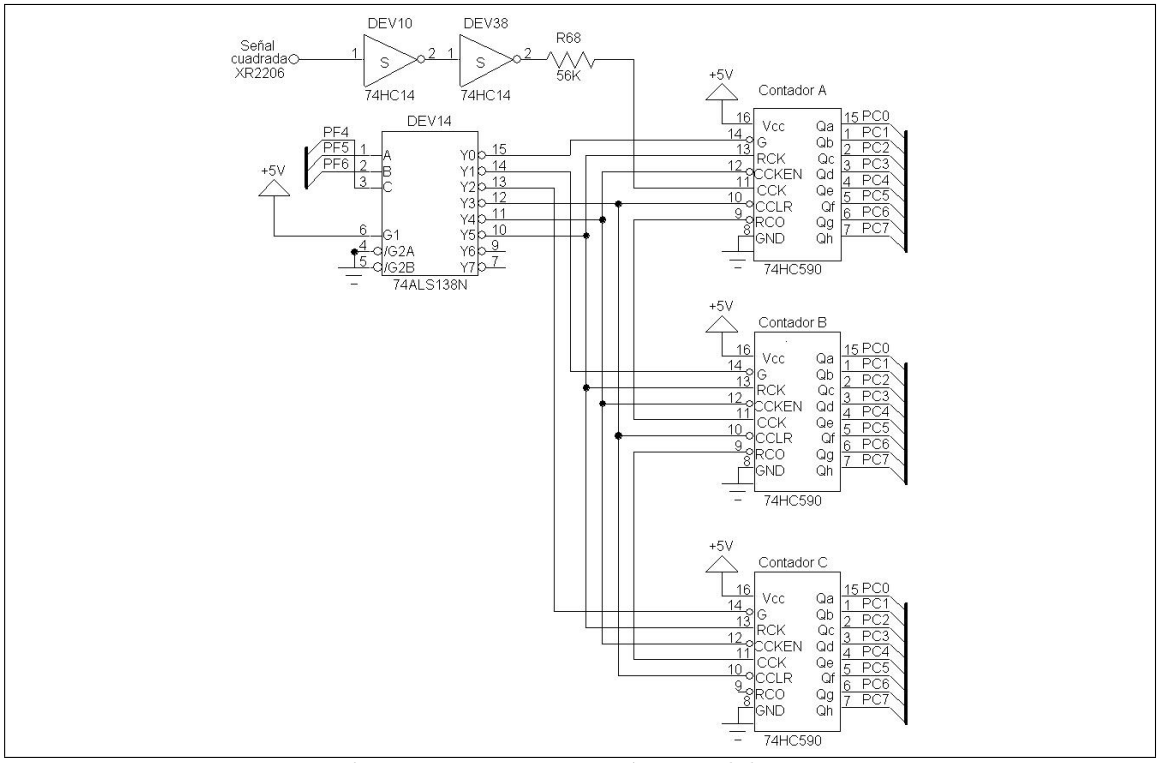


Figura 2.17: Frecuencímetro del TDN

- 512 bytes de EEPROM
- 1024 bytes de RAM
- Temporizador de 16 bits
- Circuito de interrupción en tiempo real
- Acumulador de pulsos de 8 bits
- Interfase de comunicación serial (SCI)
- 8 Canales del convertidor analógico digital
- 7 puertos paralelos (A, B, C, D, E, F y G)
- 96 Bytes de registros de control
- 256 Bytes de bootstrap ROM
- Frecuencia de operación de 2[MHz]
- 4 modos de operación (Bootstrap, Test, Single Chip, Expandido)

2.11. Control del TDN

El microcontrolador MCU68HC11F1 funciona como la interfase de comunicación del TDN y la computadora que se encarga de graficar la Traza de Nyquist. Mediante el HC11 se controlan también las diferentes tareas que lleva a cabo el TDN. A lo largo de este capítulo se ha mencionado que el HC11 interviene en el control del TDN. A continuación se presenta un resumen de las funciones de control del HC11.

- **Puerto Paralelo B $1004** Este puerto está conectado al DAC. Mediante este puerto se controla el barrido de la frecuencia una vez que se ha seleccionado el rango de la frecuencia de operación.
- **Puerto Paralelo C $1006** En este puerto está conectado el frecuencímetro del TDN. EL puerto C es bidireccional, sin embargo solo está configurado para leer los bytes de la frecuencia obtenida por el frecuencímetro.
- **Puerto Paralelo F $1005** Este puerto se utiliza para diferentes tareas. Los pines PF0, PF1, PF2 y PF3 están conectados a los circuitos que se encargan de seleccionar el rango de la frecuencia de operación del TDN. Los pines PF4, PF5 y PF6 están conectados al multiplexor que controla el arreglo de contadores del frecuencímetro. Mediante el pin PF7 se *resetea* el circuito detector de pico máximo en cada lectura del análisis.
- **El Convertidor Analógico Digital** Se utilizaron los canales ADR2, ADR3, ADR4 para tomar las lecturas de los valores obtenidos al realizar un análisis (Amplitud de la señal de entrada y una cantidad proporcional a las componentes Real e Imaginaria de los vectores $|G(j\omega)|\angle Gj\omega$ en el punto analizado respectivamente).
- **Interfase de comunicación serial** Mediante el puerto serie del HC11 se lleva a cabo la comunicación con el Software de Interfaz de Usuario. A partir del puerto serie se reciben los programas de control que posteriormente son cargados en la memoria RAM del HC11 y ejecutados. (ver capítulo 3).

Capítulo 3

Software de control

3.1. Introducción

El componente central del TDN es el microcontrolador 68HC11F1 de Motorola, ya que es el encargado de manejar los demás componentes del TDN.

El microcontrolador realiza sus tareas conforme a la ejecución de programas de cómputo. Los cuales, deben almacenarse en la memoria RAM del microcontrolador para poder ejecutarse. La carga de los programas en la memoria se realiza desde la PC por medio del puerto serie.

El presente capítulo describe el funcionamiento de los programas elaborados para el microcontrolador, así como la forma en que se ejecutan.

3.2. Ejecución de programas en el microcontrolador

El programa principal del microcontrolador es el "talker". Este se encuentra grabado en la memoria EEPROM, y se utiliza para recibir y ejecutar programas en el microcontrolador. Para ejecutar un programa en el microcontrolador se debe descargar desde la PC el código del mismo, para después iniciar su ejecución.

Se desarrollaron tres programas para el microcontrolador, con los cuales es posible obtener los datos relevantes para generar las trazas de Nyquist. Cada uno de estos programas ocupa aproximadamente 220 bytes de memoria para su ejecución así que, es posible colocar los tres programas en la memoria RAM interna del microcontrolador (el 68HC11F1 cuenta con 1024 bytes de memoria RAM), e indicarle al talker cuál de los programas ejecutar. Sin embargo, si en el futuro se requiere actualizar o agregar un nuevo programa; puede que éstos, sobrepasen la capacidad de la RAM interna del microcontrolador y sea necesario agregar una memoria RAM externa. Debido a lo anterior, se prefirió cargar en la memoria RAM el código del programa deseado, cada vez que se requiera su ejecución. Las acciones que debe llevar a cabo el microcontrolador para ejecutar un programa, de la

forma descrita, son las siguientes:

1. Se recibe por la interfaz SCI el código del programa y se almacena en la RAM.
2. Se instruye al talker para iniciar la ejecución del programa recibido.
3. Cuando el programa termina su ejecución le pasa el control al talker que queda en espera de un nuevo programa.

Las ventajas de usar este esquema son:

- Es posible ejecutar un número ilimitado de programas, uno después de otro, de manera que el microcontrolador no requiera de una memoria externa para almacenar los programas.
- Se reduce la complejidad de los programas que debe ejecutar el microcontrolador. Ya que cada uno lleva a cabo una tarea específica. Esto resulta mas adecuado en comparación a tener un solo programa que se encargue de todas las tareas.

3.3. El talker

El talker es un programa residente en el microcontrolador, que tiene como función principal el mantener la comunicación con la PC. El talker se encuentra grabado en la memoria EEPROM del microcontrolador, y está diseñado para operar en el modo single-chip del microcontrolador.

El talker está compuesto de dos partes: el programa principal y una rutina de interrupción. El programa principal realiza las siguientes operaciones:

1. Inicializa la pila del microcontrolador.
2. Inicializa la interfaz SCI a 9600 bps.
3. Ejecuta un ciclo infinito en donde prende y apaga el led testigo[1].

Cuando se recibe un byte por la interfaz SCI, la rutina de interrupción se activa y realiza las siguientes operaciones:

1. Verifica que la entrada a la interrupción se haya dado debido a la recepción normal de un byte.
2. Compara el byte recibido contra el codigo de los comandos reconocidos.

[1] La tarjeta Facil_11B tiene conectado al pin PD5 un led que tiene como finalidad el testificar la ejecución del talker

Código de cabecera	Contenido	Descripción
$00	Número de datos(1 Byte) Dirección(2 Bytes) Datos(de 1 a 256 bytes)	Colocar datos a partir de una dirección de memoria.
$01	Dirección(2 Bytes)	Iniciar ejecución desde una dirección de memoria.
$02	Ninguno	Enviar OK.
$03	Ninguno	Abortar ejecución.
$04	Ninguno	Enviar versión del talker.

Cuadro 3.1: Comandos del talker.

3. Pasa el control a la subrutina encargada de manejar cada comando.

Cabe mencionar que como el microcontrolador opera en el modo single-chip, la rutina de interrupción se activa al grabar la dirección de inicio de la misma en la dirección $FFD6, que es la dirección del vector de interrupción asociado con la interfaz SCI. De igual manera es necesario grabar en la dirección del vector RESET la dirección de inicio del programa principal del talker, que en este caso es la dirección de inicio de la memoria EEPROM ($FE00), para que al activar la tarjeta Facil_11B el talker se inicie automáticamente. El listado completo del talker se encuentra en el apendice A.1

3.3.1. Comandos del talker

El talker recibe sus comandos en forma de paquete. Cada paquete está formado por dos partes: la cabecera y el contenido. La cabecera, que es un byte, le indica al talker que operación debe realizar y el contenido, que pueden ser cero o más bytes, son los datos necesarios para la ejecución de cada comando. Los comandos reconocidos se resumen en la tabla 3.1.

A continuación se detalla el funcionamiento de cada comando:

- Comando $00: Colocar datos en la memoria.

  1. Recibe un byte que indica el número de datos a colocar en la memoria.
  2. Recibe dos bytes, que indican la dirección desde donde se deben colocar los datos. El primer y segundo byte son respectivamente la parte alta y baja de la dirección.
  3. Realiza un ciclo en el que recibe los datos y los coloca a partir de la dirección especificada en el paso anterior.

- Comando $01: Iniciar ejecución.

  1. Recibe dos bytes, que indican la dirección a donde se debe ejecutar un salto incondicional (jmp). El primer y segundo byte son respectivamente la parte alta y baja de la dirección.
  2. Reinicia la pila del microcontrolador.
  3. Efectua el salto incondicional.

- Comando $02: Enviar OK.

  1. Transmite el byte $0a .

- Comando $03: Abortar ejecución.

  1. Efectua un salto incondicional la dirección $FE00. La dirección anterior es el inicio del programa principal del talker.

- Comando $04: Versión del talker.

  1. Transmite el byte $01, inicio de paquete de datos.
  2. Transmite el byte $01, tamaño del paquete.
  3. Transmite el byte almacenado en la dirección $ffbf, que representa la versión del talker.

Es necesario mencionar que, el led testigo se activa al entrar a la rutina de interrupción, y se desactiva al terminar la ejecución de cada comando.

3.4. Software de arbitraje

Ya se ha explicado la forma en que el microcontrolador está a la espera de programas para ser ejecutados. Dichos programas llevan a cabo funciones especificas, básicamente dos: la calibración del TDN y el análisis de la respuesta en frecuencia del sistema de prueba. La comunicación entre el microcontrolador HC11 y el TDN se lleva a cabo mediante algunos de los puertos paralelos con los que cuenta el HC11. Los puertos que se utilizan son el B, C, E y F.

- El puerto B, configurado como puerto de salida, está conectado al Convertidor Digital Analógico del TDN. Mediante este puerto se puede variar el nivel del voltaje que controla la frecuencia de operación del TDN en rango que va de 0 a 3.12 [V]

- El puerto C, configurado como puerto de entrada, está conectado a la salida del frecuencimetro del TDN. Mediante este puerto, es posible obtener la frecuencia de operación del TDN.

- El puerto E, que en realidad son los canales de entrada del convetidor analógico digital del HC11. Mediante los canales ADR2/PE2 ADR3/PE3 y ADR4/PE4, obtenemos los valores de $|H(j\omega)|$, $X(j\omega)$ y $Y(j\omega)$ descritos en el capítulo 1.

- El puerto F, configurado como puerto de salida, tiene asignados los pines PF0, PF1, PF2 y PF3 al control de los rangos de la frecuencia de operación del TDN. Los pines PF4, PF5 y PF6 estan conectados al multiplexor del frecuencimetro del TDN. Mientras que mediante el PF7, es posible efectuar un reset al circuito detector de pico.

3.4.1. Proceso de calibración

Por proceso de calibración nos referimos al proceso mediante el cual se obtienen 18 valores de frecuencia diferentes de cada uno de los 4 rangos de operación del TDN. Estos valores son enviados posteriormente a la PC y es ahí en donde, utilizando un ajuste por mínimos cuadrados, se obtienen 4 ecuaciones de rectas (una para cada rango). Estas ecuaciones nos dan la relación que existe entre el byte que debemos poner en el convertidor digital analógico y la frecuencia de operación del TDN. Si se desea que el TDN operé a una frecuencia dada, es necesario que las ecuaciones byte-frecuencia describan lo mas exactamente posible el comportamiento de la frecuencia de operación del TDN, por ello es importante efectuar un proceso de calibración adecuado.

El programa que lleva a cabo esta tarea se llama: calibrar.asm. El listado completo del programa se muestra en el apéndice A. Calibrar.asm funciona, en términos generales, de la siguiente manera:

1. Se selecciona el rango de frecuencia de operación del TDN, mediante el puerto F (pines PF0, PF1, PF2, PF3)

2. Se coloca en el puerto B un $00 para que el TDN trabaje a su máxima frecuencia de operación dentro del rango seleccionado.

3. La frecuencia de operación se lee del frecuencimetro del TDN que entrega 3 bytes, uno por cada contador. Mediante el puerto F (pines PF4, PF5, PF6), se controla al multiplexor que controla la salida de los contadores. Se lee en el puerto C, cada uno de los 3 bytes, multiplexando la salida de los contadores.

4. Los 3 bytes de la frecuencia de operación se almacenan en la RAM del microcontrolador a partir de la dirección $0300.

5. Mediante a rutina *envlect*, se envian los 3 bytes obtenidos a la PC, el protocolo para enviar estos datos es el siguiente:

   a) Se manda el byte $0A para indicarle al programa que está siendo ejecutado en la PC que hay datos que van a ser enviados.

b) Se manda un byte indicando el tamaño en bytes del paquete en este caso siempre se enviara un $03 puesto que sólo son tres bytes.

c) Se manda el paquete de datos (el valor en bytes de la frecuencia).

6. Se incrementa en $0F el valor del byte que se coloca en el puerto B y se repiten los pasos 3,4 y 5. El ciclo se repite hasta que el valor del byte que se pone en el puerto B sea igual a $FF (byte que representa la minima frecuencia de operación del TDN). De está manera se obtienen las 18 muestras de uno de los 4 rangos de frecuencia de operación del TDN.

7. Se efectuan todos los pasos descritos, para el siguiente rango de frecuencia de operación del TDN. El ciclo se repite hasta obtener las 18 muestras de cada uno de los 4 rangos.

3.4.2. Proceso de análisis

El análisis de respuesta en frecuencia se puede llevar a cabo en dos escenarios distintos: analizando en un rango de frecuencias y analizando solo para una frecuencia en especial. A continuación describimos el proceso para efectuar el análisis para una frecuencia en particular.

El proceso de análisis en general consiste en aplicar una señal senoidal al sistema en análisis, para obtener sus componentes reales e imaginarias. El análisis de respuesta en frecuencia para una frecuencia en particular se efectúa con el programa: frec01. Los pasos que este lleva a cabo son los siguientes:

1. Selecciona el rango de frecuencias a utilizar, colocando el valor correspondiente en los puertos B y F.

2. Espera aproximadamente un segundo, mientras la señal generada se estabiliza.

3. Mide la frecuencia de la señal generada mediante la subrutina “medicion”.

4. Toma la lectura del convertidor analógico-digital.

5. Envía los datos de la frecuencia y los valores de los niveles del convertidor análogico-digital a la PC.

El análisis de respuesta en frecuencia en un rango de frecuencias se realiza con el programa: frec02. El cual, es similar al anterior y los pasos que realiza se describen a continuación:

1. Selecciona el rango de frecuencias a utilizar, colocando el valor correspondiente en los puertos B y F.

2. Espera aproximadamente un segundo, mientras la señal generada se estabiliza.
3. Restablece el circuito detector de pico.
4. Espera aproximadamente 20 milisegundos, mientras la señal generada se estabiliza.
5. Toma diez lecturas del convertidor analógico-digital y obtiene el promedio de las mismas.
6. Envía los datos obtenidos en el paso anterior a la PC.

El código completo de los programas descritos anteriormente se encuentran en el apéndice A.

Capítulo 4

Interfaz gráfica de usuario

4.1. Introducción

La interfaz gráfica de usuario es el medio por el cual el usuario hace uso del TDN. Mediante la interfaz, el usuario puede realizar el análisis de respuesta en frecuencia, para obtener la traza de Nyquist, y configurar algunos aspectos técnicos del TDN. El software de interfaz es un programa desarrollado en Visual Basic 5 que lleva por nombre Tradiny.

El diseño del software es orientado a objetos. A pesar de que el lenguaje de programación utilizado no tiene todas las características de un lenguaje orientado a objetos, como la herencia y el polimorfismo; empleamos los artefactos del mismo para simular los objetos del sistema. Por ejemplo, utilizamos una forma para representar una clase y a la vez a un objeto de la misma.

En el presente capítulo describimos el proceso de desarrollo de software utilizado elaborar el software Tradiny, y el diseño del mismo mediante el uso del lenguaje unificado de modelado (UML).

4.2. Proceso de desarrollo de software

Un proceso de desarrollo de software es un método para organizar las actividades relacionadas con la creación, presentación y mantenimiento de los sistemas de software. La descripción de un proceso de este tipo incluye, fundamentalmente, las actividades que abarcan desde los requerimientos hasta la presentación o entrega del producto. El proceso de desarrollo que utilizamos es una versión ligera del proceso de desarrollo unificado de *Rational* (RUP); ya que, como el software no es muy complejo, la utilización formal del proceso RUP haría más laboriosa y lenta la realización del mismo.

Desde una perspectiva de alto nivel los pasos del proceso de desarrollo utilizado son los siguientes:

1. *Planeación y elaboración*: planear y definir los requerimientos, construir prototipos, etc.
2. *Construcción*: la creación del sistema.
3. *Aplicación*: la transición de la implementación del sistema a su uso.

4.2.1. Ciclo de vida en cascada

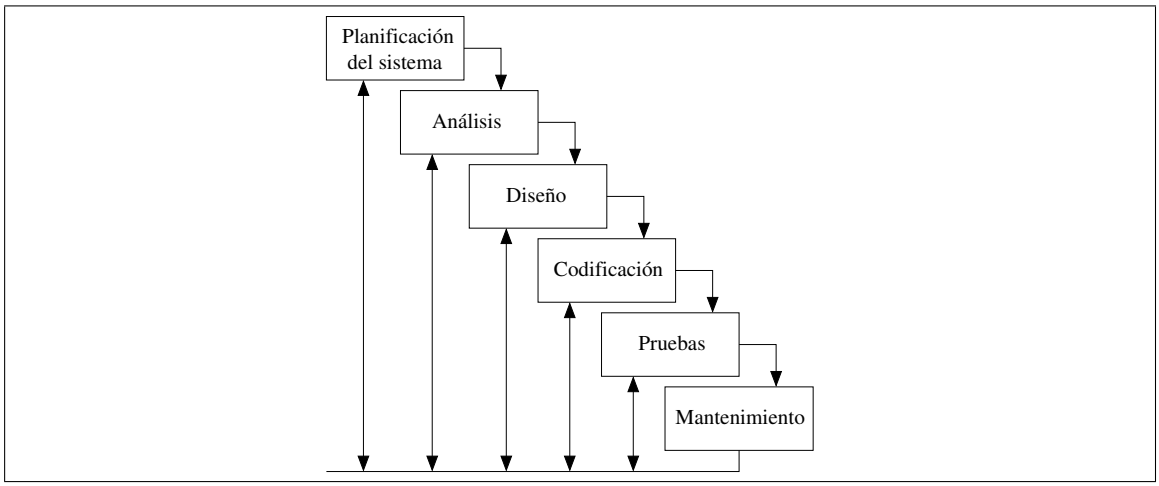


Figura 4.1: Ciclo de vida en cascada.

El modelo en cascada es el más antiguo de todos, y sirve como base para otros. El modelo contiene una serie de etapas que no se solapan, y el proyecto se va revisando tras cada una de las etapas. Para poder pasar a la siguiente etapa se tiene que haber conseguido todos los objetivos de la etapa anterior, es un proceso secuencial.

Tiene una buena aplicación cuando el problema es estable y cuando se trabaja con metodologías técnicas conocidas. Este modelo será apropiado para la migración de una aplicación o a una versión de mantenimiento bien definida.

Con este modelo se tiene un seguimiento de todas las fases del proyecto y del cumplimiento de todos los objetivos marcados en cada etapa tanto de costes, fechas de entrega y lo más importante, que pueden comprobar al final de cada etapa si el proyecto cumple todas las necesidades del usuario.

Desventajas del ciclo de vida en cascada

La característica distintiva de un ciclo de desarrollo en cascada es que contiene un solo paso de análisis, diseño y construcción; en un único paso se realiza todo el análisis, luego todo el diseño y finalmente todas las pruebas. Las principales desventajas de este enfoque son:

- Carga frontal de gran complejidad: excesiva complejidad.
- Retraso de la retroalimentación.
- Congelamiento temprano de las especificaciones, mientras cambia el dominio del problema.

4.2.2. Ciclo de vida iterativo

El ciclo de vida iterativo se basa en el agrandamiento y perfeccionamiento secuencial de un sistema a través de múltiples ciclos de desarrollo de análisis, diseño, implementación y pruebas. En el enfoque de desarrollo iterativo: el sistema crece al incorporar nuevas funciones en cada ciclo de desarrollo. De esta manera, tras una fase preliminar de planeación y especificación, el desarrollo pasa a la fase de construcción a través de una serie de ciclos de desarrollo.

