



**INSTITUTO POTOSINO DE INVESTIGACIÓN  
CIENTÍFICA Y TECNOLÓGICA, A.C.**

**POSGRADO EN CIENCIAS APLICADAS**

**Implementación de un controlador para un filtro  
activo serie usando un procesador digital de  
señales**

Tesis que presenta

**Michael Hernández Gómez**

Para obtener el grado de

**Maestro en Ciencias Aplicadas**

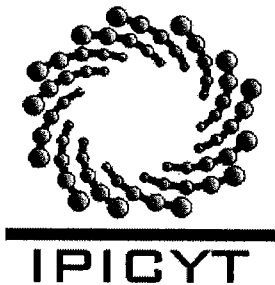
En la opción de

**Control y Sistemas Dinámicos**

Director de la Tesis:

Dr. Gerardo Escobar Valderrama

San Luis Potosí, S.L.P., Septiembre de 2004.



# Instituto Potosino de Investigación Científica y Tecnológica, A.C.

## Acta de Examen de Grado

COPIA CERTIFICADA

El Secretario Académico del Instituto Potosino de Investigación Científica y Tecnológica, A.C., certifica que en el Acta 004 del Libro Primero de Actas de Exámenes de Grado del Programa de Maestría en Ciencias Aplicadas en la opción de Control y Sistemas Dinámicos está asentado lo siguiente:

En la ciudad de San Luis Potosí a los 30 días del mes de septiembre del año 2004, se reunió a las 09:00 horas en las instalaciones del Instituto Potosino de Investigación Científica y Tecnológica, A.C., el Jurado integrado por:

<b>Dr. Jesús Leyva Ramos</b>	<b>Presidente</b>	<b>IPICYT</b>
<b>Dr. Gerardo Escobar Valderrama</b>	<b>Secretario</b>	<b>IPICYT</b>
<b>Dr. Jorge Alberto Morales Saldaña</b>	<b>Sinodal externo</b>	<b>UASLP</b>

a fin de efectuar el examen, que para obtener el Grado de:

**MAESTRO EN CIENCIAS APLICADAS  
EN LA OPCIÓN DE CONTROL Y SISTEMAS DINÁMICOS**

sustentó el C.

**Michael Hernández Gómez**

sobre la Tesis intitulada:

*Implementación de un controlador para un filtro activo serie usando un procesador digital de señales*

que se desarrolló bajo la dirección de

**Dr. Gerardo Escobar Valderrama**

El Jurado, después de deliberar, determinó

**APROBARLO**

Dándose por terminado el acto a las 10:30 horas, procediendo a la firma del Acta los integrantes del Jurado. Dando fé el Secretario Académico del Instituto.

A petición del interesado y para los fines que al mismo convengan, se extiende el presente documento en la ciudad de San Luis Potosí, S.L.P., México, a los 30 días del mes septiembre de 2004.



  
**Mtra. Ma. Elisa Lucio Aguilar**  
Jefa del Departamento de Asuntos Escolares

*Al creador, a mi familia, el más magnífico  
soporte de mi enseñanza, así como al amor  
de mi vida y a todas aquellas personas  
que han dado sentido a mi vida*

## AGRADECIMIENTOS

---

*Primero a DIOS, que sin el nada soy.*

*A mi familia, a los cuales les debo mucho y que sin ellos nada de esto hubiera sido realidad.*

*Al Dr. Gerardo Escobar Valderrama el cual me guió por a lo largo de esta empresa, no solo como asesor, sino como un verdadero amigo.*

*A mis maestros Dr. Jesús Leyva, Dr. Haret Rosu, Dr. Hugo Cabrera, Dr. David Lizárraga, Dr. Arturo Zavala, Dr. Ricardo Femat por su amistad y ejemplo.*

*A mis compañeros y amigos en el IPICYT Alejandra Herrera☺, Misael, Gabriel, Vicente, Raúl, Pánfilo, Octavio, Brani, Sosa, Andrés, Rafa, Chen por su apoyo y amistad en este periodo de mi vida y espero que siga contando con ellos.*

*A mis amigos Samuel, Nashelli, el Boti, a Víctor Morales, Cliffor, Beto, el güero, el Verde, Emeterio, Perla y su hermana Adriana, Memo, Leti, Miguel Ángel Mota, Sergio, Braulio, Arisbe, Jaime y a todos aquellos que omití y sin embargo hicieron un sendero de mi vida.*

*Al programa de becas crédito de CONACYT por el financiamiento que proporcionaron para poder concluir este trabajo.*

## RESUMEN

---

En la presente tesis se propone un controlador para el filtro activo serie, también conocido como DVR, destinado a aliviar el problema de la distorsión del voltaje de la línea. Conociendo el modelo matemático de este último, más algunas suposiciones hechas con base en las características del problema y gracias a que la distorsión en la línea es periódica, se ha podido proponer un controlador con base en las técnicas de control adaptable y basadas en pasividad. Este controlador garantiza la compensación de componentes armónicos seleccionados de la señal de voltaje de la línea. La expresión del controlador es reducida, a través de algunas transformaciones, en la interconexión en cascada de dos lazos de control, llamados lazo interno y lazo externo, tal y como se presenta en el controlador convencional, facilitando así su implementación. Se muestra que el lazo externo es básicamente un controlador proporcional-integral con ancho de banda limitado, mientras que el lazo interno contiene un banco de filtros resonantes sintonizados a las frecuencias a compensar. Este banco de filtros resonantes aparece como un refinamiento al lazo interno y es finalmente el término que ayuda a compensar los efectos de la distorsión armónica en la tensión de línea.

El controlador propuesto es programado en una tarjeta de evaluación del procesador digital de señales DSP TMS320LF2407A, específicamente se usa el formato de punto fijo de 32 bits. Esto hace más exacto el proceso, manteniendo un tiempo de ejecución reducido. Cabe remarcar que el código del programa se ha realizado en lenguaje ensamblador. Gracias a esto se logró aumentar la frecuencia de muestreo hasta 20 KHz. En el presente reporte se describen brevemente las operaciones en formato de punto fijo en ensamblador, así como el código para la implementación del controlador.

Para probar el control propuesto sin armar un prototipo de pruebas, y así ahorrar tiempo, se implementó el sistema compuesto por el DVR, la fuente de tensión distorsionada y la carga no lineal en una tarjeta dSPACE DS1103 dispuesta en una computadora personal.

La interconexión de ambas tarjetas (DSP y dSPACE) se realizó por medio de tarjetas de interfaz que tienen la finalidad de proteger a ambas. Finalmente se presentan algunas gráficas de los resultados de la evaluación experimental del controlador propuesto. Entre estos resultados cabe destacar la reducción de la distorsión armónica total (THD) de la tensión de la línea, la cual era originalmente del 11.62% (un poco mayor a la distorsión armónica que suele haber comúnmente) y se redujo al 1.22% gracias a la compensación propuesta.

# INDICE

---

<b>CAPITULO 1.- INTRODUCCION Y FORMULACION DEL PROBLEMA</b>	<b>1</b>
1.1. Introducción	1
1.2. Formulación del problema	4
1.3. Notación	6
1.4. Modelo matemático	6
1.5. Objetivos de control	7
1.6. Suposiciones	8
<b>CAPITULO 2.- DISEÑO Y DISCRETIZACION DEL CONTROLADOR</b>	<b>11</b>
2.1. Lazo interno	11
2.2. Lazo externo	16
2.3. Discretización del controlador	20
<b>CAPITULO 3.- EMULACION DE LA PLANTA</b>	<b>25</b>
3.1. Descripción del sistema de emulación	25
3.2. Características de la dSPACE	26
3.3. Implementación del sistema en la tarjeta dSPACE	27
<b>CAPITULO 4.- IMPLEMENTACION DEL CONTROLADOR EN EL DSP</b>	<b>35</b>
4.1. Introducción	35
4.2. Aritmética de punto fijo	36
4.3. Operaciones aritméticas con el formado de 32 bits	38
4.4. Conversión de los valores adquiridos por el ADC al formato de punto fijo	40
4.5. Implementación por bloques del algoritmo de control en la DSP	41
4.6. Descripción del código del programa principal en el DSP	43
4.7. Evaluación experimental	47
4.8. Gráficas para los resultados en la salida	48