El desarrollo iterativo no significa hacer una pausa, alcanzar una meta importante y luego volver a detenerse. Una definición más exacta del desarrollo iterativo es la siguiente: es un proceso planeado que consiste en revisar varias veces un área, mejorando el sistema en cada revisión. Se trata de un proceso formalmente planeado y programado; de ninguna manera fortuito.

Ventajas del ciclo de vida iterativo

Encontraste con el ciclo de vida en cascada, el ciclo iterativo se funda en el perfeccionamiento gradual de un sistema a través de múltiples ciclos de análisis, diseño, etc. Los beneficios de esto son:

- La complejidad nunca resulta abrumadora, porque en un ciclo sólo se abordan unidades cuya complejidad es manejable.
- Se genera retroalimentación en las fases iniciales, porque la implementación ocurre rápidamente en un subconjunto pequeño del sistema. Esta retroalimentación genera información para el análisis de ciclos subsecuentes.
- El equipo de desarrolladores podrá reaplicar gradualmente su dominio de las herramientas; es decir, no es necesario que desde un principio el programador se sirva de las mejores y más complejas características del lenguaje, ni tampoco que se sirva exclusivamente de técnicas no dominadas para dividir el sistema.
- Los requerimientos son ajustables para responder a las necesidades cambiantes del dominio del problema, a medida que avanza el proyecto.

El proceso de desarrollo que utilizamos se basa en el ciclo de desarrollo iterativo, debido a que el cambio dinámico de los requerimientos se observó desde el inicio del proyecto.

Un caso de uso es una descripción narrativa de un proceso de dominio; dicho de manera coloquial: los casos de uso representan las cosas que podemos hacer con el sistema. Los casos de uso tienen un papel importante en el proceso, ya que los ciclos iterativos de desarrollo los organizamos a partir de los requerimientos del caso de uso.

4.2.3. La fase de la planeación y la elaboración

Esta fase del proyecto incluye la concepción inicial, la investigación de alternativas, la planeación, la especificación de requerimientos y otras actividades. En la figura 4.2 observamos algunas de las actividades de esta fase.

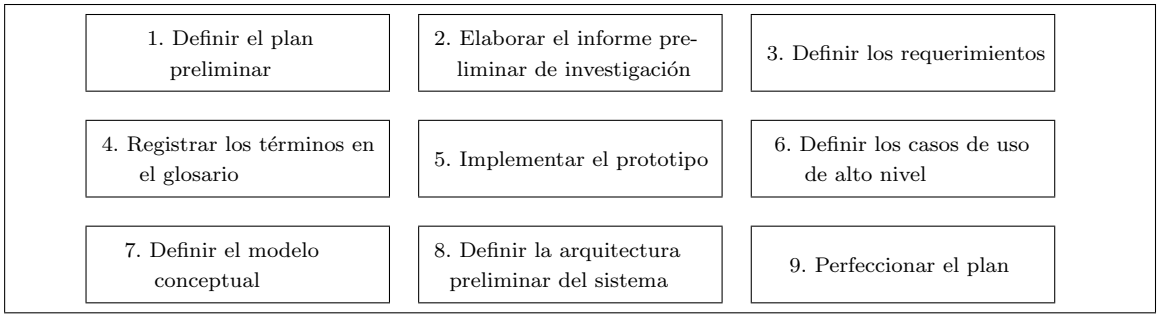


Figura 4.2: Ejemplo de las actividades de la fase de planeación y elaboración.

Los artefactos generados en esta fase son:

1. Especificación de los requerimientos: los requerimientos son una descripción de las necesidades o deseos de un producto. La finalidad es identificar y documentar lo que en realidad se necesita.
2. Glosario: diccionario del dominio del problema.
3. Casos de uso de alto nivel: descripciones narrativas de los procesos de dominio.
4. Modelo conceptual preliminar: es un bosquejo cuya finalidad es facilitar el conocimiento del vocabulario del dominio, especialmente en su relación con los casos de uso y las especificaciones de los requerimientos.
5. Prototipo del sistema: prototipo básico con el fin de facilitar la comprensión del problema.

4.2.4. La fase de construcción

La fase de construcción del proyecto requiere varios ciclos de desarrollo a lo largo de los cuales se extiende el sistema. El objetivo final es obtener un sistema funcional de software que atienda debidamente los requerimientos. En cada ciclo de desarrollo, los principales pasos se analizan y diseñan, como se muestra en las figuras 4.3 y 4.4.

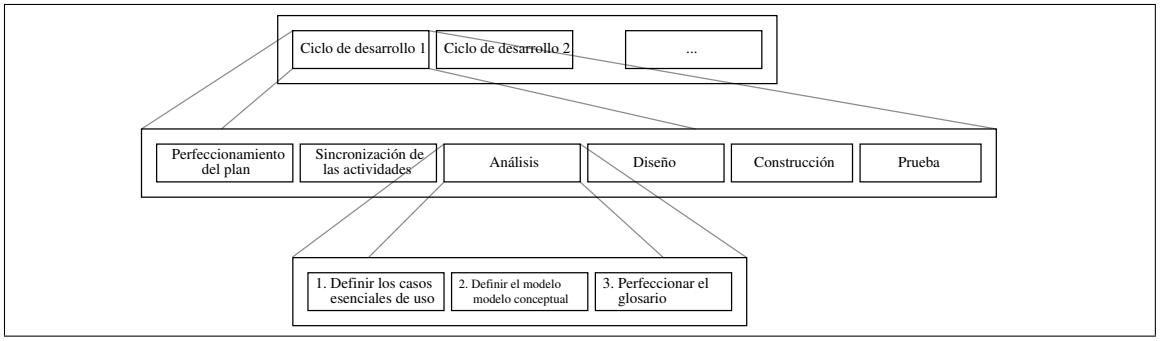


Figura 4.3: Ejemplo de las actividades de la fase de análisis.

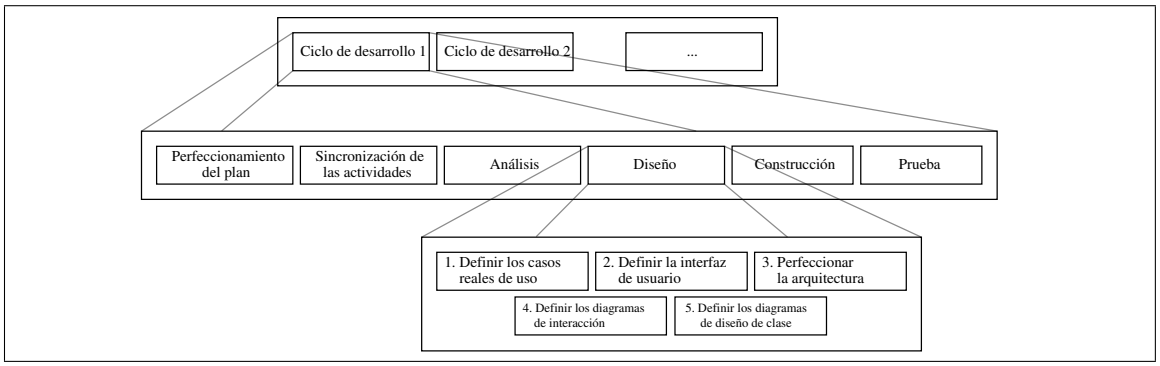


Figura 4.4: Ejemplo de las actividades de la fase de diseño.

4.2.5. La fase de aplicación

En esta fase se realiza la transición de la implementación del sistema al uso del mismo. Es aquí donde se presenta la aplicación desarrollada y donde se muestra al cliente la forma en que esta opera. Regularmente esta fase incluye la comercialización del producto terminado y la capacitación al cliente; sin embargo, para los fines de esta tesis en esta fase desarrollamos un pequeño manual de operación que indica la forma básica de utilizar el sistema elaborado. El manual de operación se encuentra en el capítulo C: Manual de operación.

4.3. Artefactos elaborados

Como se mencionó anteriormente, en la primera fase del proceso se generan ciertos artefactos que facilitan la comprensión del problema. El orden de creación de los mismos no es estrictamente como se planteó, ya que algunos artefactos pueden elaborarse paralelamente con los demás; además, no es forzoso crear todos los artefactos mencionados, sino solo los de mayor utilidad.

Para la descripción del software elaborado consideramos que los artefactos listados a

continuación, son suficientes para comprender el alcance y funcionamiento del mismo.

- Requerimientos
- Casos de uso
- Diagramas de secuencia
- Modelo conceptual
- Modelo de clases
- Diagrama de distribución
- Glosario

En lo que resta del presente capítulo describimos los artefactos anteriores en el orden que aparecen listados.

4.3.1. Requerimientos

Los requerimientos encontrados durante el análisis del dominio del TDN se resumen en los cuadros 4.1 y 4.2. En los cuadros citados aparece una columna que especifica la categoría de la función en evidente u oculta. Evidente, significa que la función debe realizarse y el usuario debería saber que se ha realizado; mientras que, oculta significa que la función debe realizarse aunque no sea visible para el usuario.

Ref #	Función	Categoría
1.1	Mostrar el progreso del análisis en ejecución.	Evidente
1.2	Permitir la cancelación de un análisis durante su ejecución.	Evidente
1.3	Desplegar una gráfica con los datos obtenidos durante el análisis al final del mismo.	Evidente
1.4	Permitir al usuario analizar la gráfica obtenida punto por punto.	Evidente
1.5	Ofrecer un mecanismo de almacenamiento y recuperación de los análisis efectuados.	Evidente

Cuadro 4.1: Funciones básicas

Ref #	Función	Categoría
2.1	Debe identificar los puertos serie libres.	Evidente
2.2	Debe mostrar el progreso de la calibración.	Evidente
2.3	Permitir la cancelación durante su realización.	Evidente
2.4	Almacenar los cambios en la configuración.	Oculta

Cuadro 4.2: Funciones de configuración

Los atributos del sistema son sus características o dimensiones; no son funciones, pero pueden ser específicas para algunas de ellas. Los atributos del sistema se resumen en el cuadro 4.3.

Ref #	Atributo	Detalles
1.1, 1.2, 1.3, 1.4, 2.2, 2.3	Tolerancia a fallas.	Debe verificar la comunicación con el TDN antes y durante el proceso.
	Metáfora de la interfaz	Pantallas basadas en el esquema *Multiple Document Interface* (MDI)
	Sistema operativo	*Microsoft Windows* 95, 98, Me, 2000 y XP
	Tiempo de respuesta	Percepción inmediata, 5 segundos máximo.

Cuadro 4.3: Atributos del sistema

4.3.2. Casos de uso

El caso de uso es un documento narrativo que describe la secuencia de eventos de un actor, usuario, agente externo, que utiliza un sistema para completar un proceso. Los casos de uso son historias o casos de utilización de un sistema; ejemplifican e incluyen tácitamente los requerimientos en las historias que narran.

En la figura 4.5 se muestran los casos de uso reales agrupados por funcionalidad, así como los actores que intervienen en ellos. Un caso de uso real describe concretamente el proceso a partir de su diseño concreto actual.

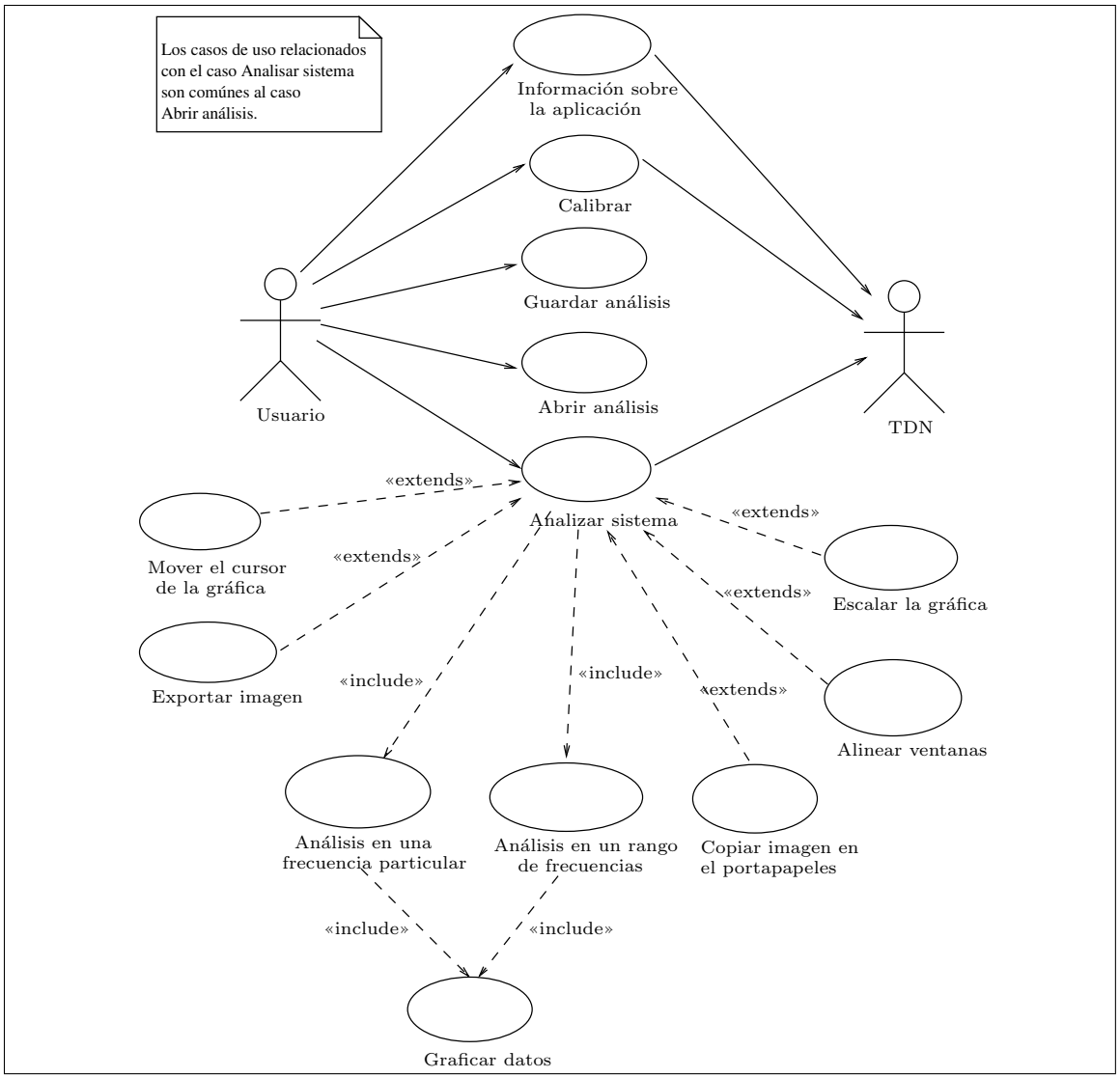


Figura 4.5: Diagrama de casos de uso.

Además de los casos de uso, el UML incluye entre su notación a los diagramas de interacción, los cuales explican gráficamente las interacciones entre los objetos del sistema. El UML define dos tipos de estos diagramas: los de colaboración y los de secuencia. Los diagramas de colaboración describen las interacciones entre los objetos usando un formato de grafo, mientras que los diagramas de secuencia emplean un formato de cerca o muro. Estos diagramas se elaboran para un escenario particular de un caso de uso: el flujo normal de eventos; aunque si es necesario, se pueden elaborar para los escenarios alternos más interesantes. Ambos diagramas representan la misma información; para los propósitos del presente trabajo emplearemos los diagramas de secuencia por considerarlos más fáciles de interpretar.

A continuación se lista la descripción de cada caso de uso identificado seguido de su respectivo diagrama de secuencia.

1. Analizar sistema.

1.1. Breve descripción.

Es el caso en el que el usuario inicia un nuevo análisis.

1.2. Flujo de eventos

1.2.1. Comienza cuando el usuario presiona el botón “Nuevo” o selecciona la opción “Nuevo” desde el menú “Analizar”.

1.2.2. El sistema muestra la forma frmAnalisis.

1.2.3. Si el usuario selecciona la opción “Análisis para una frecuencia particular” inicia el caso de uso 2..

1.2.4. Si el usuario selecciona la opción “Análisis en un rango de frecuencias” inicia el caso de uso 3..

1.2.5. Si el usuario presiona el botón “Cancelar” comienza el flujo alterno 1.3.1.

1.2.6. Si el análisis realizado falló, comienza el flujo alterno 1.3.2.

1.2.7. El sistema crea una gráfica y la despliega.

1.2.8. Comienza el caso de uso “Graficar datos”.

1.2.9. Fin del caso de uso.

1.3. Flujos alternos

1.3.1. Cancelar caso de uso.

a) Se oculta la forma frmAnalisis.

b) Se establece que el análisis no se efectuó.

c) Termina el caso de uso.

1.3.2. Análisis fallido.

a) Se oculta la forma frmAnalisis.

b) Se establece que el análisis no se efectuó.

c) Termina el caso de uso.

1.4. Poscondiciones

1.4.1. Se debe incrementar la cuenta de los análisis realizados por el usuario.

1.4.2. Los menús “Edición” y “Ventana” deben ser visibles para los usuarios.

1.4.3. Las opciones “Guardar” y “Exportar” deben ser habilitadas.

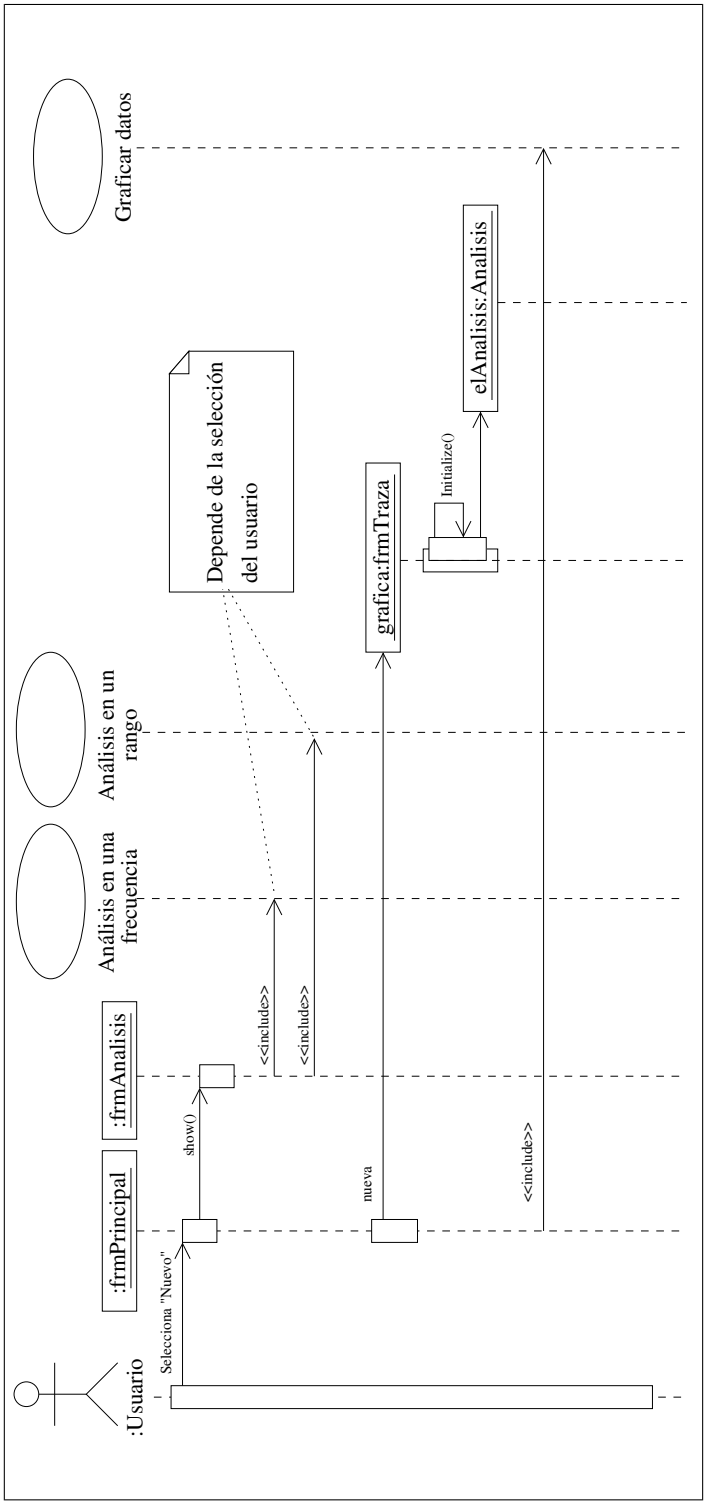


Figura 4.6: Diagrama de secuencia del caso de uso 1.

2. Análisis en una frecuencia particular.

2.1. Breve descripción.

Es el caso en el que el usuario desea analizar un sistema eléctrico, aplicando una señal de frecuencia fija establecida por él mismo.

2.2. Flujo de eventos

2.2.1. Comienza cuando el usuario selecciona la opción "Análisis para una frecuencia particular" en la forma frmAnalisis.

2.2.2. El usuario introduce un valor de frecuencia en la caja de texto etiquetada como "Frecuencia [Hz]" y presiona el botón ejecutar. Si el valor de frecuencia introducido está fuera de los límites de operación del TDN comienza flujo alterno 2.3.1.

2.2.3. El valor de frecuencia es comparado contra los límites de rango para identificar el rango a seleccionar.

2.2.4. El valor de frecuencia es evaluado en la función de frecuencia del rango escogido, para determinar el byte a colocar en el CDA.

2.2.5. Se envía al TDN el programa "frec01.s19".

2.2.6. Se escribe en la memoria del TDN el byte para el rango y el byte para el CDA. Si la comunicación con el TDN falla comienza flujo alterno 2.3.2.

2.2.7. El sistema indica al TDN iniciar la ejecución del programa almacenado. Si la comunicación falla comienza el flujo alterno 2.3.2.

2.2.8. El sistema espera la respuesta del TDN, 6 bytes. Si la comunicación falla inicia flujo alterno 2.3.2.

2.2.9. Los 6 bytes recibidos, se convierten a valores reales para obtener las componentes X y Y, el valor de la amplitud y la frecuencia real generada y se almacenan como un análisis.

2.2.10. Fin del caso de uso

2.3. Flujos alternos

2.3.1. Frecuencia fuera de límites.

a) El sistema despliega un cuadro de mensaje indicando los límites de frecuencia válidos.

b) El sistema establece que el análisis falló.

c) Termina el caso de uso.

2.3.2. Falla en la comunicación.

a) El sistema despliega un cuadro de mensaje indicando la falla en la comunicación y una sugerencia sobre su solución.

b) El sistema establece que el análisis falló.

c) Termina el caso de uso.