INDICE

<b>CAPITULO 5.- CONCLUSIONES Y TRABAJO A FUTURO</b>	<b>59</b>
<b>APENDICE A.- ARQUITECTURA INTERNA DEL DSP TMS320LF2407A</b>	<b>61</b>
<b>APENDICE B.- CODIGO DEL PROGRAMA</b>	<b>75</b>
<b>APENDICE C.- DESCRIPCION DE LOS MACROS UTILIZADAS</b>	<b>91</b>
<b>APENDICE D.- PUERTOS Y TARJETAS DE INTERFAZ</b>	<b>103</b>
<b>REFERENCIAS</b>	<b>109</b>

# CAPITULO 1

---

## INTRODUCCION Y FORMULACION DEL PROBLEMA

*En este capítulo se presenta la motivación, el estado del arte, y las contribuciones de la propuesta hecha en este trabajo, las cuales se explicarán más detalladamente en los siguientes capítulos. Además se establece el modelo matemático de la topología del filtro activo serie bajo estudio y las principales suposiciones que sirvieron de base para el diseño de la ley de control*

### **1.1. Introducción**

#### *Motivación*

Varios eventos durante el transcurso de este siglo han influenciado el cambio radical de las cargas eléctricas y los servicios de los sistemas de distribución. Se podría situar como el primero de estos eventos a la invención del transistor, de donde nace la electrónica de potencia surgiendo como consecuencia inmediata las cargas no lineales. Un segundo evento consiste en el problema de la escasez de energéticos ocurridos en los años setenta donde la electrónica se masifica con el afán de producir equipos más eficientes con menor consumo de energía. Finalmente, una tercera influencia mundial ocurre afines de los ochenta donde se busca mejorar la productividad con equipos más eficientes, conjugando así la reducción de costos y a su vez tratando de ofrecer servicios de mayor calidad [1].



A principios de este siglo las cargas eléctricas típicas eran básicamente de tipo resistivo, inductivo o capacitivo, es decir, cargas lineales pasivas. Resulta paradójico entonces, que a principios del siglo XXI las nuevas cargas eléctricas son, en un alto porcentaje, cargas no lineales. Si se revisa la tendencia actual del desarrollo tecnológico, se encontrará que muchos dispositivos tienen, entre sus componentes internos, fuentes de alimentación de modo conmutado, o bien simples rectificadores no controlados de diodos [2].

Hoy en día, en muchas empresas eléctricas, a nivel mundial, se habla mucho de la eficiencia, de la optimización de recursos, de la reducción de costos y de la calidad en todos los aspectos que esta pueda estar entendida [1]. Debido a lo extenso de estos problemas, ha sido necesario desarrollar técnicas y lineamientos para la instalación de equipos y control de armónicos. Por ejemplo, las normas estadounidenses con respecto a los armónicos han sido agrupadas por la IEEE en la norma 519, la cual es básicamente un conjunto de recomendaciones para limitar la distorsión armónica inyectada a la red por el consumidor, y la suministrada por la compañía de distribución [4].

### *Antecedentes*

Existen diversas propuestas de sistemas encaminados al mejoramiento de la calidad de la energía eléctrica los cuales han sido agrupados bajo el título de sistemas flexibles de transmisión de corriente alterna (FACTS) [5]. Entre estos destacan los filtros activos, cuya construcción se basa en los inversores de voltaje (VSI) que son dispositivos compuestos de ramas de interruptores conectados en serie y trabajando en forma complementaria. Dependiendo de la aplicación, estos interruptores pueden ser construidos con dispositivos transistores de oxido metálico de efecto de campo (MOSFET) y transistores bipolares de compuerta aislada (IGBT), entre otros.

La idea detrás de los filtros activos es que, controlando los estados de encendido y apagado de los interruptores, y auxiliados de filtros pasivos, es posible reconstruir señales periódicas destinadas a compensar la distorsión en las señales de interés. Los filtros activos que serán tratados a lo largo del presente proyecto son filtros activos serie mejor conocidos comercialmente bajo el nombre de restauradores dinámicos de voltaje (DVR) [6], y cuya topología puede observarse en la Fig 1.1. Estos dispositivos, a diferencia de los simples capacitores, ofrecen un gran número de ventajas en el control de sistemas de potencia, que incluyen mayor velocidad y precisión de las respuestas, logrando así un mejor desempeño y una mayor versatilidad.

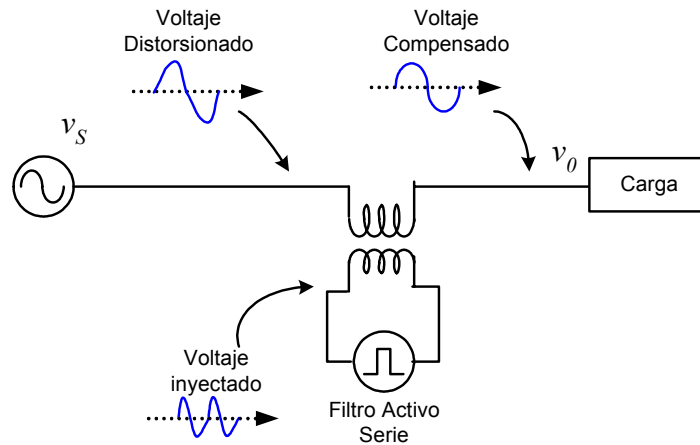


Fig 1.1. Topología del Filtro Activo Serie.

Los filtros activos serie, también conocidos bajo el nombre de restauradores dinámicos de voltaje (DVR), se introdujeron en los ochentas con la intención de operarlos principalmente como reguladores de voltaje para compensar caídas e incrementos de tensión, así como desbalance de voltaje. La idea básica es inyectar el voltaje necesario tal que sumado al voltaje distorsionado proveniente de la fuente, se pueda ofrecer a la carga un voltaje compensado de mejor calidad. Diferentes métodos han sido propuestos en la literatura para realizar el control de dichos filtros [7], [8] y [9], siendo la teoría de la potencia instantánea (PQ) [10] la más utilizada a la fecha. A esta técnica todavía se le puede añadir la descripción de variables en términos de sus componentes simétricas a fin de poder tratar la compensación de desbalance y de la componente homopolar, algunos resultados han sido ya presentados en [11], [12] y [13].

### *Contribución*

Las contribuciones principales de este trabajo son las siguientes:

*-Propuesta de un control novedoso.* El diseño de este controlador está basado en la inversión parcial de la planta, control adaptable [14], así como en ideas de pasividad [3]. El controlador propuesto presenta las ventajas de ser robusto con respecto a variación e incertidumbre en los parámetros del sistema. El controlador es también inmune a la distorsión de la corriente de carga. Utiliza un número reducido de señales para su implementación. Incorpora un lazo externo de control el cual regula la tensión del capacitor del lado de continua, por lo que se evita el uso de una fuente externa de tensión. Este controlador pertenece a la clase de compensadores de armónicos selectivos, ya que el controlador permite la compensación de un conjunto seleccionado de armónicos. Se trató de hacer el controlador lo más sencillo posible pensando en una aplicación práctica, realizable en algún controlador digital, en este caso un procesador digital de señales (DSP). De hecho el controlador obtenido es muy cercano al convencional, excepto que aparece un banco de filtros resonantes,

derivados de la aplicación de transformaciones a las leyes adaptables. Estas últimas, destinadas a compensar los armónicos seleccionados.

*-Programación del controlador propuesto en el DSP TMS320LF2407A.* La programación del controlador propuesto en el siguiente capítulo es implementada en el DSP TMS320LF2407 de Texas Instruments. El código en su totalidad esta escrito en ensamblador que es el lenguaje de más bajo nivel. La programación en ensamblador de este controlador aumenta el poder de procesamiento con respecto a otros lenguajes de programación, ya que es el más cercano al código de máquina. Tomando en cuenta la rapidez de la aplicación presentada en esta tesis, se trato de reducir al máximo el código del programa. La meta para el periodo de muestreo se fijo a 50  $\mu$ s equivalente a una frecuencia de 20 kHz. Por ello se eligió la aritmética de punto fijo para realizar las operaciones dentro del DSP. La ventaja del punto fijo es que utilizando la propia arquitectura del DSP es posible realizar las operaciones del controlador de forma rápida. Por desgracia, la resolución de estas operaciones esta limitada por un rango que depende del tamaño de la palabra utilizada para representar los números en este formato.