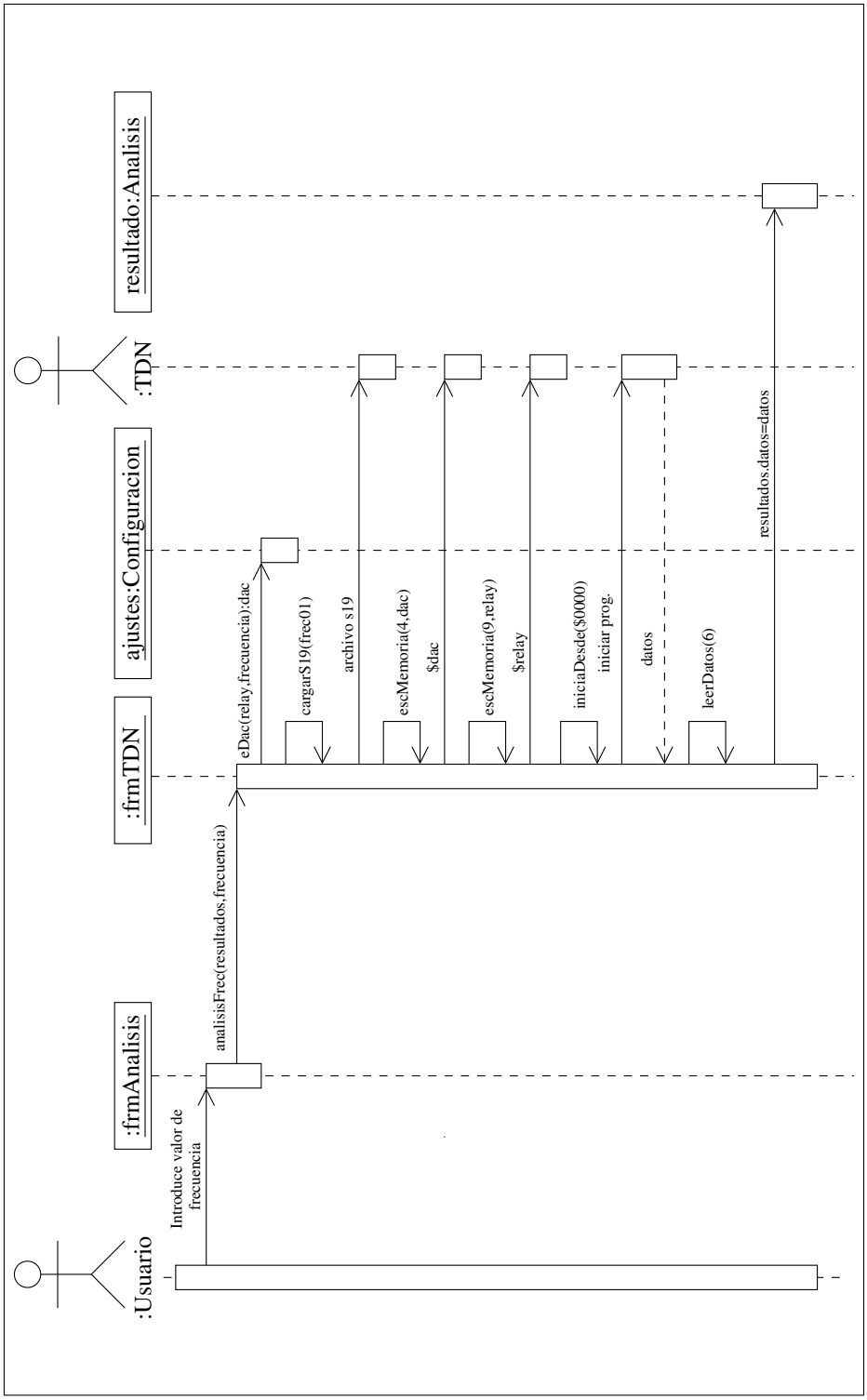


Figura 4.7: Diagrama de secuencia del caso de uso 2.

3. Análisis en un rango de frecuencias.

3.1. Breve descripción.

Es el caso en el que el usuario desea analizar un sistema eléctrico, aplicando una señal que varía en frecuencia dentro de un límite establecido por él mismo.

3.2. Flujo de eventos

3.2.1. Comienza cuando el usuario selecciona la opción "Análisis en un rango de frecuencias" en la forma frmAnalisis.

3.2.2. El usuario selecciona los límites inferior y superior ya sea en las listas desplegables etiquetadas como "Desde" y "hasta" respectivamente, o mediante la opción "Personalizado" donde introduce los límites manualmente en las cajas de texto contiguas a la opción mencionada. Si los límites introducidos están fuera de los límites reales, comienza el flujo alterno 3.3.1.

3.2.3. El sistema compara los límites introducidos contra los límites reales para determinar los rangos en los que deberá realizarse el análisis y el número de fases necesarias para efectuarlo.

3.2.4. El sistema envía al TDN el programa "frec02.s19". Si la comunicación con el TDN falla, comienza el flujo alterno 3.3.2.

3.2.5. El sistema indica al TDN iniciar el análisis para cada rango y fase.

3.2.6. El sistema espera por los resultados del análisis y actualiza la forma frmBarra conforme el análisis se efectúa.

3.2.7. Los datos recibidos se convierten en valores de voltaje y se almacenan como un análisis.

3.2.8. Si el usuario escogió la opción "Suavizar curva" en la forma "Análisis" entonces el análisis se ajusta para obtener una curva más definida.

3.2.9. Los datos del análisis se discriminan para escoger únicamente los puntos representativos de los límites solicitados por el usuario.

3.2.10. Fin del caso de uso.

3.3. Flujos alternos

3.3.1. Frecuencia fuera de límites.

a) El sistema despliega un cuadro de mensaje indicando los límites de frecuencia válidos.

b) El sistema establece el resultado del análisis como fallido.

c) Termina el caso de uso.

3.3.2. Falla en la comunicación.

a) El sistema despliega un cuadro de mensaje indicando la falla en la comunicación y una sugerencia sobre su solución.

b) El sistema establece el resultado del análisis como fallido.

c) Termina el caso de uso.

3.3.3. Cancelado por el usuario.

a) El sistema oculta la forma frmBarra.

b) El sistema despliega un cuadro de mensaje indicando la falla en la comunicación y una sugerencia sobre su solución.

c) El sistema establece el resultado del análisis como fallido.

d) Termina el caso de uso.

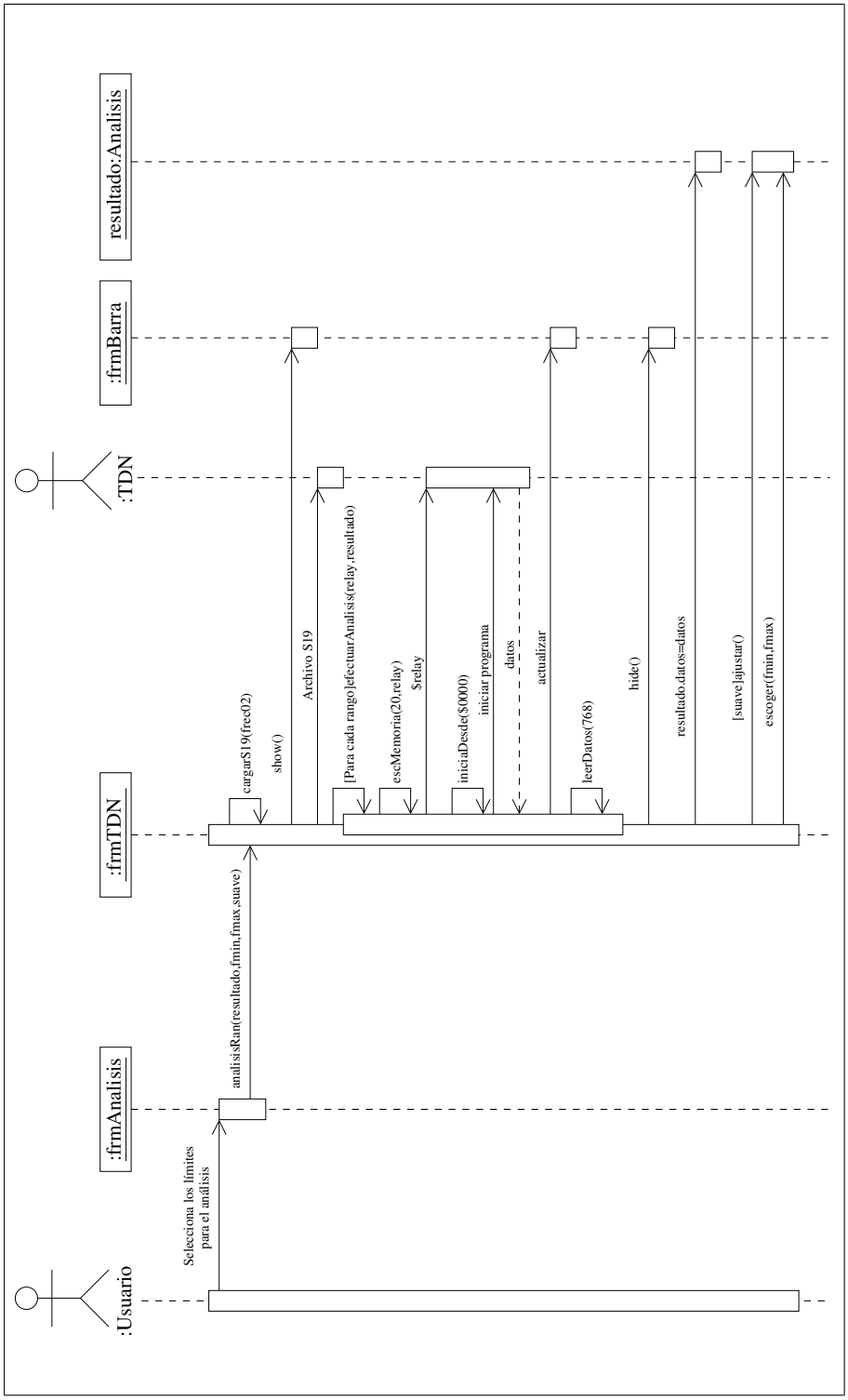


Figura 4.8: Diagrama de secuencia del caso de uso 3.

4. Abrir análisis.

4.1. Breve descripción.

Es el caso en el que el usuario desea abrir un análisis guardado anteriormente, para observar los resultados del mismo.

4.2. Flujo de eventos

4.2.1. Comienza cuando el usuario presiona el botón "Abrir" en la forma frmPrincipal o selecciona la opción "Abrir..." en el menú "Análisis".

4.2.2. Se despliega el cuadro de dialogo común "Abrir análisis en frecuencia...", mostrando el directorio desde donde se inició la aplicación o mostrando el directorio del último lugar donde se abrió un análisis.

4.2.3. El usuario escoge el directorio y el nombre del archivo que desea abrir y presiona el botón "Abrir". Si el usuario presiona el botón "Cancelar" se inicia el flujo alterno 4.3.1.

4.2.4. El sistema crea una nueva gráfica y la despliega.

4.2.5. El sistema lee el análisis contenido en el archivo seleccionado por el usuario.

4.2.6. Comienza el caso de uso "Graficar datos".

4.2.7. Fin del caso de uso.

4.3. Flujos alternos

4.3.1. Cancelar caso de uso.

a) Se oculta el cuadro de dialogo común "Abrir análisis en frecuencia...".

b) Se establece que el análisis no se abrió.

c) Termina el caso de uso.

4.4. Poscondiciones

4.4.1. El sistema incrementa el contador que testifica el número de análisis realizados.

4.4.2. El sistema vuelve visibles los menús: Edición y Ventana.

4.4.3. El sistema activa las opciones "Guardar" y "Exportar".

4.4.4. El sistema establece que el análisis en pantalla no requiere ser guardado al cerrarse.

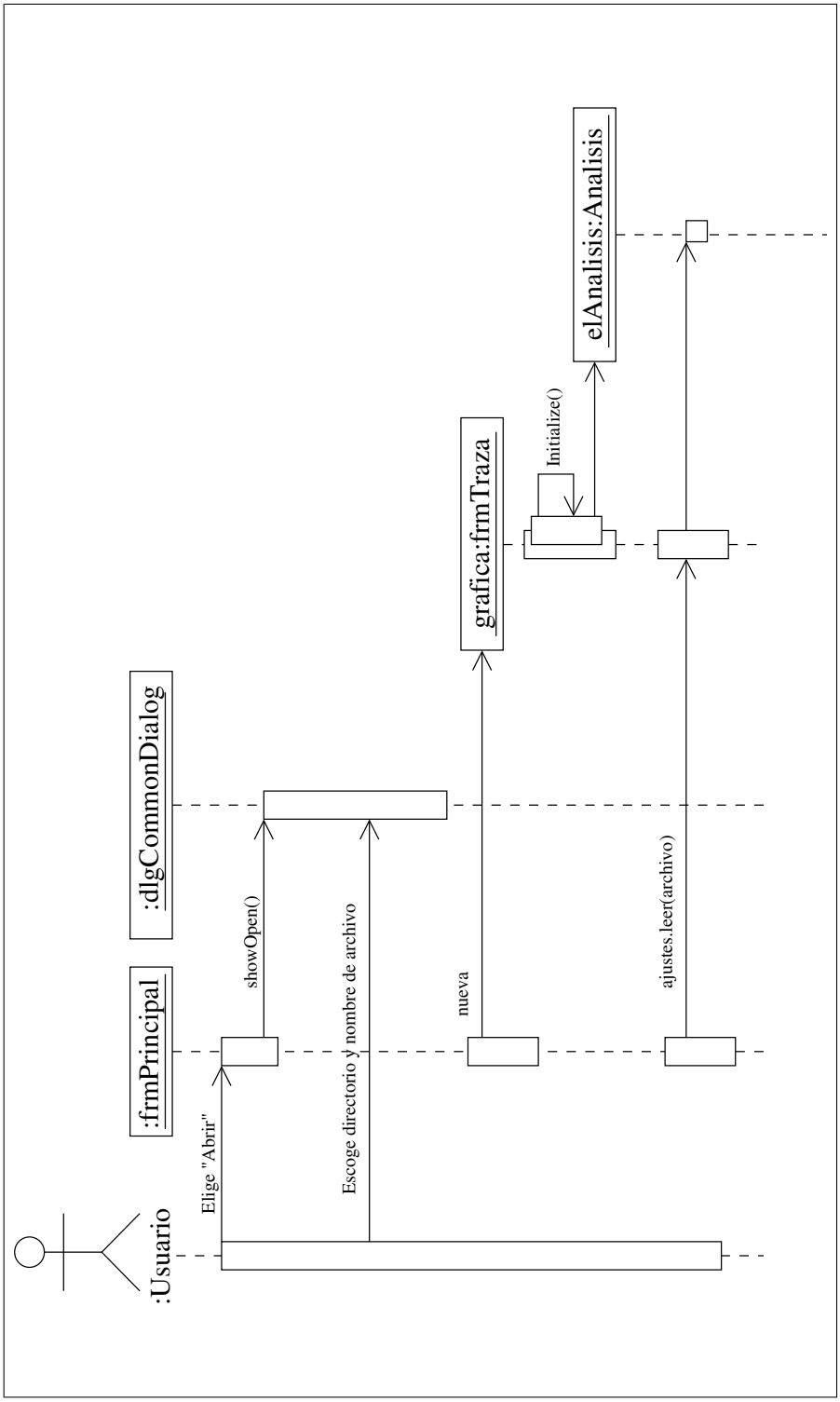


Figura 4.9: Diagrama de secuencia del caso de uso 4.

5. Guardar análisis.

5.1. Breve descripción.

Es el caso en el que el usuario desea guardar en un archivo los resultados del análisis actual.

5.2. Flujo de eventos

5.2.1. Comienza cuando el usuario presiona el botón “Guardar” en la forma frmGuardar o selecciona la opción “Guardar...” en el menú “Análisis”.

5.2.2. Se despliega el cuadro de dialogo común “Guardar análisis actual como...”, mostrando el directorio desde donde se inició la aplicación o mostrando el directorio del último lugar donde se guardó un análisis.

5.2.3. El usuario escoge el directorio y el nombre del archivo con el que desea guardar el análisis actual, y presiona el botón “Guardar”. Si el usuario presiona el botón “Cancelar” se inicia el flujo alterno 5.3.1.

5.2.4. El análisis es guardado en el archivo selccionado por el usuario. Si ocurre algún error al momento de guardar el análisis se inicia el flujo alterno 5.3.2.

5.2.5. Fin del caso de uso.

5.3. Flujos alternos

5.3.1. Cancelar caso de uso.

a) Se oculta el cuadro de dialogo común “Guardar análisis actual como...”.

b) Se establece que el análisis no se guardó.

c) Termina el caso de uso.

5.3.2. Error al guardar archivo.

a) Se oculta el cuadro de dialogo común “Guardar análisis actual como...”.

b) Se despliega un cuadro de mensaje indicando el tipo de error encontrado.

c) Se establece que el análisis no se guardó.

d) Termina el caso de uso.

5.4. Precondiciones

5.4.1. Debe haber una gráfica desplegada en la pantalla.

5.5. Poscondiciones

5.5.1. El sistema establece que el análisis ya no requiere ser guardado al cerrarse.

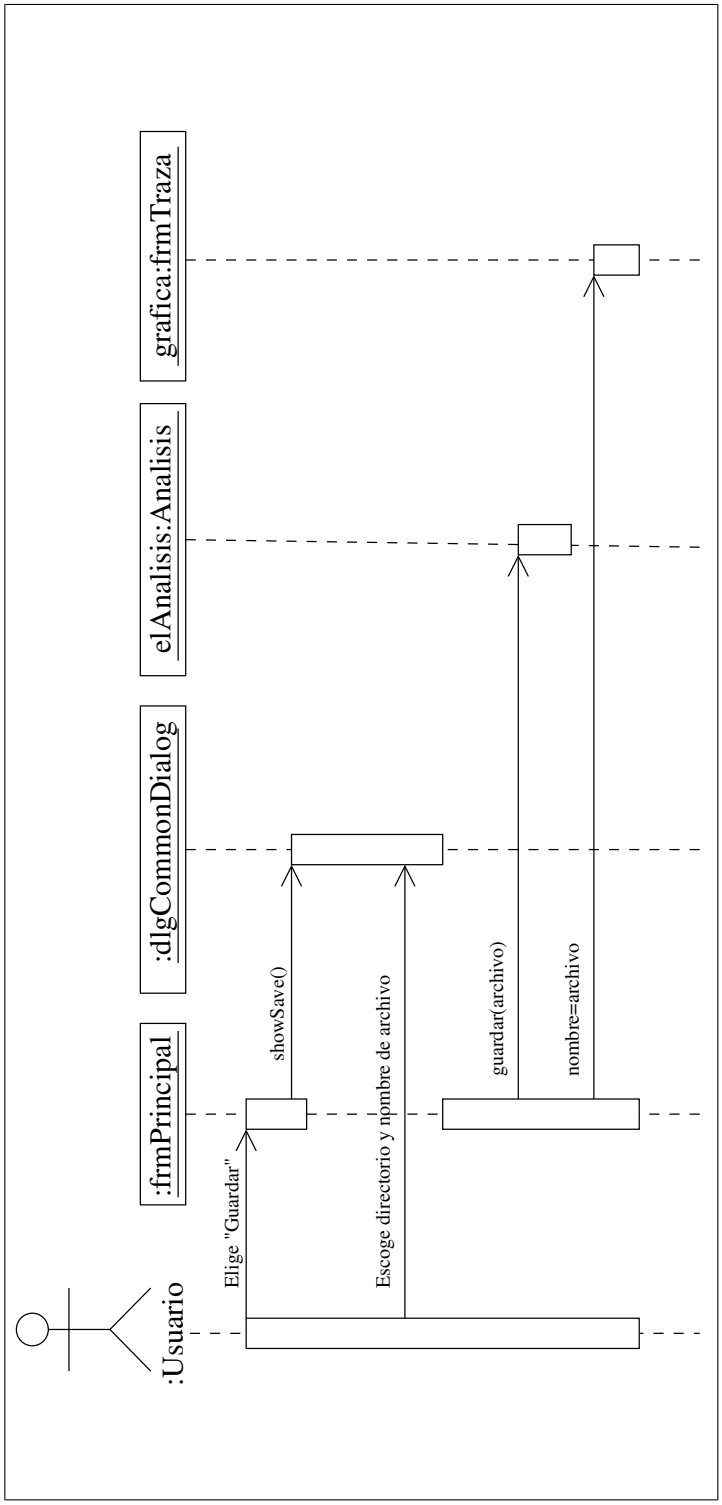


Figura 4.10: Diagrama de secuencia del caso de uso 5.

6. Graficar datos.

6.1. Breve descripción.

Este caso de uso se incluye en los casos 1. y 4., y sucede cuando la aplicación tiene listos los datos del análisis y procede a graficarlos en la forma activa.

6.2. Flujo de eventos

6.2.1. El caso de uso comienza cuando el sistema graficará los datos de un análisis.

6.2.2. El sistema dibuja los ejes coordenados en la gráfica actual.

6.2.3. El sistema examina los datos del análisis, y si contienen solo un punto comienza el flujo alterno 6.3.1

6.2.4. El sistema grafica los datos del análisis.

6.2.5. Fin del caso de uso.

6.3. Flujos alternos

6.3.1. Graficar un punto.

a) El punto del análisis se grafica como un vector: un circulo en el punto preciso y una línea del punto al origen de la gráfica.

b) Termina el caso de uso.

6.4. Precondiciones

6.4.1. Debe haber una gráfica desplegada en la pantalla.

6.5. Poscondiciones

6.5.1. Se establece que el análisis graficado debe guardarse antes de cerrarse.

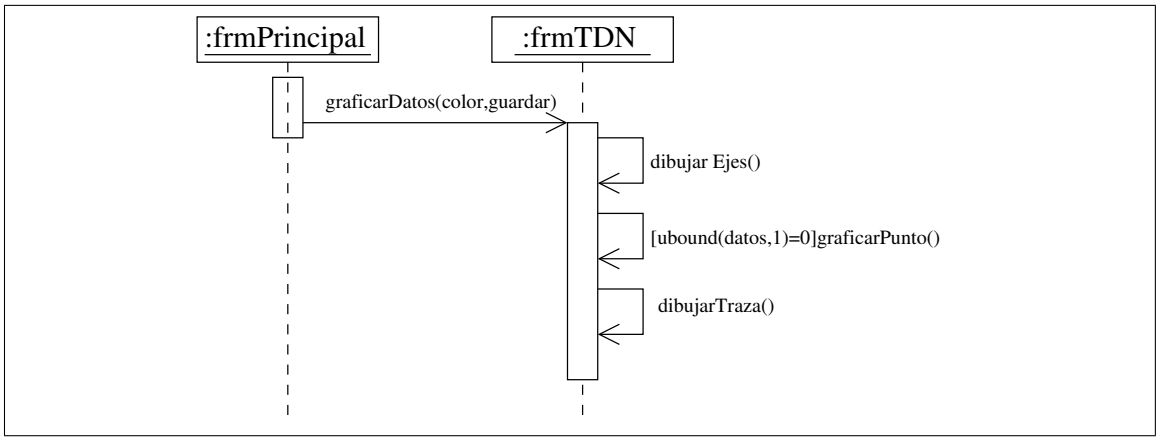


Figura 4.11: Diagrama de secuencia del caso de uso 6.