### *Contenido de la tesis*

El Capítulo 1 presenta la motivación, el estado del arte y las contribuciones de este proyecto, así como el planteamiento el problema. Esto último incluye la modelización matemática del sistema bajo estudio, las suposiciones principales y el establecimiento de los objetivos de control. Teniendo bien fundamentado el problema, en el Capítulo 2 se procede a desarrollar la ley de control propuesta, así como su discretización. En el Capítulo 3 se presenta la implementación de la planta, integrada por la fuente de voltaje, el DVR y la carga no lineal, utilizando para ello la tarjeta dSPACE. La programación del algoritmo de control desarrollado en el Capítulo 2 se muestra en el Capítulo 4, en donde se da una breve descripción de las principales partes que conforman al código del programa. En este mismo capítulo, se presentan las graficas de los resultados de la evaluación experimental del controlador propuesto. Finalmente en el Capítulo 5 se plantean las conclusiones y los planes a futuro.

## **1.2. Formulación del problema**

En esta sección se obtiene el modelo matemático de la topología del filtro activo serie bajo estudio y se plantean las principales suposiciones que sirvieron de base para el diseño de la ley de control.

La topología de filtro activo serie que se seleccionó se presenta en la Fig. 1.2. Esta topología incluye los siguientes elementos: un transformador de acoplamiento en serie, un inversor de voltaje y un filtro de alisamiento. El funcionamiento de cada uno de estos elementos se describe a continuación.

### **Transformador**

Elemento cuyo secundario se conecta en serie entre la fuente de voltaje y la carga. Su función es inyectar el voltaje de compensación proveniente del filtro activo serie. Su relación de transformación permite manejar normalmente una corriente 10 veces inferior en el lado del filtro activo serie, aunque a un voltaje 10 veces mayor. Esto se hace con el fin de reducir el efecto de la distorsión de la corriente en el lado del filtro activo.

### Inversor de voltaje (VSI)

Integrado por dos ramas de interruptores  $S_1$ - $S_2$  y  $S_3$ - $S_4$ , como se muestra en la Fig. 1.2. Estas ramas están conectadas en su lado de continua (DC) a un capacitor  $C$  de capacitancia relativamente elevada. A diferencia de otras topologías que usan una fuente de voltaje como suministro de energía, se ha preferido mantener solo al capacitor, ya que esto involucra problemas de control de mayor interés.

### Filtro de alisamiento

Este filtro permite reducir los efectos de la conmutación del VSI. Esta conformado por el arreglo de un inductor  $L_r$  y un capacitor  $C_r$ , como se muestra en la Fig. 1.2, y se encarga de suavizar la señal entregada por el inversor, esto es eliminar los efectos de conmutación.

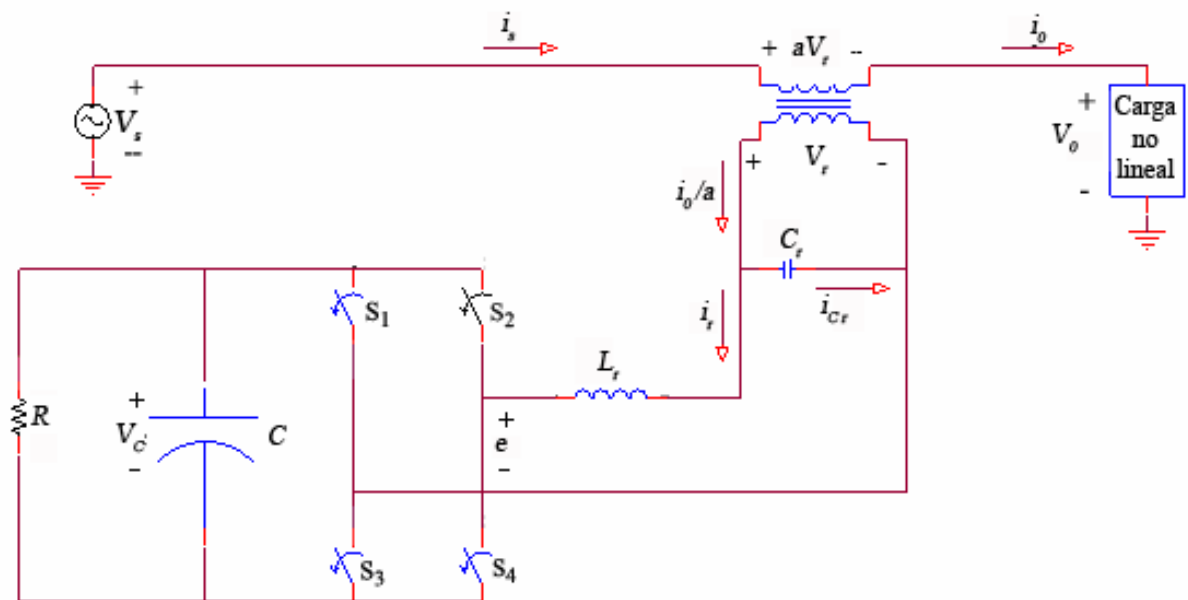


Fig 1.2. Diagrama esquemático del filtro activo serie.

### 1.3. Notación

Es importante que la conmutación de los arreglos de interruptores tanto  $S_1$ - $S_3$  como  $S_2$ - $S_4$  sea complementaria, esto es, mientras los interruptores  $S_1$  y  $S_4$  estén encendidos, los interruptores  $S_2$  y  $S_3$  deberán estar apagados, y viceversa. De lo contrario se cortocircuitarían las terminales del capacitor poniendo en riesgo al VSI. Más aún, es necesario incluir tiempos muertos para permitir una conmutación más segura.

Con base en el dibujo esquemático de la Fig. 1.2 se obtiene el modelo matemático del sistema, en donde se usa la siguiente notación:

$u$	Entrada de control (ciclo de trabajo), $u \in [-1,1]$
$V_s$	Voltaje de línea
$i_L$	Corriente de línea
$i_0$	Corriente de carga
$aV_r$	Voltaje inyectado por el DVR
$S_1, S_2, S_3, S_4$	Posiciones de los interruptores $S_i \in \{0,1\}$ , ( $i = 1,2,3,4$ )
$L_r$	Valor de la inductancia
$C_r$	Valor de la capacitancia
$R$	Valor de la resistencia de pérdidas
$i_r$	Corriente que pasa por la inductancia del filtro de alisamiento
$i_{cr}$	Corriente que pasa por el capacitor del filtro de alisamiento
$e$	Voltaje inyectado por el VSI
$V_C$	Voltaje del capacitor del lado de DC

### 1.4. Modelo matemático

El control del DVR se realiza por medio de la conmutación de los interruptores  $S_1$ - $S_2$  y  $S_3$ - $S_4$  de la siguiente manera:

$$S_1, S_4 \text{ encendidos, por lo tanto } e = -V_C,$$

$$S_2, S_3 \text{ encendidos, por lo tanto } e = V_C$$

Dado que ambas ramas de interruptores funcionan en forma complementaria solo existe una secuencia de conmutación generada usualmente por una modulación de ancho de pulso (PWM), la cual estará gobernada por un solo ciclo de trabajo  $u$  que es una señal continua y acotada en el rango  $[-1,1]$ . A partir de este momento suponemos la señal portadora del PWM esta a una frecuencia relativamente elevada, por lo que se considerará a la señal  $u$  como la entrada de control. De esta forma podemos escribir

$$e = uV_C, \quad u \in [-1,1]$$

Bajo las consideraciones anteriores y aplicando las leyes de voltajes y corrientes de Kirchhoff, se obtienen las siguientes ecuaciones que describen la dinámica del DVR:

$$C_r \frac{dV_r}{dt} = ai_0 - i_r \quad (1.1)$$

$$L_r \frac{di_r}{dt} = V_r - e \quad (1.2)$$

$$C \frac{dV_C}{dt} = ui_r - \frac{V_C}{R} \quad (1.3)$$

$$aV_r = V_s - V_0 \quad (1.4)$$

Se nota que la ecuación (1.3) también se puede expresar como el balance de energía almacenada en el capacitor.

$$V_C C \frac{dV_C}{dt} = ei_r - \frac{V_C^2}{R} \quad (1.5)$$

## 1.5. Objetivos de control

*Seguimiento del voltaje de salida* hacia una referencia.

$$V_0 \rightarrow V_0^* = V_s^1 + \rho i_0^1 \quad (1.6)$$

El primer objetivo se define como el seguimiento del voltaje de salida a la componente fundamental del voltaje de la fuente, la cual es tomada como referencia. Existe, sin embargo, la necesidad de mantener cargado al capacitor del lado de DC, por lo cual, se incluye un término compuesto por el producto de la variable escalar  $\rho$  multiplicando a la componente fundamental de la corriente de carga. El cálculo de la variable  $\rho$  se explica más adelante, aunque es claro que su valor deberá tender a una constante para evitar mayor distorsión.