7. Calibrar.

7.1. Breve descripción.

Es el caso cuando el usuario decide calibrar el TDN.

7.2. Flujo de eventos

7.2.1. El caso de uso comienza cuando el usuario selecciona la opción “Calibrar...” en el menú “Trazador”, o cuando la aplicación es ejecutada por primera vez.

7.2.2. El sistema envía al TDN el programa “calibrar.s19”. Si la comunicación con el TDN falla comienza el flujo alterno 7.3.2.

7.2.3. El sistema despliega la forma frmBarra con el título “Calibración en proceso...”

7.2.4. El sistema envía al TDN la orden de iniciar la ejecución del programa almacenado.Si la comunicación con el TDN falla, comienza el flujo alterno 7.3.2.

7.2.5. El sistema espera a que el TDN le envíe 432 bytes de información y conforme la información se recibe, la forma frmBarra se actualiza para testificar el avance del proceso. Si la comunicación con el TDN falla, comienza el flujo alterno 7.3.2. Si el usuario presiona el botón “Cancelar” en la forma frmBarra, comienza el flujo alterno 7.3.1.

7.2.6. La información recibida se procesa para obtener los ajustes del trazador: límites superior e inferior de cada rango; los factores de corrección de los canales X y Y; y los factores de las ecuaciones características de cada rango.

7.2.7. El sistema oculta la forma frmBarra y actualiza los datos del archivo “ajustes.ini” con los ajustes obtenidos.

7.2.8. Fin del caso de uso.

7.3. Flujos alternos

7.3.1. Cancelar caso de uso.

a) El sistema oculta la forma frmBarra.

b) El sistema despliega un cuadro de mensaje indicando la falla en la comunicación y una sugerencia sobre su solución.

c) El sistema establece el resultado de la calibración como fallido.

d) Termina el caso de uso.

7.3.2. Falla en la comunicación.

a) El sistema oculta la forma frmBarra.

b) El sistema despliega un cuadro de mensaje indicando la falla en la comunicación y una sugerencia sobre su solución.

c) Se establece el resultado de la calibración como fallido.

d) Termina el caso de uso.

7.4. Precondiciones

7.4.1. No debe haber un análisis en progreso.

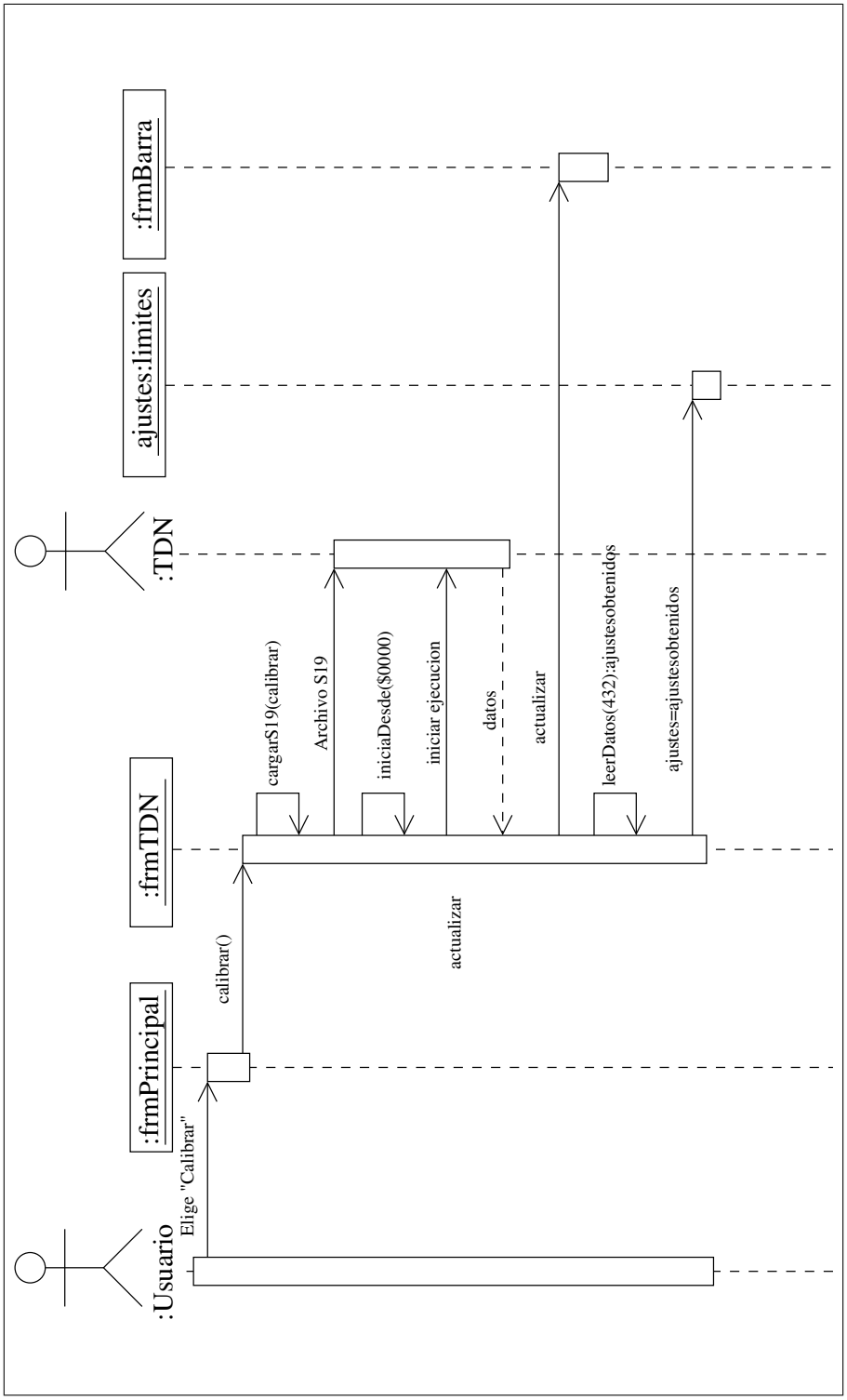


Figura 4.12: Diagrama de secuencia del caso de uso 7.

8. Configuración de comunicaciones.

8.1. Breve descripción. Es el caso en el que el usuario cambia la configuración del puerto de comunicación serie.

8.2. Flujo de eventos

8.2.1. El caso de uso inicia cuando el usuario selecciona la opción “Opciones...” en el menú “Ver”, o cuando la aplicación se ejecuta por primera vez.

8.2.2. El sistema despliega la forma frmConfig.

8.2.3. El usuario selecciona uno de los puertos disponibles de la lista desplegable etiquetada como “Puerto serie” y presiona el botón “Aceptar”.

8.2.4. El puerto escogido por el usuario se utiliza durante el resto de la sesión. Si el puerto esta siendo ocupado por otra aplicación, comienza el flujo alterno 8.3.1

8.2.5. La forma frmConfig se oculta ella misma.

8.2.6. Termina el caso de uso.

8.3. Flujos alternos

8.3.1. Puerto serie ocupado.

a) El sistema despliega un cuadro de mensaje indicando que el puerto esta ocupado y solicita al usuario escoger otro puerto diferente.

b) El flujo continua en el paso 8.2.3

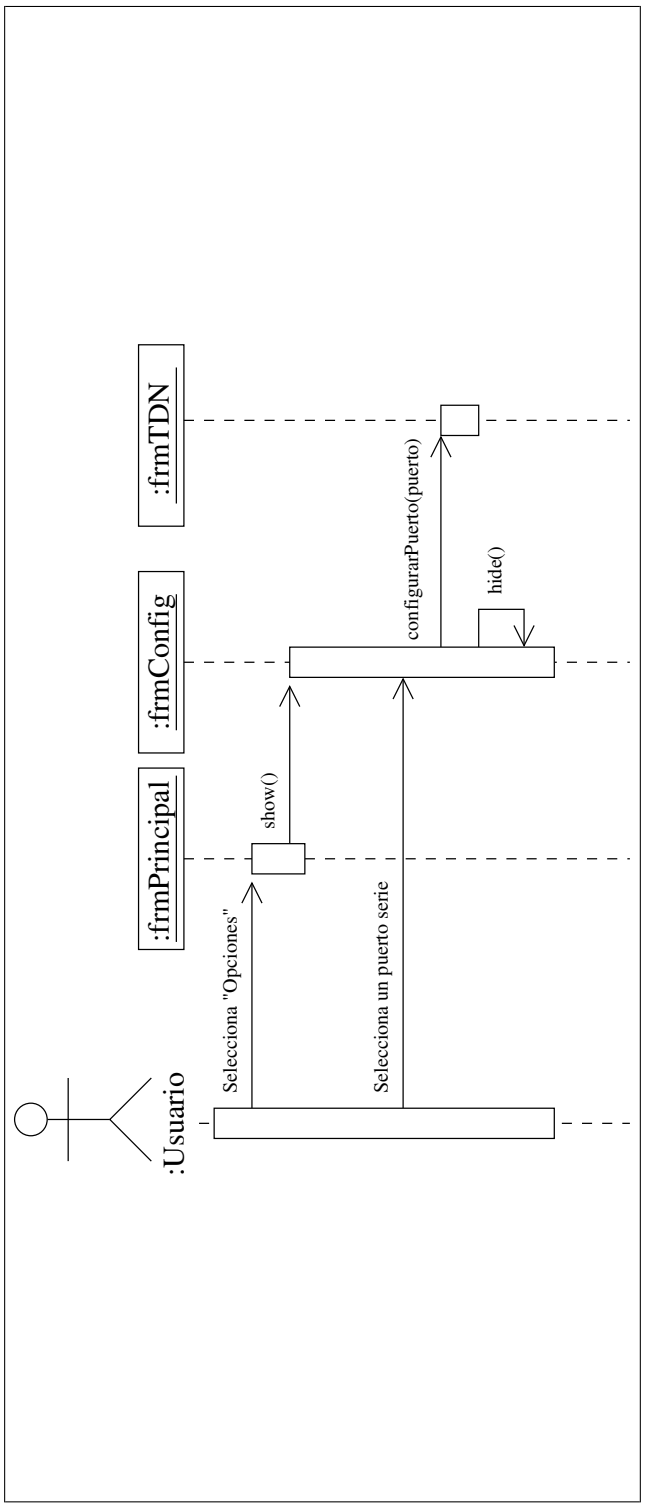


Figura 4.13: Diagrama de secuencia del caso de uso 8.

9. Cerrar gráfica.

9.1. Breve descripción.

Es el caso en el que el usuario decide cerrar una ventana que contiene una gráfica.

9.2. Flujo de eventos

9.2.1. El sistema muestra un cuadro de mensaje indicando al usuario si desea guardar el análisis actual.

9.2.2. Si el usuario decide guardar el análisis, comienza el caso de uso 5..

9.2.3. Si el usuario selecciona el botón “Cancelar” en el cuadro de mensaje desplegado, el caso de uso termina.

9.2.4. El sistema decrementa el número de ventanas desplegadas en pantalla. Si el número de ventanas desplegadas es cero, comienza el flujo alterno 9.3.1

9.2.5. El sistema descarga de la memoria la gráfica actual..

9.2.6. Fin del caso de uso.

9.3. Flujos alternos

9.3.1. Cero ventanas desplegadas.

a) El sistema oculta los menús: “Edición” y “Guardar”.

b) El sistema desactiva las opciones: “Guardar...”, “Exportar...”.

c) El flujo continúa en el paso 9.2.5.

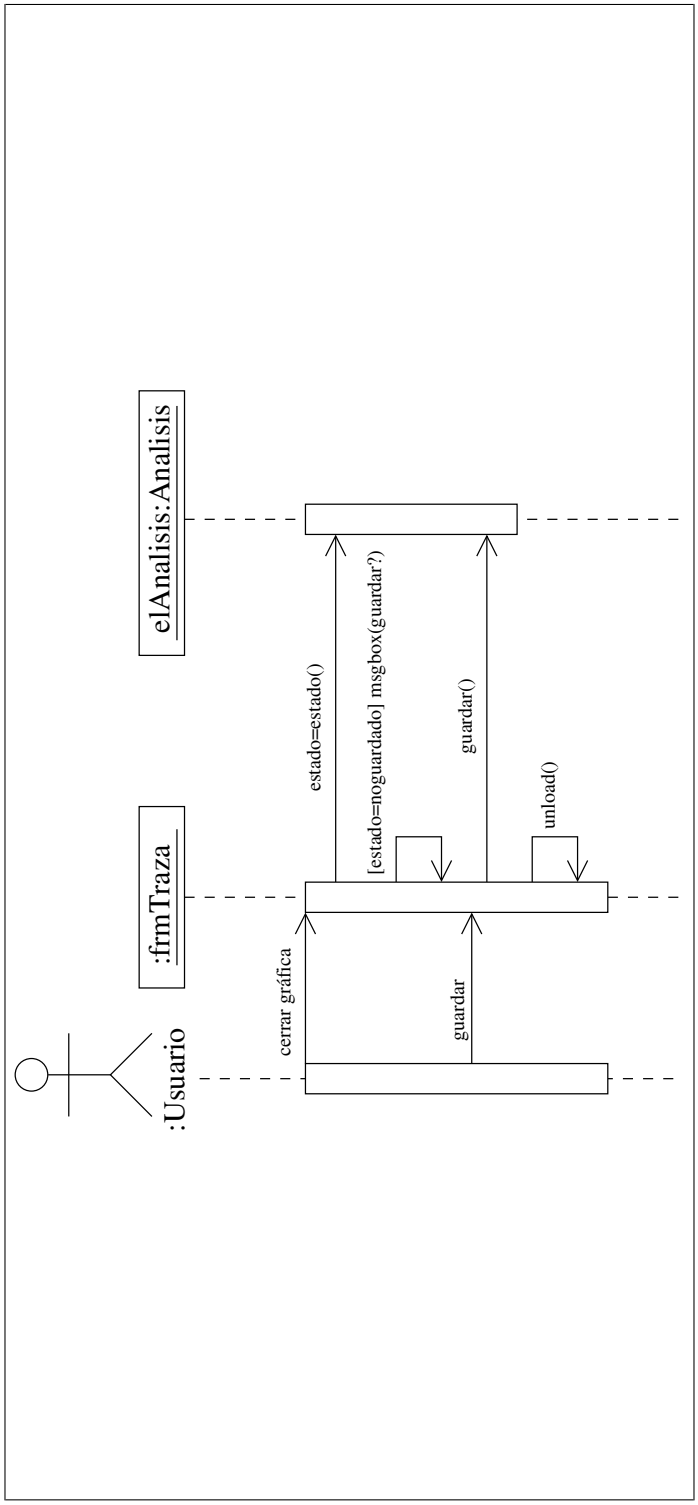


Figura 4.14: Diagrama de secuencia del caso de uso 9.

10. Información sobre la aplicación.

10.1. Breve descripción.

Es el caso de uso en el que el usuario solicita al sistema la información sobre la versión del mismo y los autores del mismo.

10.2. Flujo de eventos

10.2.1. El caso comienza cuando el usuario selecciona la opción “Acerca de ...” en el menú “Ayuda”.

10.2.2. El sistema solicita al TDN le envíe la versión del mismo. Si la comunicación con el TDN falla comienza el flujo alterno 10.3.1.

10.2.3. El sistema despliega la forma “Acerca de Tradiny” y muestra en la misma la versión de software y hardware actuales.

10.2.4. Fin del caso de uso.

10.3. Flujos alternos

10.3.1. TDN desconectado.

a) Se despliega un cuadro de mensaje indicando la falla en la comunicación y una sugerencia sobre su solución.

b) Se establece que la versión del TDN es “TDN no conectado”

c) El flujo continúa en el paso 10.2.3.

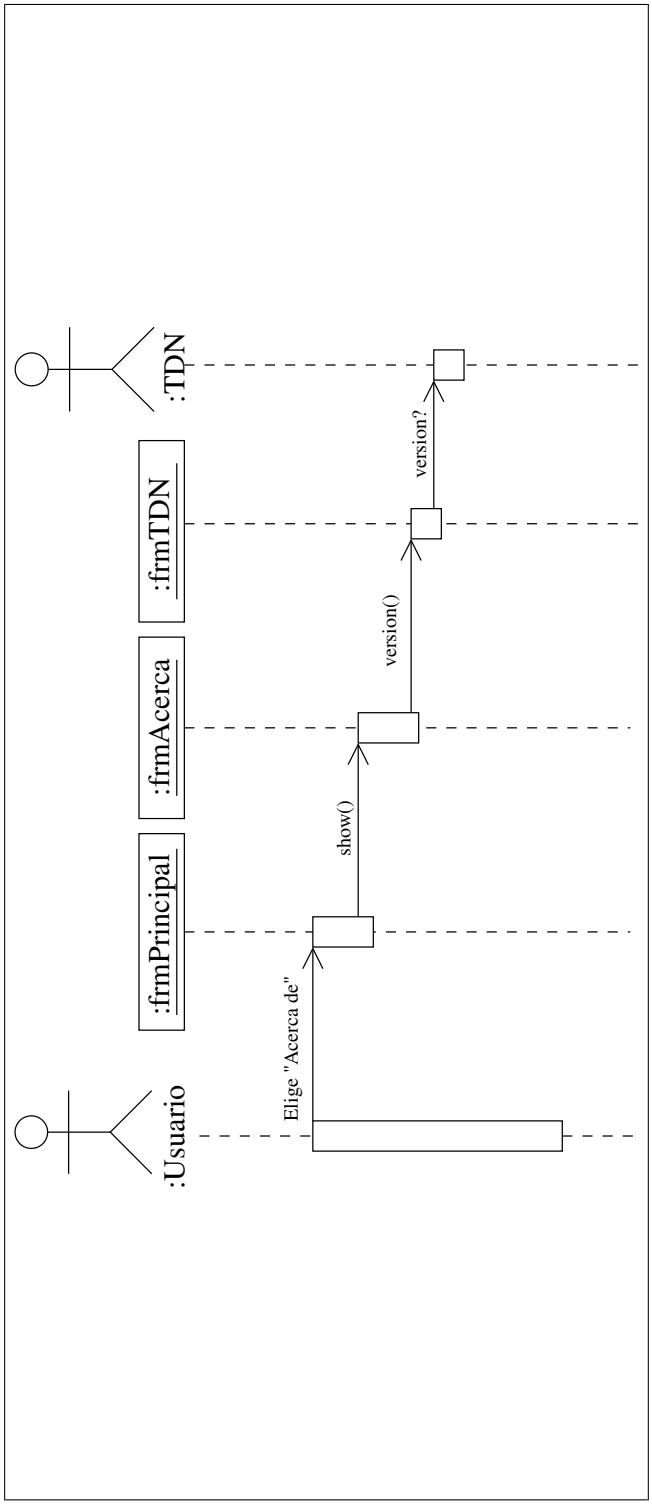


Figura 4.15: Diagrama de secuencia del caso de uso 10.

11. Copiar la gráfica en el portapapeles.

11.1. Breve descripción.

Es el caso de uso en el que el usuario desea copiar al portapapeles del sistema operativo, la gráfica desplegada en la pantalla.

11.2. Flujo de eventos

11.2.1. El caso de uso comienza cuando el usuario selecciona la opción “Copiar” del menú “Edición”.

11.2.2. El sistema borra el contenido del portapapeles.

11.2.3. El sistema copia en el portapapeles la gráfica actual.

11.2.4. Fin del caso de uso.

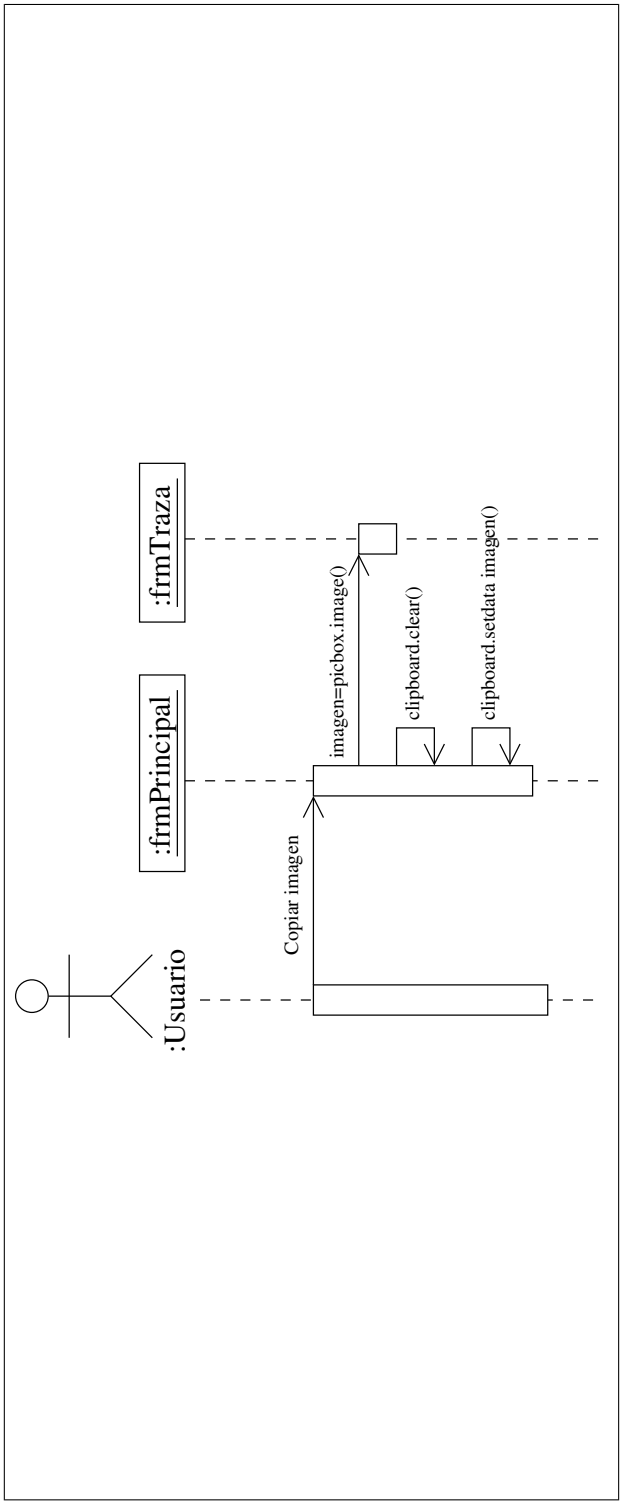


Figura 4.16: Diagrama de secuencia del caso de uso 11.