*Regulación del voltaje  $V_C$*  hacia un valor de referencia constante predeterminado  $V_D$ .

$$V_C \rightarrow V_C^* = V_D \quad (1.7)$$

Como segundo objetivo esta la regulación del voltaje en el capacitor de continua hacia una referencia constante  $V_D$ . El capacitor debe mantenerse cargado a un voltaje

constante para así garantizar la existencia de energía disponible para llevar a cabo el objetivo de seguimiento antes mencionado.

## 1.6. Suposiciones

- 1) Se considera que las señales de entrada son periódicas; y por lo tanto, se pueden representar en el dominio de la frecuencia de la siguiente manera:

$$V_S = \sum_{k \in H} \rho_k^T V_{S,k} \quad (1.8)$$

$$i_0 = \sum_{k \in H} \rho_k^T I_{0,k} \quad (1.9)$$

$$V_{S,k} \triangleq \begin{pmatrix} a_k \\ b_k \end{pmatrix}, \quad I_{0,k} \triangleq \begin{pmatrix} c_k \\ d_k \end{pmatrix}, \quad \rho_k \triangleq \begin{pmatrix} \cos(k\omega t) \\ \sin(k\omega t) \end{pmatrix}, \quad \forall k \in H \quad (1.10)$$

donde  $V_{s,k}$  e  $I_{0,k}$  representan los fasores de las señales de línea, y H representa un conjunto finito de los índices de componentes armónicos, esto es, suponemos que las señales de perturbación contienen un número finito de componentes armónicos, los cuales serán considerados para su compensación en la etapa de diseño de control. Más aún, en adelante se usará únicamente  $\sum_k$  en lugar de  $\sum_{k \in H}$  por facilidad la notación.

Esta notación resulta muy conveniente para facilitar la manipulación de las expresiones matemáticas, tal y como se muestra en el siguiente ejemplo donde se calcula la derivada temporal de  $V_S$ :

$$\dot{V}_S = \sum_{k=1}^{\infty} \dot{\rho}_k^T V_{S,k} \quad (1.11)$$

donde el valor de la derivada temporal de  $\rho_k$  se expresa como

$$\dot{\rho}_k = \begin{pmatrix} -k\omega \cdot \sin(k\omega t) \\ k\omega \cdot \cos(k\omega t) \end{pmatrix} = k\omega J \rho_k, \quad J \triangleq \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Sustituyendo lo anterior en la ecuación (1.11) obtenemos la derivada

$$\dot{V}_S = \sum_{k=1}^{\infty} k\omega \rho_k^T J V_{S,k}$$

Así mismo, para el caso de la derivada de la corriente de carga se tiene

$$\dot{i}_0 = \sum_{k=1}^{\infty} k \omega \rho_k^T J I_{s,k}$$

- 2) El valor de los parámetros del sistema  $L_r$ ,  $C_r$  y  $C$  puede cambiar dentro de un cierto rango alrededor del valor nominal, sin embargo, para propósitos de diseño del controlador se suponen como constantes desconocidas.
- 3) Los vectores  $V_{s,k}$  e  $I_{0,k}$  ( $k = 1, \dots, \infty$ ) de los componentes armónicos de las señales de voltaje de línea y de corriente de carga, respectivamente, son supuestos vectores constantes desconocidos.
- 4) Se considera a la frecuencia  $\omega$  de la línea como una constante conocida. Esto se justifica dada la casi nula variación en la frecuencia de línea por parte del proveedor del servicio eléctrico. De hecho, esta variación está limitada a un máximo del 3%.
- 5) *Suposición de desacoplo*: Se puede observar que para grandes valores del capacitor  $C$ , la dinámica del voltaje de éste, i.e., ecuación (1.5), es más lenta que las dinámicas del filtro de alisamiento (corriente de inductor  $i_r$  y voltaje de capacitor  $V_r$ ). Esto permite dividir el estudio del sistema en dos partes. La primera, compuesta por las dinámicas del inductor  $L_r$  y el capacitor  $C_r$ , y la segunda, asociada con el capacitor  $C$ . En el proceso de diseño se referirá a estas como lazo interno de control y lazo externo de control, respectivamente.