12. Exportar imagen.

12.1. Breve descripción.

Es el caso en el que el usuario desea exportar la gráfica desplegada en pantalla al formato de mapa de bits.

12.2. Flujo de eventos

12.2.1. El caso de uso comienza cuando el usuario selecciona la opción “Exportar gráfica...” del menú “Análisis”.

12.2.2. Se despliega el cuadro de dialogo común “Exportar gráfica actual como...”, mostrando el directorio desde donde se inició la aplicación o mostrando el directorio del último lugar donde se abrió o guardó un análisis.

12.2.3. El usuario escoge el directorio y el nombre del archivo con el que desea exportar la gráfica, y presiona el botón “Guardar”. Si el usuario presiona el botón “Cancelar” se inicia el flujo alterno 12.3.1.

12.2.4. El archivo escogido por el usuario es creado y la imagen de la gráfica actual se guarda en él usando el formato de mapa de bits. Si el archivo no pudo ser creado comienza el flujo alterno 12.3.2.

12.2.5. Fin del caso de uso.

12.3. Flujos alternos

12.3.1. Cancelar caso de uso.

a) Se oculta el cuadro de dialogo común “Guardar análisis actual como...”.

b) Se establece que el análisis no se guardó.

c) Termina el caso de uso.

12.3.2. Error al guardar archivo.

a) Se oculta el cuadro de dialogo común “Guardar análisis actual como...”.

b) Se despliega un cuadro de mensaje indicando el tipo de error encontrado.

c) Se establece que el análisis no se guardó.

d) Termina el caso de uso.

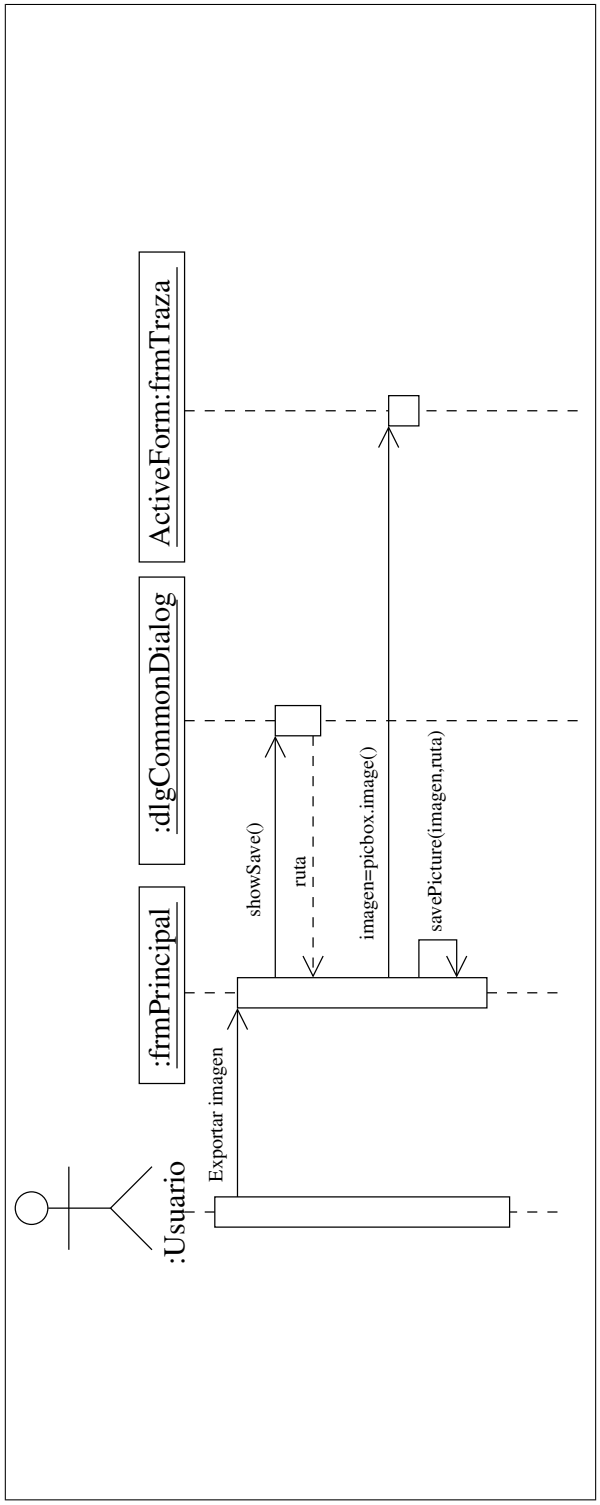


Figura 4.17: Diagrama de secuencia del caso de uso 12.

Nota: A menos que se especifique lo contrario todos los pasos mencionados en las secciones: flujo de eventos; flujos alternos; precondiciones y poscondiciones de los casos de uso listados anteriormente; son ejecutados por la aplicación. Existen además de los casos de uso documentados, otras interacciones que califican como casos de uso: alinear ventanas en forma de cascada; horizontal y verticalmente; ocultar la barra de herramientas, etc. Estas interacciones no fueron programadas explícitamente por nosotros, sino que el asistente para la creación de proyectos de *Visual Basic* las crea por defecto para las aplicaciones basadas en el esquema *Multiple Document Interface*. Cuando el usuario manipula el tamaño de la gráfica desplagada en pantalla y cuando cambia la escala de la misma, interactúa con el sistema; sin embargo, estas interacciones son parte de los requerimientos del sistema y por lo tanto no tienen un caso de uso asociado.

Secuencia de inicio y terminación de la aplicación

Las operaciones que realiza el sistema durante su inicio y terminación se detallan a continuación.

Inicio de la aplicación

1. Flujo de eventos

1.1. El sistema establece que la configuración no ha cambiado.

1.2. El sistema establece que el directorio de trabajo es el directorio donde reside la aplicación.

1.3. El sistema busca en el directorio donde reside la aplicación, los archivos: “calibrar.s19”, “frec01.s19” y “frec02.s19”. Si alguno de estos archivos no se encuentra comienza el flujo alterno 2.1..

1.4. El sistema busca en el directorio de trabajo el archivo “ajustes.ini”. Si no lo encuentra comienza el flujo alterno 2.2..

1.5. El sistema abre el archivo “ajustes.ini” y lee del mismo los ajustes del trazador.

1.6. El sistema carga la forma frmTrazador.

1.7. El sistema abre el puerto serie indicado en el resgitro del sistema operativo y lo configura a 9600 bps, 8N1.

1.8. El sistema recupera del registro del sistema operativo, las últimas coordenadas de la forma frmPrincipal.

1.9. El sistema recupera del registro del sistema operativo, el nombre del último directorio en el que el usuario abrió o guardó un análisis.

1.10. El sistema oculta los menús: “Edición” y “Guardar”.

1.11. El sistema desactiva las opciones: “Guardar...”, “Exportar...”.

1.12. Se establece que el número de ventanas hijas actualmente en pantalla es cero.

1.13. Se establece que el número de análisis realizados es cero.

1.14. El sistema es totalmente funcional y en este punto pueden comenzar cualquiera de los siguientes casos de uso: 1., 7., 4., 8. y 4.3.2.

2. Flujos alternos

2.1. Archivos faltantes.

2.1.1. El sistema muestra un cuadro de información indicando el nombre del archivo faltante.

2.1.2. El sistema aborta su ejecución.

2.1.3. Termina el caso de uso.

2.2. Archivo de configuración faltante, ejecución por primera vez.

2.2.1. Inicia el caso de uso 8..

2.2.2. Inicia el caso de uso 7..

2.2.3. El sistema guarda la configuración obtenida en los casos de uso mencionados.

2.2.4. El flujo continúa en el paso 1.6..

Término de la aplicación

1. Flujo de eventos

1.1. Comienza cuando el usuario selecciona la opción “Salir” del menú “Análisis” o cuando hace clic sobre el botón “X”.

1.2. El sistema guarda en el registro del sistema operativo las coordenadas de la forma frmPrincipal.

1.3. El sistema guarda en el registro del sistema operativo el nombre del último directorio donde el usuario abrió o guardó un análisis.

1.4. Si la configuración cambió, actualiza la información del archivo “ajustes.ini”.

1.5. El sistema descarga cada una de las gráficas desplegadas en pantalla y finalmente la forma frmPrincipal.

1.6. El sistema termina su ejecución.

4.3.3. Modelo conceptual

El modelo conceptual es una representación de conceptos en un dominio del problema. En UML se representa como un grupo de diagramas de estructura estática donde no se define ninguna operación. La designación de modelo conceptual ofrece la ventaja de subrayar fuertemente una concentración en los conceptos del dominio, no en las entidades del software. La creación del modelo conceptual es el artefacto más importante a crear durante el análisis orientado a objetos.

En la figura 4.18 se ilustra el modelo conceptual que obtuvimos después de analizar el problema.

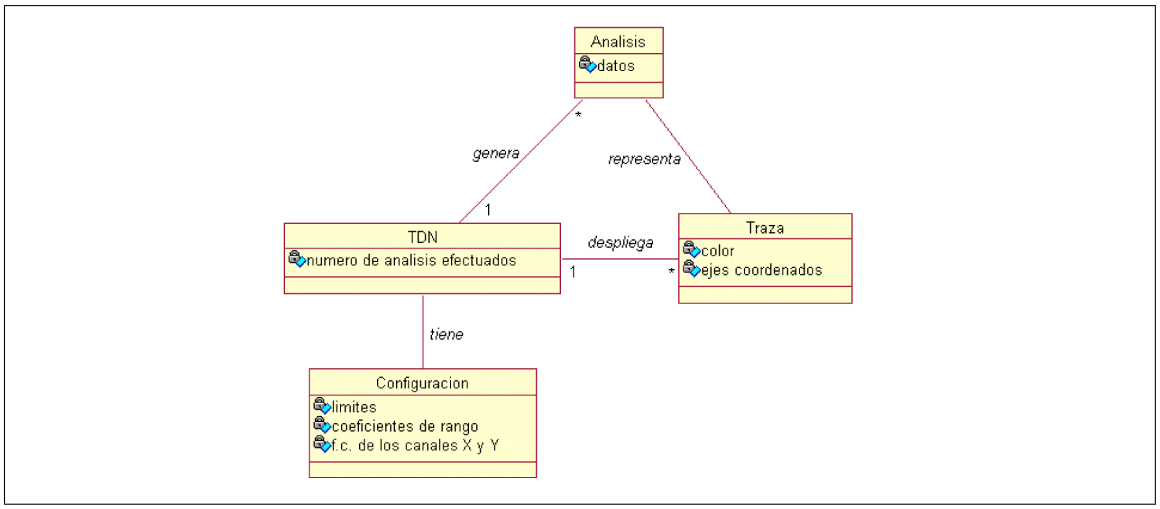


Figura 4.18: Diagrama de clases (conceptual).

4.3.4. Diagrama de clases

El diagrama de clases describe los tipos de objetos que hay en el sistema y las diversas clases de relaciones estáticas que existen entre ellos. También muestran los atributos y operaciones de una clase y las restricciones a que se ven sujetos, según la forma en que se conecten los objetos. Muestra las clases de la implementación del sistema.

La figura 4.19 corresponde al diagrama de clases. Se puede ver que el mismo es más complejo que el modelo conceptual que las asociaciones son diferentes. Para facilitar la comprensión del mismo no se incluyeron las propiedades ni los métodos de las clases.

En el apéndice B, se encuentra la documentación de las clases mostradas en la figura 4.19, junto con el diagrama completo de clases (incluyendo métodos y propiedades).

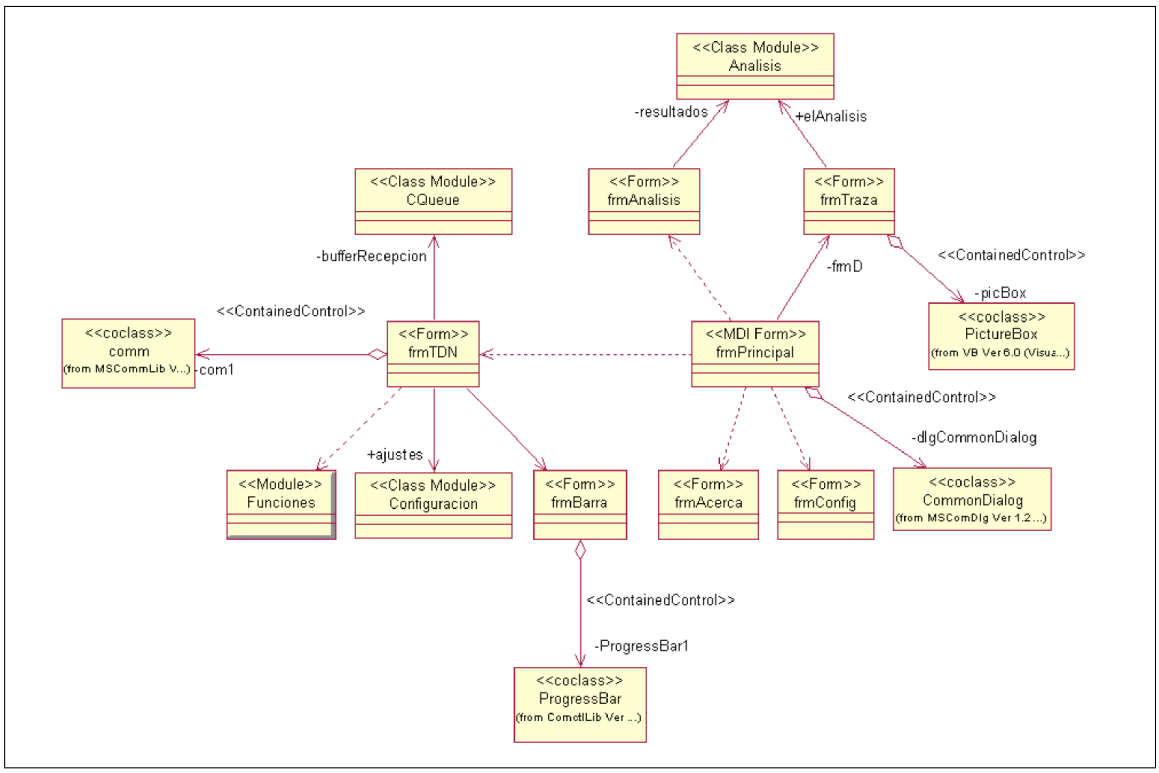


Figura 4.19: Diagrama de clases (implementación).

4.3.5. Diagrama de distribución

El diagrama de distribución muestra la implementación física de los componentes de software en componentes de hardware, así como el tipo de conexión entre los mismos. En la figura 4.20, se muestra el diagrama distribución para el sistema en su conjunto.

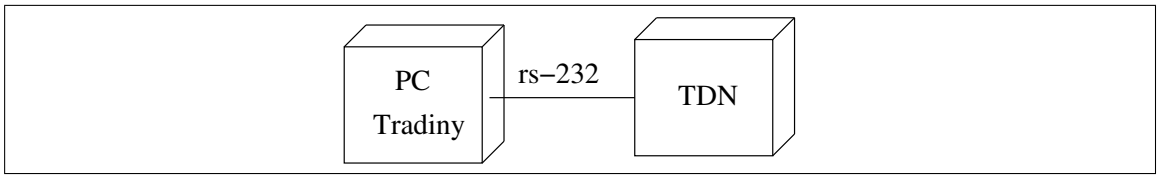


Figura 4.20: Diagrama de distribución.

4.3.6. Glosario

Ajustes Son los datos obtenidos durante la calibración del TDN: límites y coeficientes de rango; factores de corrección de los canales X y Y.

Análisis Los datos obtenidos por el TDN al analizar un sistema eléctrico.

Canales X y Y Para obtener la representación gráfica de un análisis es necesaria la representación de cada punto del mismo, mediante coordenadas cartesianas (x,y). Los canales X y Y del TDN proporcionan las coordenadas de los puntos.

Gráfica actual La ventana desplegada en pantalla que contiene la representación gráfica del análisis efectuado o recuperado.

Límites Véase la descripción de rango.

Rango El TDN genera señales senoidales desde 10 hasta 100 mil hertz; sin embargo, las genera por etapas. Son cuatro etapas diferentes y cada una abarca una década de frecuencias: la primera abarca desde 10 hasta 100 hertz; la segunda desde 100 hasta 1000 hertz; etc. Cada una de estas etapas es un rango y los límites son la frecuencia inicial y final del mismo.

Sesión El tiempo durante el cual el usuario hace uso del sistema.

Sistema Con este término nos referimos a la aplicación en sí, al software Tradiny.

Sistema eléctrico Un circuito eléctrico conectado externamente al TDN.

TDN Trazador Digital de Nyquist. El nombre del hardware elaborado para esta tesis.

Tradiny El nombre del software, interfaz gráfica de usuario, elaborado para el TDN.

Usuario La persona que hace uso del sistema.

$dac Dato binario (byte) que representa el dato a colocar en el convertidor D/A del TDN.

$datos Información binaria que tiene algún significado para el programa.

$relay Dato binario (byte) que representa el número de relevador que debe activar el TDN.

Capítulo 5

Ejemplos de aplicación

5.1. Introducción

En el presente capítulo se hablará de las pruebas del funcionamiento del TDN. Se presentarán algunos análisis de respuesta en frecuencia para sistemas cuya función de transferencia ya es conocida, esto con la finalidad de comparar el análisis efectuado a partir del Trazador Digital con un software de análisis profesional como puede ser *Matlab* en su versión 5.3. A partir de esta comparación se podrá obtener una idea general de los alcances del TDN, teniendo en cuenta que es una aplicación enfocada al auxilio didáctico.

5.2. Análisis de un circuito RC

La figura (5.1) muestra un sistema conformado por dos circuitos RC en cascada. Considerando que la señal de entrada del sistema es v_E y la señal de salida es v_S, calcularemos la función de transferencia del sistema.

Las ecuaciones para este sistema son las siguientes.

$$\frac{1}{C_1}\int (i_1 - i_2)dt + R_1 i_1 = v_E \tag{5.1}$$

$$\frac{1}{C_1}\int (i_2 - i_1)dt + R_2 i_2 + \frac{1}{C_2}\int i_2 dt = 0 \tag{5.2}$$

$$\frac{1}{C_2}\int i_2 = v_S \tag{5.3}$$

Si aplicando transformada de Laplace y suponiendo condiciones iniciales nulas obtenemos:

$$\frac{1}{C_1 s}[I_1(s) - I_2(s)] + R_1 I_1(s) = V_E(s) \tag{5.4}$$

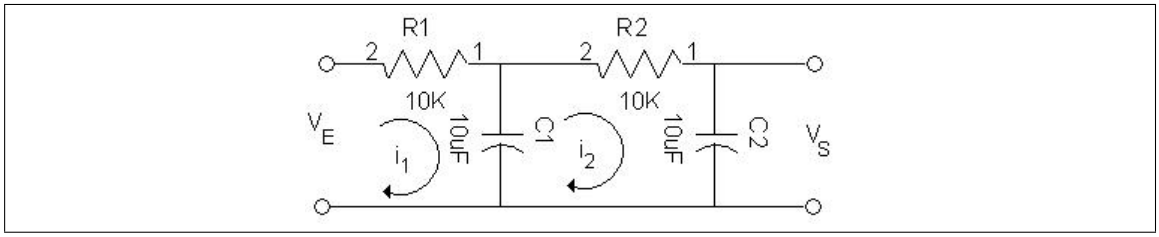


Figura 5.1: Circuito RC

$$\frac{1}{C_1 s}[I_2(s) - I_1(s)] + R_2 I_2(s) + \frac{1}{C_2 s} I_2(s) = 0 \tag{5.5}$$

$$\frac{1}{C_2} I_2(s) = V_S(s) \tag{5.6}$$

Eliminando $I_1(s)$ de las ecuaciones (5.4) y (5.5) y escribiendo $V_E(s)$ en términos de $I_2(s)$ encontramos que la función de transferencia entre $V_E(s)$ y $V_S(s)$ es:

$$\begin{aligned} \frac{V_S(s)}{V_E(s)} &= \frac{1}{(R_1 C_1 s)(R_2 C_2 s + 1) + R_1 C_2 s} \\ &= \frac{1}{R_1 C_1 R_2 C_2 s^2 + (R_1 C_1 + R_2 C_2 + R_1 C_2) s + 1} \end{aligned} \tag{5.7}$$

Considerando que, de la figura (5.1) los valores de los capacitores y resistencias son $R_1 = R_2 = 10K\Omega$ y $C_1 = C_2 = 0{,}1\mu F$, entonces nuestro sistema de prueba tiene como función de transferencia:

$$G(s) = \frac{1}{0{,}000001 s^2 + 0{,}003 s + 1} \tag{5.8}$$

Al obtener la respuesta en frecuencia de este sistema (haciendo $s = j\omega$), observamos que cuando la frecuencia de la señal de entrada es 0, el primer punto de la traza de Nyquist es el $|G(j\omega)| \angle G(j\omega) = 1 \angle 0^o$.

$$\begin{aligned} G(j\omega) &= \frac{1}{-0{,}000001\omega^2 + 0{,}003 j\omega + 1} \\ G(j\omega)|_{\omega=0} &= 1 \\ |G(j\omega)| \angle G(j\omega)|_{\omega=0} &= 1 \angle 0^o \end{aligned} \tag{5.9}$$

Otro punto interesante de la traza de Nyquist se obtiene cuando $\omega = 1000$:

$$\begin{aligned} G(j\omega)|_{\omega=1000} &= -\frac{1}{3} j \\ |G(j\omega)| \angle G(j\omega)|_{\omega=1000} &= -\frac{1}{3} \angle 270^o \end{aligned} \tag{5.10}$$

La traza de Nyquist de la función de transferencia (5.8) se muestra en la figura (5.2), y se obtuvo mediante el empleo del programa *Matlab* para el intervalo de frecuencias $10 < f < 100,000$ [Hz], donde f es la frecuencia de la señal senoidal de entrada al sistema de prueba. Es importante mencionar que con *Matlab* se obtiene una traza de Nyquist completa, ya que también efectúa el análisis para frecuencias negativas. En la figura (5.2), pueden observarse los puntos (5.9) y (5.10). La figura (5.3) muestra el resultado de un análisis hecho con el TDN. Como puede observarse, las características de la respuesta en frecuencia del circuito RC se pueden analizar a partir de la gráfica obtenida con el TDN. Se observa que $|G(j\omega)|$ no es constante a medida que la frecuencia cambia, y que la frecuencia de corte [1] es de 1000[Hz].

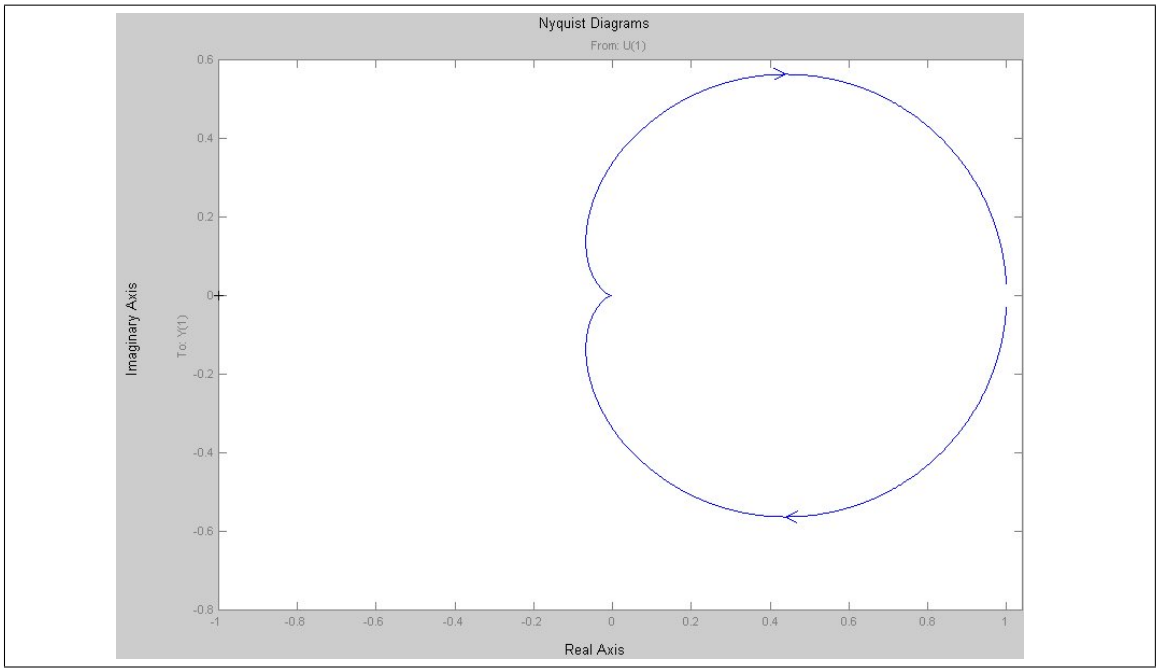


Figura 5.2: Traza de Nyquist obtenida mediante el programa Matlab para el circuito RC en cascada

5.3. Análisis de un filtro activo pasa-todo

La figura (5.4) muestra el circuito de un filtro pasa-todo diseñado para trabajar a una frecuencia de operación (f_o) de 1000 [Hz] [2]. El circuito RC que modifica la frecuencia

[1] El circuito de prueba es un filtro pasivo paso-bajas de segundo orden con una cierta frecuencia de corte

[2] El filtro pasa-todo es un filtro que permite el paso de señales de cualquier frecuencia. Tiene como finalidad proporcionar un cambio de fase a la señal de entrada del filtro. Cuando se ajusta a una cierta frecuencia de operación, debe existir un desfasamiento de 90° entre la señal de entrada y la señal de salida

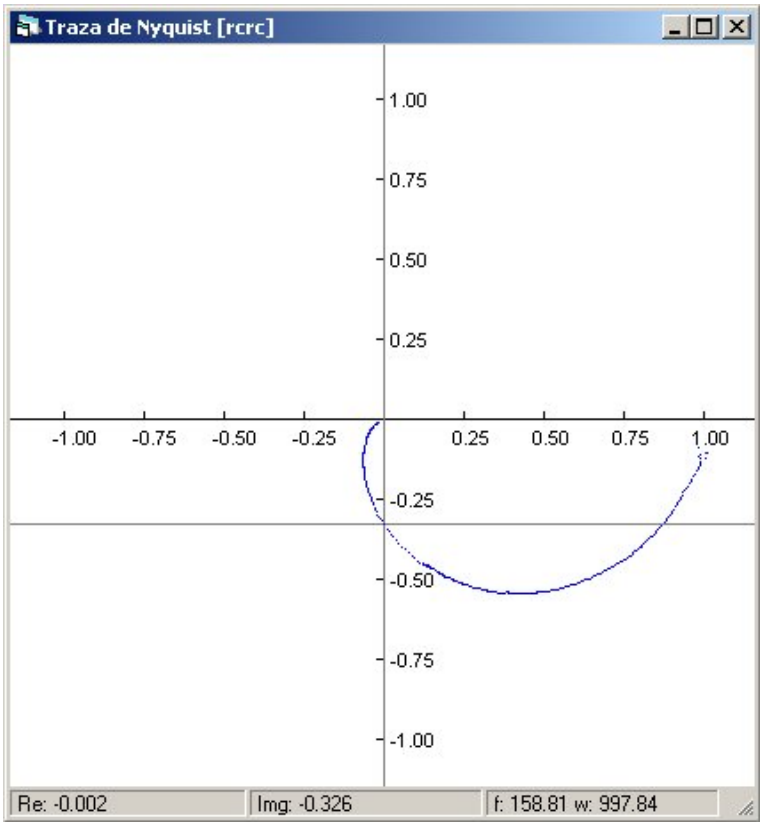


Figura 5.3: Traza de Nyquist obtenida mediante el TDN para un circuito RC en cascada

está conformado por la resistencia R y el capacitor C de la figura (5.4). Considerando que el capacitor $C = 0{,}1\mu F$, que se desea una $f_o = 1000$[Hz], y que:

$$f_o = \frac{1}{2\pi RC} \tag{5.11}$$

Entonces

$$\begin{aligned} R &= \frac{1}{2\pi f_o C} \\ &= \frac{1}{2\pi(1000[Hz])(0{,}1\text{x}10^{-6}[F])} \\ &= 1591{,}55\Omega \end{aligned} \tag{5.12}$$

Para obtener la función de transferencia de este sistema de prueba, se aplica el principio de superposición(el sistema es lineal), tal y como se muestra en la figura (5.5). Realizando

cuando la señal de entrada tiene la misma frecuencia que la de operación

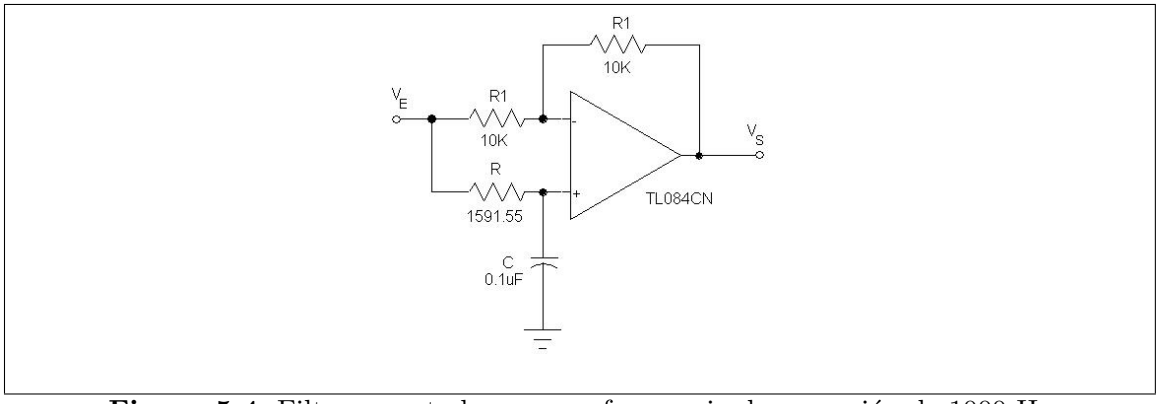


Figura 5.4: Filtro pasa-todo con una frecuencia de operación de 1000 Hz

este análisis, obtenemos las siguientes ecuaciones:

$$V_S = V_{S1} + V_{S2} \tag{5.13}$$

$$V_{S1} = -V_E \tag{5.14}$$

$$V_{S2} = \frac{2}{RCs+1} V_E \tag{5.15}$$

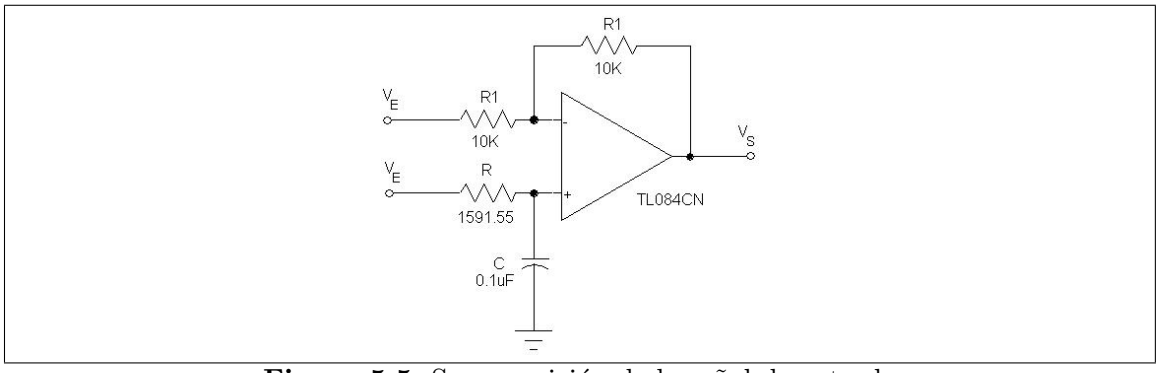


Figura 5.5: Superposición de la señal de entrada

Aplicando el principio de superposición y sustituyendo las ecuaciones (5.14) y (5.15) en (5.13)

$$\begin{aligned} V_S &= \frac{2}{RCs+1} V_E - V_E \\ &= \left(\frac{2}{RCs+1} - 1 \right) V_E \\ &= \left(\frac{2-(RCs+1)}{RCs+1} \right) V_E \end{aligned}$$

$$\begin{aligned} H(s) &= \frac{V_S(s)}{V_E(s)} \\ H(s) &= \frac{1 - RCs}{1 + RCs} \end{aligned} \tag{5.16}$$

Donde $H(s)$ es la función de transferencia del sistema de prueba, a partir de la cual es posible obtener su respuesta en frecuencia.

$$\begin{aligned} H(s)|_{s=j\omega} &= \frac{1 - j\omega RC}{1 + j\omega RC} \\ |H(j\omega)| &= \frac{|1 - j\omega RC|}{|1 + j\omega RC|} \\ &= \frac{1 - \omega^2 R^2 C^2}{1 - \omega^2 R^2 C^2} \\ &= 1 \ \forall \omega \end{aligned} \tag{5.17}$$

$$\begin{aligned} \angle|H(j\omega)| &= \arctan(-\omega RC) - \arctan(\omega RC) \\ &= -2\arctan(\omega RC) \end{aligned} \tag{5.18}$$

La figura (5.6) muestra la Traza de Nyquist obtenida a partir de la función de transferencia utilizando el programa *Matlab*, mientras que la figura (5.7) nos muestra la Traza de Nyquist obtenida a partir del TDN. Como puede observarse, no existe diferencia significativa entre las dos trazas. La gráfica (5.7), nos permite observar las características más importantes de este filtro:

- La magnitud es constante para toda ω
- Cuando la frecuencia de la señal de entrada es 1[KHz], el ángulo de desfasamiento es -90º.

5.4. Análisis de un filtro activo pasa-bajas de cuarto orden

La figura (5.8) muestra el circuito que corresponde a un filtro activo de 4to orden diseñado a partir del programa [8] cuya plantilla de diseño aparece en la figura (5.9), y que tiene las siguientes características:

- Frecuencia de corte $f_c = 100[Hz]$
- Frecuencia de supresión $f_s = 200[Hz]$
- Perdida de 60[dB] entre f_c y f_s.
- Ganancia unitaria $K = 1$

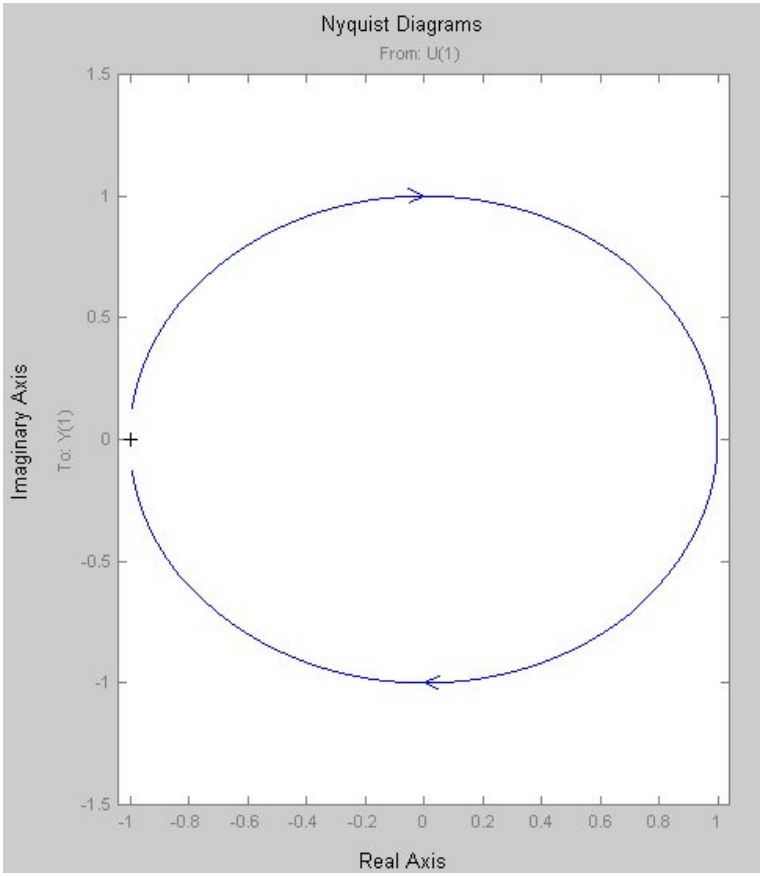


Figura 5.6: Traza de Nyquist de un filtro pasa-todo obtenida a partir de su función de transferencia

La figura (5.10), muestra un análisis efectuado a partir del TDN, para este filtro. En esta figura podemos observar las características del filtro. Observamos que cuando la frecuencia de análisis es igual a la frecuencia de corte $f_c = 100[Hz]$ la magnitud de la señal empieza a disminuir hasta convertirse en 0 cero poco después de que la frecuencia de análisis rebasa los 200 [Hz].

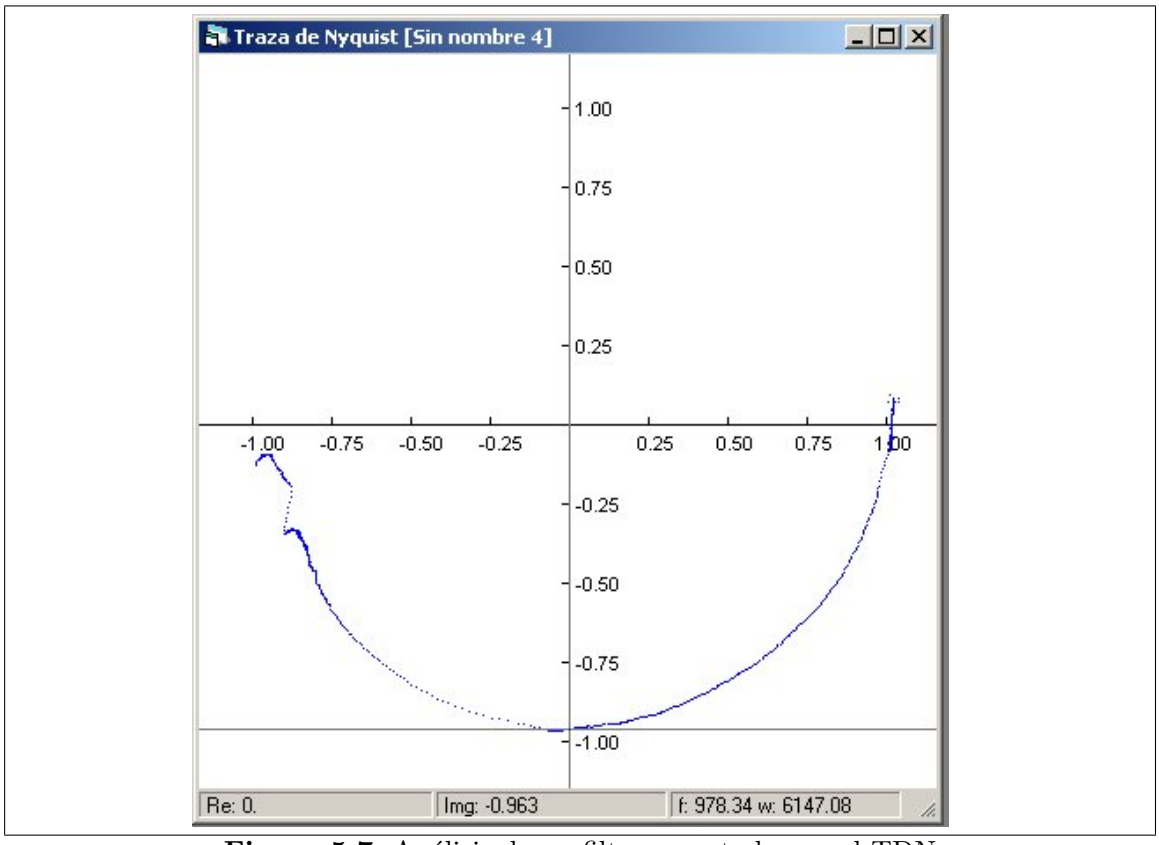


Figura 5.7: Análisis de un filtro pasa-todo con el TDN

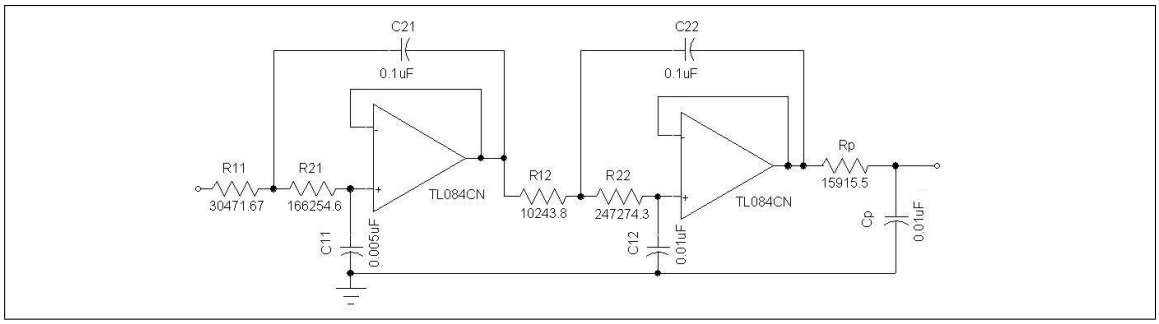


Figura 5.8: FIltro activo paso-bajas de 4to orden

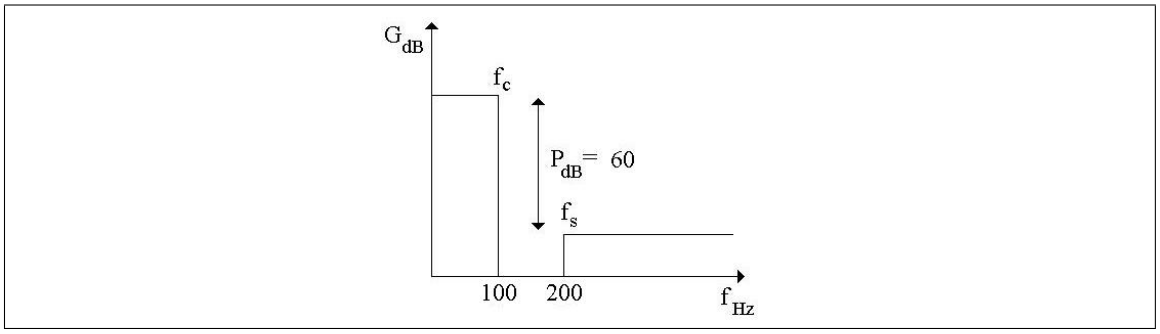


Figura 5.9: Plantilla de diseño de un filtro paso-bajas de 4to orden con frecuencia de corte en 100 [Hz]

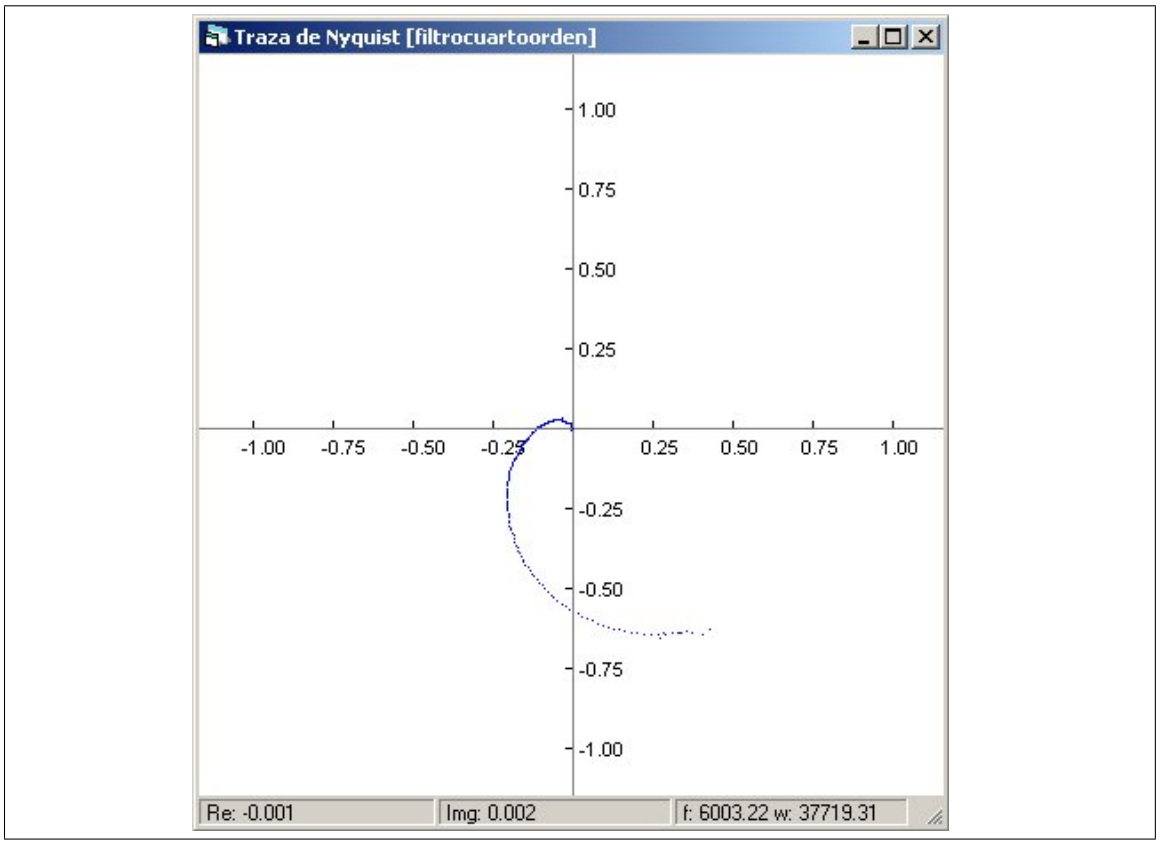


Figura 5.10: Análisis de un filtro paso-bajas de cuarto orden mediante el TDN

Conclusiones

Como se pudo observar en los ejemplos de aplicación los análisis efectuados con el TDN son confiables porque se apegan a los modelos matemáticos de los sistemas analizados.

Para desarrollar un circuito, instrumento, o sistema de control es necesario obtener su función de transferencia. Obtener este modelo matemático puede resultar una tarea muy laboriosa, por lo que se debe recurrir a análisis experimentales. Un método eficaz de análisis es la respuesta en frecuencia y uno de sus métodos gráficos es la Traza de Nyquist. La importancia de este método es que la función de transferencia de un sistema o de cualquiera de sus componentes, puede ser determinada experimentalmente por simples mediciones de respuesta en frecuencia.

Una vez que se ha medido la relación entre la magnitud y el ángulo de fase de un sistema en un número suficiente de frecuencias dentro de un rango de frecuencias de interés, es posible realizar la representación de la respuesta en frecuencia mediante la Traza de Nyquist y partir de está, de otras maneras (Bode, Nichols). Los alumnos que utilicen este sistema, podrán efectuar análisis de sistemas relativamente complejos, además de que podrán diseñar sistemas con características especificas, concentrándose exclusivamente en el desarrollo de su sistema, evaluándolo de manera interactiva al observar la respuesta (tanto gráficamente, como numéricamente) que entrega el TDN después de cada modificación realizada sobre su sistema. También podrán obtener conclusiones de sistemas ya construidos con respecto a su estabilidad, linealidad y calidad de desempeño en algún rango de frecuencias de interés.

El proceso de diseño o análisis de un sistema mediante el TDN, podrá repetirse tantas veces como se requiera. El TDN pretende servir a sus usuarios como una herramienta didáctica, que ayude a ejemplificar el concepto de análisis de respuesta en frecuencia, además de servir como una herramienta que ayude a disminuir el tiempo que un diseñador emplea para construir sus sistemas.

Apéndice A

Listado de programas

A continuación se presentan el listado de los programas mencionados en el capitulo software de control.

A.1. Programa enlace.asm (talker)

```
$include  "lib\equs.asm"

          org       $fe00
          ldx       #$1000
          lds       #$03ff
          bset      option,x,$80; activar el cad
          ldaa      #$30; recepcion y transmision a 9600 bps
          staa      baud,x
          ldaa      #$2c
          staa      sccr2,x; habilitar interrupcion por recep
          cli
          bset      ddrd,x,$20

ciclo     com       ptod,x
          ldy       #$ffff
c1        dey
          bne       c1
          bra       ciclo

********* Rutina de interrupcion principal
enlace    ldx       #$1000
          brset     scsr,x,$20,en_recep; verificar recepcion *normal*
          rti
en_recep  sei
          bset      ddrd,x,$20
          bclr      ptod,x,$20
          ldaa      scdr,x
          cmpa      #$00
          beq       en_s19; recibir programa s19
          cmpa      #$01
          beq       en_ini; iniciar ejecucion desde una direccion
          cmpa      #$02
```

```
          beq       en_listo; enviar version
          cmpa      #$03
          beq       en_det; detener la ejecucion del programa actual
          cmpa      #$04
          beq       en_ver;
          bset      ptod,x,$20
          cli
          rti

********* recibir programa s19
en_s19    brclr     scsr,x,$20,en_s19; esperar byte (longitud de paquete)
          ldaa      scdr,x
          psha
en_c1     brclr     scsr,x,$20,en_c1; esperar byte  (p. alta dir inicio paq)
          ldaa      scdr,x
en_c2     brclr     scsr,x,$20,en_c2; esperar byte  (p. baja dir inicio paq)
          ldab      scdr,x
          xgdy
          pula
en_c3     brclr     scsr,x,$20,en_c3; esperar byte  (bytes del paquete)
          ldab      scdr,x
          stab      $00,y
          iny
          deca
          bne       en_c3
          bset      ptod,x,$20
          jmp       $fe00


********* Iniciar ejecucion a partir de una direccion
en_ini    brclr     scsr,x,$20,en_ini; esperar byte (p. alta dir inicio)
          ldaa      scdr,x
en_c4     brclr     scsr,x,$20,en_c4; esperar byte (p. baja dir inicio)
          ldab      scdr,x
          xgdy
          lds       #$03ff; reiniciar pila
          bset      ptod,x,$20
          cli
          jmp       $00,y; ejecutar desde direccion

********* Enviar el paquete *listo*
en_listo  brclr     scsr,x,$60,en_listo; esperar si se esta transmitiendo o recibiendo
          ldaa      #$01
          staa      scdr,x
          bset      ptod,x,$20
          cli
          rti


********* Enviar la version del talker
en_ver    brclr     scsr,x,$40,en_ver; enviar cabecera de datos *0a*
          ldaa      #$0a
          staa      scdr,x
en_c5     brclr     scsr,x,$40,en_c5; enviar num datos
          ldaa      #$01
          staa      scdr,x
en_c6     brclr     scsr,x,$40,en_c6; enviar version
          ldaa      $ffbf
```

```
          staa      scdr,x
          bset      ptod,x,$20
          cli
          rti

********* Detener la ejecucion saltando al inicio del talker
en_det    jmp       $fe00

********* Version del talker
          org       $ffbf
          db        $11

********* Configurar vectores de interrupcion
          org       $fffe
          dw        $fe00
          org       $ffd6
          dw        enlace
```

A.2. Programa calibrar.asm

```
***Puertos del microcontrolador y sus registros de configuracion
ptob      equ       $04
ptof      equ       $05
ptoc      equ       $06
ddrc      equ       $07
ptod      equ       $08
ddrd      equ       $09

***Registros del puerto serie del microcontrolador
baud      equ       $2b
sccr2     equ       $2d
scsr      equ       $2e
scdr      equ       $2f

***Registros del convertidor analogico digital
adctl     equ       $30
adr1      equ       $31
adr2      equ       $32
adr3      equ       $33
adr4      equ       $34
option    equ       $39

***Direccion de inicio del buffer dentro de la RAM.
buffer    equ       $0300

          org       $0000
          ldx       #$1000
          ldy       #buffer

*Seleccion del primer rango de frecuencia y el valor mas alto.
*Se toman 18 mediciones de $00 a $FF, cada medicion es de 3 bytes
*Se envian mediante la rutina envlect
*Lo anterior se hace para los 4 rangos de frecuencia.

          ldab      #$01
```

```
ciclo2    stab      ptof,x
          jsr       ret1s;      retardo para estabilizar
          ldaa      #$00
ciclo1    jsr       aats
          jsr       medicion
          jsr       leerad
          jsr       envlect
          adda      #$0f
          cmpa      #$ff
          beq       sig1
          bra       ciclo1
sig1      jsr       aats
          jsr       medicion
          jsr       leerad
          jsr       envlect
          aslb
          cmpb      #$10
          beq       sig2
          bra       ciclo2
sig2      jmp       $fe00

********* envlect: envia 6 bytes iniciando en #buffer
envlect   psha
          ldaa      #$06
          staa      envNum
          ldy       #buffer
          sty       envDir
          jsr       envDatos
          pula
          rts

* medicion
* ret1s

********* medicion: toma una medicion y la almacena en donde apunta Y
medicion  psha
          bset      ptof,x,$30
          bclr      ptof,x,$30
          bset      ptof,x,$40
          jsr       ret1s
          bclr      ptof,x,$40
          bset      ptof,x,$50
          bclr      ptof,x,$50

          ldaa      ptoc,x
          staa      $00,y
          iny

          bset      ptof,x,$10
          ldaa      ptoc,x
          staa      $00,y
          iny
          bclr      ptof,x,$10

          bset      ptof,x,$20
          ldaa      ptoc,x
          staa      $00,y
          iny
```

```
          pula
          rts

********* retardo de 1 segundo *exacto*
ret1s     pshy
          ldy       #$c350
r00       dey
          beq       retfin
          cpx       $00
          cpx       $00
          cpx       $00
          cpx       $00
          cpx       $00
          cpx       $00
          bra       r00
retfin    puly
          rts

********** leerad: lee los valores de adr2,adr3 y adr4
********** y los almacena en donde apunta Y
leerad    psha
          ldaa      #$10
          staa      adctl,x
leerad0   ldaa      adctl,x ; esperar hasta
          anda      #$80    ; terminar la
          beq       leerad0 ; conversion en el cad
finlad    ldaa      adr2,x
          staa      $00,y
          iny
          ldaa      adr3,x
          staa      $00,y
          iny
          ldaa      adr4,x
          staa      $00,y
          iny
          pula
          rts

********* envDatos(envDir,envNum)
********* envDir 2 bytes
********* envNum 1 byte
envDatos  psha
          pshb
          pshy

env00     brclr     scsr,x,$40,env00; enviar cabecera de datos *0a*
          ldaa      #$0a
          staa      scdr,x

env01     brclr     scsr,x,$40,env01; enviar tamano de paquete
          ldab      envNum
          stab      scdr,x
          ldy       envDir

env02     brclr     scsr,x,$40,env02; enviar datos
          ldaa      $00,y
          staa      scdr,x
          iny
```

```
          decb
          bne       env02

          puly
          pulb
          pula
          rts

envNum    db        $aa
envDir    dw        $aaaa
*********************************



*********************************
* Invertir acumulador a b7 a b0 *
*********************************
aats      psha
          pshb
          lsra; b0
          rolb
          lsra; b1
          rolb
          lsra; b2
          rolb
          lsra; b3
          rolb
          lsra; b4
          rolb
          lsra; b5
          rolb
          lsra; b6
          rolb
          lsra; b7
          rolb
          stab      ptob,x
          pulb
          pula
          rts
```

A.3. Programa frec01.asm

```
*Puertos del microcontrolador
ptob      equ       $04
ptof      equ       $05
ptoc      equ       $06
ddrc      equ       $07
ptod      equ       $08
ddrd      equ       $09

*Registros del puerto serie
baud      equ       $2b
sccr2     equ       $2d
scsr      equ       $2e
scdr      equ       $2f
```

```
*Registros del convertidor AD
adctl     equ       $30
adr1      equ       $31
adr2      equ       $32
adr3      equ       $33
adr4      equ       $34
option    equ       $39

*Direccion de inicio del buffer
buffer    equ       $0300

          org       $0000
          ldx       #$1000
*Seleccion de rango y frecuencia de operacion
          ldaa      #$f0;dac
          jsr       aats
          ldaa      #$01;relay
          staa      ptof,x

*Reset al circuito detector de maximos
          ldy       #buffer
          bset      ptof,x,$80
          jsr       ret1s
          bclr      ptof,x,$80
          jsr       medicion
          jsr       leerad

          ldaa      #$06
          staa      envNum
          ldd       #$0300
          std       envDir
          jsr       envDatos
          jmp       $fe00


********** leerad: lee los valores de adr2,adr3 y adr4
********** y los almacena en donde apunta Y
leerad    psha
          ldaa      #$10
          staa      adctl,x
leerad0   ldaa      adctl,x ; esperar hasta
          anda      #$80    ; terminar la
          beq       leerad0 ; conversion en el cad
finlad    ldaa      adr2,x
          staa      $00,y
          iny
          ldaa      adr3,x
          staa      $00,y
          iny
          ldaa      adr4,x
          staa      $00,y
          iny
          pula
          rts


* medicion
* ret1s
```

```
********* medicion: toma una medicion y la almacena en donde apunta Y
medicion  psha
          bset      ptof,x,$30
          bclr      ptof,x,$30
          bset      ptof,x,$40
          jsr       ret1s
          bclr      ptof,x,$40
          bset      ptof,x,$50
          bclr      ptof,x,$50

          ldaa      ptoc,x
          staa      $00,y
          iny

          bset      ptof,x,$10
          ldaa      ptoc,x
          staa      $00,y
          iny
          bclr      ptof,x,$10

          bset      ptof,x,$20
          ldaa      ptoc,x
          staa      $00,y
          iny
          pula
          rts

********* retardo de 1 segundo *exacto*
ret1s     pshy
          ldy       #$c350
r00       dey
          beq       retfin
          cpx       $00
          cpx       $00
          cpx       $00
          cpx       $00
          cpx       $00
          cpx       $00
          bra       r00
retfin    puly
          rts

********* envDatos(envDir,envNum)
********* envDir 2 bytes
********* envNum 1 byte
envDatos  psha
          pshb
          pshy
          bset      ddrd,x,$20
          bclr      ptod,x,$20

env00     brclr     scsr,x,$40,env00; enviar cabecera de datos *0a*
          ldaa      #$0a
          staa      scdr,x

env01     brclr     scsr,x,$40,env01; enviar tamano de paquete
          ldab      envNum
```

```
          stab      scdr,x
          ldy       envDir

env02     brclr     scsr,x,$40,env02; enviar datos
          ldaa      $00,y
          staa      scdr,x
          iny
          decb
          bne       env02

          bset      ptod,x,$20
          puly
          pulb
          pula
          rts

envNum    db        $aa
envDir    dw        $aaaa
*********************************

*********************************
* Invertir acumulador a b7 a b0 *
*********************************
aats      psha
          pshb
          lsra; b0
          rolb
          lsra; b1
          rolb
          lsra; b2
          rolb
          lsra; b3
          rolb
          lsra; b4
          rolb
          lsra; b5
          rolb
          lsra; b6
          rolb
          lsra; b7
          rolb
          stab      ptob,x
          pulb
          pula
          rts
```

A.4. Programa frec02.asm

```
ptob      equ       $04
ptof      equ       $05
ptoc      equ       $06
ddrc      equ       $07
ptod      equ       $08
ddrd      equ       $09

baud      equ       $2b
```

```
sccr2     equ       $2d
scsr      equ       $2e
scdr      equ       $2f

adctl     equ       $30
adr1      equ       $31
adr2      equ       $32
adr3      equ       $33
adr4      equ       $34

option    equ       $39

buffer    equ       $0300

          org       $0000
          ldx       #$1000

          bset      option,x,$80; activar el cad

          ldaa      #$03
          staa      envNum
          ldy       #buffer
          sty       envDir

          ldaa      #$04;  relevador
          staa      ptof,x

          ldaa      #$00;  dac
          staa      ptob,x
          jsr       ret1s

ciclo     jsr       aats
          bset      ptof,x,$80
          jsr       ret20
          bclr      ptof,x,$80
          jsr       ret20
          jsr       mediciones
          jsr       envdatos
          inca
          bne       ciclo
          jmp       $fe00

mediciones:
          pshx
          pshy
          psha

          ldy       #$0200
          ldaa      #$0a
me0       jsr       leerad
          deca
          bne       me0

          ldy       #$300
          ldx       #$200
me2       pshx
          pshy
          ldy       #$000a
```

```
          ldd       $00,x
me1       addd      $06,x
          inx
          inx
          inx
          inx
          inx
          inx
          dey
          bne       me1

          ldx       #$000a
          idiv
          xgdx
          puly
          stab      $00,y
          iny
          pulx
          inx
          inx
          cpy       #$0303
          bne       me2

          pula
          puly
          pulx
          rts


********* ret20: c=26+11N
ret20     pshy
          ldy       #$4703; 100 ms
ret20c    cpx       #$0000
          dey
          bne       ret20c
          puly
          rts

********** leerad: lee los valores de adr2,adr3 y adr4 y los almacena en donde Y
leerad    psha
          pshb
          ldaa      #$10
          staa      adctl,x
leerad0   ldaa      adctl,x ; esperar hasta
          anda      #$80    ; terminar la
          beq       leerad0 ; conversion en el cad

          ldaa      #$00
          ldab      adr2,x
          std       $00,y
          iny
          iny

          ldab      adr3,x
          std       $00,y
          iny
          iny
```

```
          ldab      adr4,x
          std       $00,y
          iny
          iny

          pulb
          pula
          rts

* ret1s

********* retardo de 1 segundo *exacto*
ret1s     pshy
          ldy       #$c350
r00       dey
          beq       retfin
          cpx       $00
          cpx       $00
          cpx       $00
          cpx       $00
          cpx       $00
          cpx       $00
          bra       r00
retfin    puly
          rts

********* envDatos(envDir,envNum)
********* envDir 2 bytes
********* envNum 1 byte
envDatos  psha
          pshb
          pshy
          bset      ddrd,x,$20
          bclr      ptod,x,$20

env00     brclr     scsr,x,$40,env00; enviar cabecera de datos *0a*
          ldaa      #$0a
          staa      scdr,x

env01     brclr     scsr,x,$40,env01; enviar tamano de paquete
          ldab      envNum
          stab      scdr,x
          ldy       envDir

env02     brclr     scsr,x,$40,env02; enviar datos
          ldaa      $00,y
          staa      scdr,x
          iny
          decb
          bne       env02

          bset      ptod,x,$20
          puly
          pulb
          pula
          rts
envNum    db        $aa
envDir    dw        $aaaa
```

```
*********************************


*********************************
* Invertir acumulador a b7 a b0 *
*********************************
aats      psha
          pshb
          lsra; b0
          rolb
          lsra; b1
          rolb
          lsra; b2
          rolb
          lsra; b3
          rolb
          lsra; b4
          rolb
          lsra; b5
          rolb
          lsra; b6
          rolb
          lsra; b7
          rolb
          stab      ptob,x
          pulb
          pula
          rts
```

Apéndice B

Documentación de clases

B.1. Forma frmAcerca

Métodos

cmdOK_Click () Ocurre cuando el usuario presiona el botón "OK". Cierra la ventana actual.

Form_Load () Pregunta por la versión del hardware al TDN e inicializa las etiquetas de la forma y

B.2. Forma frmAnalisis

Propiedades

resultados Tipo: Analisis. Los datos del análisis efectuado.

Métodos

btnCancelar_Click () Ocurre cuando el usuario presiona el botón "Cancelar". Cierra la ventana actual.

btnEjecutar_Click () Ocurre cuando el usuario presiona el botón "Ejecutar". Inicia el análisis solicitado por el usuario y genera la gráfica correspondiente.

chkPersonal_Click () Ocurre cuando el usuario activa el checkbox "Personal". Deshabilita las listas combo etiquetadas como "Desde" y "hasta" y habilita los cuadros de texto anexos.

Form_Load () Ocurre cuando la forma se despliega por primera vez. Inicializa las listas combo con los valores de frecuencias para los cuales se efectuarán los análisis.

opFrecPart_Click () Ocurre cuando el usuario activa la opción "Análisis para una frecuencia particular"

opRanFrec_Click () Ocurre cuando el usuario activa el checkbox "Análisis en un rango de frecuencias".

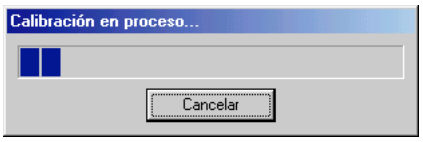
resultado () Regresa el valor de la variable Res.

Variables

mvarresultados Tipo: Analisis. Variable asociada con la propiedad resultados.

Res Tipo: Integer. Guarda el estado del resultado del análisis: 0 si el análisis solicitado tuvo éxito o un valor negativo si el análisis falló.

B.3. Forma frmBarra



Métodos

btnCancelar_Click () Ocurre cuando el usuario presiona el botón "Cancelar". Cancela el la operación en progreso.

B.4. Forma frmConfig

Métodos

btnAceptar_Click () Ocurre cuando el usuario presiona el botón "Aceptar". Valida el puerto escogido por el usuario en la lista desplegable "Puerto" y oculta la forma.

btnCancelar_Click () Ocurre cuando el usuario presiona el botón "Cancelar". Oculta la forma.

Form_Load () Ocurre cuando la forma se despliega por primera vez. Inicializa la lista desplegable "Puertos".

B.5. Forma frmPrincipal

Métodos

MDIForm_Load () Ocurre cuando la forma se despliega por primera vez. Inicializa las variables de la forma; busca los archivos necesarios para la aplicación; obtiene del registro del sistema operativo las ultimas coordenadas de la forma.

MDIForm_Unload (Cancel) Ocurre cuando la forma se descarga de la memoria. Guarda las coordenadas de la forma en el resgistro del sistema operativo.

mnuAbrir_Click () Ocurre cuando el usuario selecciona la opción "Abrir". Despliega una cuadro de dialogo común; abre el archivo seleccionado por el usuario y despliega la gráfica correspondiente.

mnuCalibrar_Click () Ocurre cuando el usuario selecciona la opción "Calibrar". Llama al metodo calibrar de la forma frmCalibrar.

mnuEditCopy_Click () Ocurre cuando el usuario selecciona la opción "Copiar". Borra el contenido del portapapeles y copia en el mismo la imagen de la gráfica actual.

mnuExporar_Click () Ocurre cuando el usuario selecciona la opción "Exportar". Muestra un cuadro de dialogo común y guarda en el archivo escogido por el usuario la imagen de la gráfica actual usando el formato bmp.

mnuGuardar_Click () Ocurre cuando el usuario selecciona la opción "Guardar". Despliega una cuadro de dialogo común; crea el archivo seleccionado por el usuario guarda en él los datos del análisis asociado con la gráfica actual.

mnuHelpAbout_Click () Ocurre cuando el usuario selecciona la opción "Acerca de". Muestra la forma frmAcerca.

mnuHelpContents_Click () Ocurre cuando el usuario selecciona la opción "Contenido". Invoca la ayuda del sistema operativo y muestra la ayuda de la aplicación.

mnuHelpSearch_Click () Ocurre cuando el usuario selecciona la opción "Buscar en la ayuda". Invoca la ayuda del sistema operativo y muestra una ventana en donde el usuario puede realizar busquedas dentro de la ayuda de la aplicación.

mnuNuevo_Click () Ocurre cuando el usuario selecciona la opción "Nuevo". Despliega la forma frmAnalisis.

mnuSalir_Click () Ocurre cuando el usuario selecciona la opción "Salir". Cierra la aplicación.

mnuViewOptions_Click () Ocurre cuando el usuario selecciona la opción "Opciones". Despliega la forma frmConfig.

mnuViewToolbar_Click () Ocurre cuando el usuario selecciona la opción "Barra de herramientas". Intercambia la visibilidad de la barra de herramientas.

mnuWindowArrangeIcons_Click () Ocurre cuando el usuario selecciona la opción "Alinear iconos". Ordena los iconos de las ventanas minimizadas dentro de la aplicación.

mnuWindowCascade_Click () Ocurre cuando el usuario selecciona la opción "Cascada". Acomoda las ventanas desplegadas dentro de la aplicación en cascada.

mnuWindowTileHorizontal_Click () Ocurre cuando el usuario selecciona la opción "Exportar".Acomoda las ventanas desplegadas dentro de la aplicación de manera horizontal.

mnuWindowTileVertical_Click () Ocurre cuando el usuario selecciona la opción "Exportar".Acomoda las ventanas desplegadas dentro de la aplicación de manera vertical.

OSWinHelp (hwnd, HelpFile, wCommand, dwData) Declaración externa para ligar la ayuda del sistema operativo con la apliación.

tbToolBar_ButtonClick (Button) Ocurre cuando el usuario presiona alguno de los botones de la barra de herramientas.

Variables

frmD Tipo: frmTraza. Asociación con la forma frmTraza.

numAnal Tipo: Integer. Guarda el número de ventanas desplagadas en la aplicación.

B.6. Forma frmTDN

*Esta forma no es visible para el usuario.

Propiedades

error Tipo: Integer. Regresa el estado del último error encontrado.

Métodos

Todas los metodos siguientes actualizan el estado de (error) al terminar su ejecución.

abortar () Envia al TDN el comando abortar.

analisisFrec (resultado:Analisis, Frec:Long) Tipo: Integer. Efectúa el análisis de respuesta en frecuencia para la frecuencia solicitada representada por el parámetro `Frec`. Regresa los resultados del análisis en en el parámetro `resultado`. Regresa 0 si el análisis se efectúo con éxito, o un valor negativo de lo contrario.

analisisRango (resultado:Analisis, fmin:Double, fmax:Double, suave:Boolean) Tipo: Integer. Efectúa el análisis de respuesta en frecuencia desde fmin hasta fmax (parametros de la función). Regresa los resultados del análisis en el parámetro `resultado`. Regresa 0 si el análisis se efectúo con éxito, o un valor negativo de lo contrario.

calibrar () Efectúa la calibración del TDN, guarda los resultados del mismo en ajustes.

cancelarLectura () Activa dos banderas que indican a los demás métodos abortar la ejecución de los mismos.

cargarS19 (ArchivoS19:String) Envía al TDN un archivo en formato S19, cuya ruta y nombre se encuentran representados por el parámetro `archivoS19`.

com1_OnComm () Evento. Ocurre cuando el sistema recibe datos del TDN.

configurarPuerto (puerto) Abre el puerto serie representado por el parámetro `puerto`.

efectuarAnalisis (relay:Integer, resultados:Variant) Tipo:integer. Indica al TDN efectuar el análisis de respuesta en frecuencia para el rango indicado en el parámetro `relay`. Almacena los resultados obtenidos en el parámetro `resultados`. Regresa 0 si el método tuvo éxito, o un valor negativo de lo contrario.

escMemoria (DirHex:Long, dato:Integer) Escribe en la dirección de memoria del TDN representada por el parámetro `DirHex` el dato indicado. Regresa 0 si el método tuvo éxito, o un valor negativo de lo contrario.

esperar (milseg:Integer) Detiene la ejecución del programa en el punto donde es invocada la función durante un cierto tiempo en milisegundos representado por el parámetro `milseg`.

estasListo () Tipo:integer.Pregunta al TDN si se encuentra listo para ejecución de otro comando.Regresa 0 si el método tuvo éxito, o un valor negativo de lo contrario.

Form_Initialize () Ocurre cuando la forma se carga en memoria. Inicializa las variables de la forma.

iniciaDesde (DirHex:Long) Tipo:integer. Indica al TDN iniciar la ejecución a partir de la dirección de memoria representada por el parámetro `DirHex`. Regresa 0 si el método tuvo éxito, o un valor negativo de lo contrario.

leerDatos (cantidad) Tipo: Variant.Espera a que el TDN le envíe (cantidad) de datos. Conforme los datos llegan se actualiza la forma frmBarra. Regresa los datos obtenidos.

recibirpaquetes () Recibe los paquetes de datos que envía el TDN y los coloca en el buffer de recepción. Utiliza el evento OnComm.

tempo_Timer () Activa la bandera de tiempo fuera para el temporizador 1.

tempo2_Timer () Activa la bandera de tiempo fuera para el temporizador 2.

version () Tipo: byte. Envía al TDN el comando versión y devuelve el resultado del mismo.

Variables

ajustes Tipo: Configuracion. Almacena los ajustes obtenidos por calibrar().

barra Tipo: frmBarra. Asociación con la forma frmBarra.

bufferRecepcion Tipo: CQueue. Buffer de recepción. Almacena los datos que el TDN envía.

cancelar Tipo:boolean. Bandera de cancelación.

cancelarPaq Tipo:boolean. Bandera de cancelación.

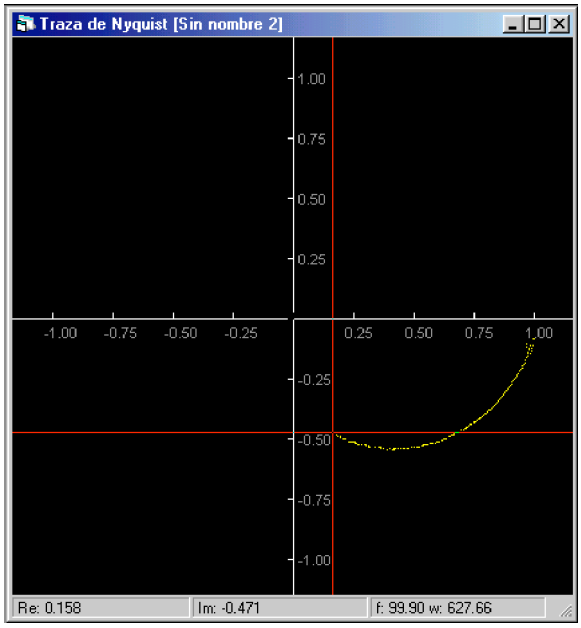
errno Tipo:Integer Variable asociada con la propiedad error.

listo Tipo:Integer. Bandera que indica si el TDN se encuentra listo para ejecutar otro comando.

tiempoFuera Tipo: boolean. Bandera de tiempo fuera para el temporizador 1.

tiempoFuera2 Tipo: boolean. Bandera de tiempo fuera para el temporizador 2.

B.7. Forma frmTraza



Propiedades

nombre Tipo: String. Almacena el nombre de la gráfica.

Métodos

dibujarEjes () Dibuja los ejes coordenados en la gráfica.

dibujarTraza () Dibuja los datos del análisis en la gráfica y el cursor.

Form_Initialize () Ocurre cuando la forma se carga en la memoria. Inicializa las variables de la forma.

Form_Load () Ocurre cuando la forma se muestra por primera vez. Establece el tamaño de la ventana de la gráfica.

Form_QueryUnload (Cancel, UnloadMode) Ocurre cuando el usuario presiona el botón X de la forma. Pregunta al usuario si desea guardar los datos del análisis; guarda el análisis si es necesario.

Form_Resize () Ocurre cuando la ventana de la gráfica cambia de tamaño. Redibuja los ejes y la traza.

Form_Unload (Cancel) Ocurre cuando la forma es descargada de la memoria. Decrementa el número de ventanas abiertas.

graficarDatos (color, guardado) Grafica los datos del análisis en la forma actual.

graficarPunto () Grafica el punto del análisis actual.

picBox_KeyDown (KeyCode, Shift) Ocurre cuando el usuario presiona y mantiene presionadas las teclas de flecha. Mueve el cursor en la dirección de la flecha presionada.

picBox_KeyPress (KeyAscii) Ocurre cuando el usuario presiona una tecla normal. Escala la gráfica en un factor de 2 si la tecla es "+", y en un factor de .5 si la tecla es "-".

picBox_MouseDown (Button, Shift, mX, mY) Ocurre cuando el usuario presiona el botón del ratón sobre la ventana de la gráfica. Coloca el cursor sobre la traza, lo más cercano al lugar donde fue presionado el botón.

Variables

elAnalisis Tipo:Analisis. Contiene los datos del análisis desplegado en la gráfica.

mvarnombre Tipo:String. Variable asociada con la propiedad nombre.

cursor Tipo:Integer. Almacena el indice de posición del cursor dentro de la gráfica.

estado Tipo:Boolean. Indica si la gráfica actual se ha inicializado con los datos de un análisis.

incx, incy Tipo:Double. Valor relativo a los limites de eje que indica la posición de las etiquetas de la escala de los ejes.

xmax, xmin, ymax, ymin Tipo: Double. Limites de los ejes coordenados.

B.8. Clase Analisis

Propiedades

datos Tipo: Variant. Los datos del análisis.

estado Tipo: Boolean. El estado del análisis: Falso, no guardado; Verdadero, guardado.

Métodos

ajustar () Ajusta los puntos del arreglo obteniendo el centroide de 10 puntos a la vez.

Class_Initialize () Inicializa las variables de la clase.

escoger (fmin:Double, fmax:Double) Discrimina los datos del análisis. Elimina los datos no comprendidos entre (fmin) y (fmax).

guardar (ubicacion:String) Tipo: Integer. Guarda los datos en el archivo indicado por el parámetro `ubicacion`. Regresa 0 si el método tuvo éxito, o un valor negativo de lo contrario.

leer (ubicacion) Tipo:Integer. Lee del archivo indicado por el parámetro `ubicacion` los datos del análisis.

mvardatos Tipo: Variant. Variable asociada con la propiedad datos.

mvarestado Tipo: Boolean. Variable asociada con la propiedad estado.

B.9. Clase Configuracion

Propiedades

coeficientes Tipo: Variant. Coeficientes de las ecuaciones de cada rango.

fccX, fccY Tipo:Double. Factores de corrección de los canales X y Y.

kdp Tipo: Double. Ganancia del detector de pico.

limites Tipo: Variant. Límites de cada rango.

Métodos

Class_Initialize () Inicializa las variables de la clase.

eDac (rango:Integer, Frec:Long) Tipo: byte. Ecuación byte=dac(frec,rango). Regresa el byte necesario para generar la frecuencia representada por el parámetro `Frec`, en el rango solicitado indicado por el parámetro `rango`.

eFrec (rango, dac) Tipo: Double. Ecuación frec=Frec(byte,rango). Regresa la frecuencia que se genera al utilizar el byte indicado en el rango seleccionado.

guardar () Tipo:Integer. Guarda las variables de la clase en el archivo "ajustes.ini". Regresa 0 si el método tuvo éxito, o un valor negativo de lo contrario.

leer () Tipo:Integer. Recupera del archivo "ajustes.ini" los valores de las variables de clase.

mvarcoeficientes Tipo: Variant. Variable asociada a la propiedad coeficientes.

mvarfccX, mvarfccY Tipo:Double. Variable asociada a la propiedades fccX y fccY.

mvarkdp Tipo: Double.Variable asociada a la propiedad kdp.

mvarlimites Tipo: Variant. Variable asociada a la propiedad limites.

B.10. Clase CQueue

Propiedades

Count Tipo:Long. Contiene el número de elementos en la cola.

IsEmpty Tipo: Boolean. Indica si la cola está vacía.

Métodos

Class_Initialize () Inicializa las variables de clase.

Class_Terminate () Borra las variables de clase.

Clear () Elimina los elementos de la cola.

Dequeue () Tipo: Byte. Entrega el primer elemento de la cola.

Enqueue (vItem: Byte) . Coloca el elemento indicado en el parámetro `vItem` en la cola.

Peek () Tipo: byte. Entrega el primer elemento de la cola sin sacarlo de la misma.

m_avItems Tipo:Long. Variable asociada a la propiedad Count.

m_lCount Tipo: Boolean. Variable asociada a la propiedad IsEmpty.

B.11. Módulo Funciones

Funciones

minimosRecta(datos():Double) Tipo: Variant. Ajusta el arreglo de datos mediante el método de los mínimos cuadrados. Regresa los coeficientes obtenidos (m,b0): Y=mx+b0.

Variables

cconfig Tipo: Boolean. Bandera que indica si la configuración ha cambiado.

direjec Tipo: String. Contiene la ruta del directorio desde donde se ejecuta la aplicación.

dirusr Tipo: String. Contiene la ruta del directorio donde el usuario ha guardado o leído algún archivo.

ventVis Tipo: Integer. Contiene el número de ventanas abiertas.

Apéndice C

Guía básica de operación

C.1. Introducción

El trazador digital de Nyquist (TDN) se opera mediante una PC y un programa desarrollado para tal efecto. Tradiny es el software de interfaz de usuario desarrollado para el TDN; el cual, establece una comunicación entre la PC y el TDN por medio del puerto serie. Dicho programa se desarrolló para el sistema operativo Windows, utilizando el lenguaje de programación Visual Basic 5.

Tradiny permite al usuario efectuar el análisis de respuesta en frecuencia de tres maneras posibles:

- Análisis para una frecuencia particular.
- Análisis de frecuencias por décadas.
- Análisis para un rango de frecuencias definido por el usuario.

además de estos análisis, Tradiny se encarga del proceso de calibración del TDN descrito en la sección 3.4.1.

C.2. Requerimientos de operación

Tradiny requiere una PC con la siguiente configuración mínima:

- Procesador Pentium a 120 MHz
- 32 MB en memoria RAM
- 8 MB de espacio en disco duro
- Puerto serie estándar (RS-574 IBM PC/AT)
- Sistema operativo Windows 95 o superior

Figura C.1: Trazador digital de Nyquist

C.3. Instalación del software Tradiny

Esta tesis viene acompañada por un disco compacto que incluye: este trabajo en formato digital, los programas desarrollados para la tarjeta Facil11, y el software de interfaz Tradiny; tanto su código fuente como su versión ejecutable.

La forma de instalación de Tradiny, es la acostumbrada para todas las aplicaciones de Windows. Un programa de instalación llamado "setup.exe"guía al usuario a través del proceso de instalación; mismo, que al finalizar crea un acceso directo al programa en el menú de inicio de Windows. El programa de instalación se encuentra en el directorio "exe"del disco compacto.

C.4. Instalación del hardware

El TDN (ver fig. C.1) se debe conectar al puerto serie del PC, utilizando un cable serie de 9 pines. Posteriormente se debe conectar una fuente bipolar de 12[v] a los bornes correspondientes en el TDN como se ilustra en la fotografía de la figura (C.3)

Hecho lo anterior se debe conectar el sistema a analizar (ver fig. C.5). Para lo cual, es necesario conectar la señal "SAL"del TDN, a la entrada del sistema a analizar y conectar la salida del mismo la entrada "SIS"del TDN. Seguido esto, se conecta la tierra del TDN, con la tierra del sistema a analizar.

Por último, se acciona el interruptor de encendido de la fuente y se alimenta el sistema a analizar (en caso de requerirlo). Se debe verificar que el led "ENC"se encuentre iluminado y que el led "ESP"se encuentre iluminado intermitentemente. Si lo anterior no se cumple habrá que revisar las conexiones y verificar los voltajes de alimentación con un multímetro.

Figura C.2: Descripción de terminales del panel trazero.

Figura C.3: Conexión de la alimentación.

Figura C.4: Descripción de terminales del panel delantero.

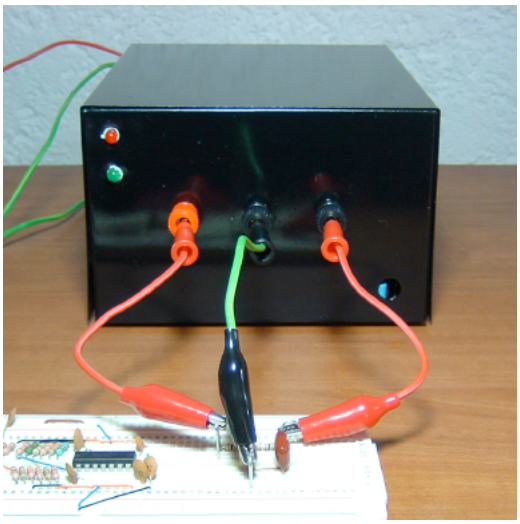


Figura C.5: Conexión de un sistema de prueba.

C.5. Operación de Tradiny

Cuando Tradiny se ejecuta por primera vez, después de llevar a cabo la instalación, aparece una ventana en donde se selecciona el puerto serie a utilizar, como se muestra en la figura C.6.

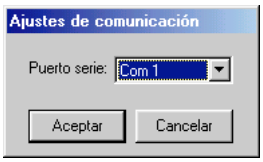


Figura C.6: Cofiguración del puerto serie.

Posteriormente se ejecuta automáticamente la calibración del TDN solicitando al usuario conectar un puente entre las terminales “TIERRA” y “SIS” como se muestra en la figura (C.7).

Durante este proceso, se obtienen los valores de frecuencia posibles para efectuar los análisis de respuesta en frecuencia, y algunos ajustes necesarios para la ejecución propia del TDN. Antes de iniciar la calibración se debe colocar un cable entre las terminales “Tierra” y “Sistema” del TDN. El proceso despliega una barra de progreso durante su ejecución (figura C.8, ya que la calibración tarda aproximadamente un minuto. Lo que sucede durante la calibración se explica en detalle en la sección 3.4.1.

Lo anterior solo se ejecuta la primera vez que se utilice el software; sin embargo, recomendamos efectuar la calibración cada vez que se utilice el TDN ya que la temperatura

Figura C.7: Conexión para el proceso de calibración.

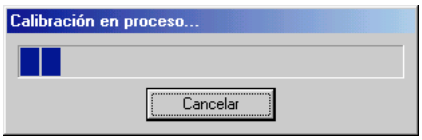


Figura C.8: Avance de la calibración.

es un factor que altera los resultados de los análisis.

En la figura C.9, se muestra la pantalla principal del software. Dentro del menú "Análisis" se encuentran las funciones primarias del software:

- Nuevo: Permite efectuar un nuevo análisis de respuesta en frecuencia.
- Abrir...: Abre un análisis guardado con anterioridad, con el fin de mostrar la gráfica correspondiente.
- Guardar...: Guarda los resultados del análisis actual en un archivo.
- Exportar gráfica: Guarda una copia de la gráfica actual utilizando el formato bmp.

Cuando se selecciona la opción "Nuevo" del menú "Análisis" aparece una ventana que muestra los tipos de análisis que se pueden realizar (figura C.10). Con la opción "Análisis para una frecuencia particular" el TDN aplicará una señal senoidal, de frecuencia definida por el usuario, al sistema en análisis; para después mostrar una gráfica con un solo punto que representa la componente real e imaginaria de la traza de Nyquist para la frecuencia solicitada. Mediante la opción "Análisis en un rango de frecuencias"el TDN generará una señal senoidal de frecuencia variable, que oscilará entre los límites de frecuencia establecidos por el usuario, para después mostrar en pantalla la traza de Nyquist obtenida del sistema en análisis.

Por ejemplo, para el sistema de prueba descrito en la sección 5.2, se obtiene la gráfica de la figura C.11, al realizar el análisis en el rango de frecuencias de 10 a 1000 [Hz] C.11

Figura C.9: Pantalla principal.

Figura C.10: Análisis posibles.

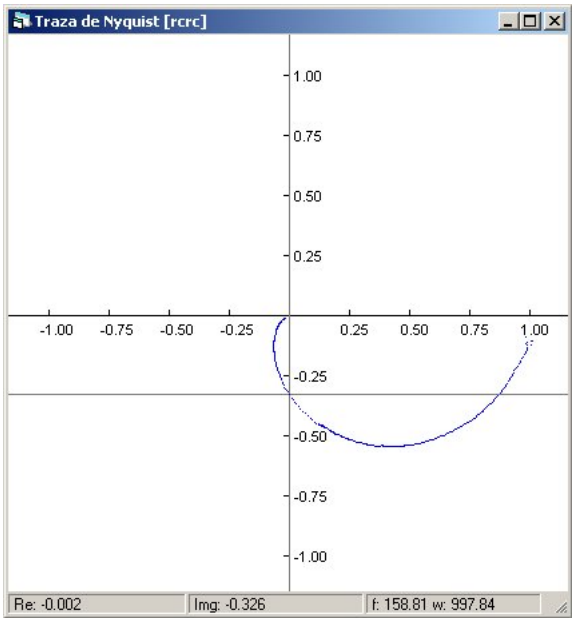


Figura C.11: Traza de Nyquist de un filtro paso bajas pasivo de segundo orden.

Bibliografía

[1] Boylestad, Robert L. *Análisis introductorio de circuitos.* Ed. Prentice-Hall, octava edición, México, 1997.

[2] Boylestad, Robert L; Nashelsky, Louis. *Electrónica: Teoría de circuitos.* Ed. Pearson, sexta edición, México, 1997.

[3] EXAR Corporation. *XR2206 Monolithic Function Generator Datasheet.* EEUU, 1997.

[4] Fowler, Martin; Scott, Kendall. *UML gota a gota.* Ed. Pearson, México, 1999.

[5] Larman, Craig. *UML y patrones. Introducción al análisis y diseño orientado a objetos.* Ed. Prentice Hall, México,1999.

[6] Maron, Melvin J.; Lopez, Robert J., *Análisis numérico. Un enfoque práctico.* Compañia editorial continental, S.A. de C.V., México, 1995.

[7] Ogata, Katsuhiko. *Ingeniería de control moderno.* Ed. Pearson, México, 1999.

[8] Salvá Calleja, Antonio. *Programas interactivos para el diseño de filtros activos.* Memoria de la IX Conferencia internacional de investigación y desarrollo en Ingeniería Eléctrica. Cuernavaca Morelos, México,1983.

[9] Salvá Calleja, Antonio; Sánchez Esquivel,Víctor. *Filtros activos: Análisis y diseño.* Facultad de Ingeniería, UNAM. México, 1992.

[10] Salvá Calleja, Antonio; Escalante Ramírez, Boris; Sánchez Esquivel, Víctor. *Diseño y construcción de un graficador de la traza polar de Nyquist y vectoroscopio.* Memoria de la conferencia internacional de investigación, desarrollo y aplicación en Ingeniería Eléctrica Electrónica Comunicaciones y Computación. México, D.F., 1984.

[11] Tobey G.E.; Graeme J.D.; Huelsman L.P. *Operational amplifiers, design and applications.* Ed. Mc Graw-Hill, EEUU, 1974.