## CAPITULO 2

---

### DISEÑO Y DISCRETIZACION DEL CONTROLADOR

*En esta sección, tomando como base el modelo matemático obtenido del DVR y las suposiciones del capítulo anterior, se diseña el controlador, y se realiza la discretización del mismo a fin de implementarlo en un Procesador Digital de Señales. El diseño del controlador es dividido en dos etapas: el lazo de control interno y el externo. El primero está destinado a cumplir el objetivo de seguimiento en el voltaje, mientras que el segundo está encargado de la regulación del voltaje del lado de continua.*

#### 2.1. Lazo interno

Manipulando las ecuaciones (1.1) y (1.2), se puede escribir la dinámica de  $V_r$  como el sistema de segundo orden siguiente,

$$L_r C_r \ddot{V}_r = L_r a \dot{i}_0 + e - V_r \quad (2.1)$$

donde  $e$  entra como la señal de control y se define  $\dot{i}_0$  como la derivada temporal de  $i_0$  (se usa esta notación para evitar conflictos entre el punto de la  $i_0$  y el punto de la derivada).

### 2.1.1 Caso parámetros conocidos.

Se plantea primero el caso de parámetros conocidos a fin de motivar la estructura de esta etapa del controlador. En este caso, observando la forma de la expresión (2.1), se propone cancelar el término  $V_r$ , evaluar el resto de la expresión en  $V_r = V_r^*$ , resolver para  $e$  y añadir términos de amortiguamiento, con esto se obtiene la siguiente ley de control:

$$e = V_r - L_r a \hat{i}_0 + L_r C_r \ddot{V}_r^* - k_2 \dot{\tilde{V}}_r - k_1 \tilde{V}_r \quad (2.2)$$

donde se ha definido  $\tilde{V}_r \triangleq V_r - V_r^*$  y así  $\ddot{\tilde{V}}_r = \ddot{V}_r - \ddot{V}_r^*$ , con  $k_1$  y  $k_2$  parámetros de diseño.

El sistema en lazo cerrado da como resultado el modelo de error siguiente

$$L_r C_r \ddot{\tilde{V}}_r + k_2 \dot{\tilde{V}}_r + k_1 \tilde{V}_r = 0 \quad (2.3)$$

Notar que se trata de un sistema LTI cuyo punto de equilibrio es el origen, el cual es exponencialmente estable, siempre y cuando, los parámetros de diseño  $k_1$  y  $k_2$  sean seleccionados como constantes positivas.

Notar que se han incluido los términos  $(-k_2 \dot{\tilde{V}}_r - k_1 \tilde{V}_r)$  los cuales permiten añadir amortiguamiento al sistema, y así garantizar la estabilidad.

### 2.1.2 Caso parámetros desconocidos:

Motivados por la estructura del controlador (2.2), se propone el siguiente control para el caso de parámetros desconocidos:

$$e = -k_1 \tilde{V}_r - k_2 \dot{\tilde{V}}_r + V_r + \underbrace{k_2 \dot{\hat{V}}_r^* - a \hat{L}_r \hat{i}_0 + \hat{L}_r \hat{C}_r \ddot{\hat{V}}_r^*}_{desconocido} \quad (2.4)$$

donde se usa ( $\hat{\cdot}$ ) para representar a los estimados. Observar que los tres últimos términos de (2.4) son perturbaciones periódicas, por lo que pueden ser expresadas en forma de sumatorias y concentradas en un solo término como a continuación

$$e = -k_1 \tilde{V}_r - k_2 \dot{\tilde{V}}_r + V_r - \sum_k \rho_k^T \hat{\phi}_k \quad (2.5)$$

donde se ha definido

$$\sum_k \rho_k^T \hat{\phi}_k \triangleq -k_2 \dot{\hat{V}}_r^* + a \hat{L}_r \hat{i}_0 - \hat{L}_r \hat{C}_r \ddot{\hat{V}}_r^* \quad (2.6)$$

Escribiendo la expresión (2.1) en términos del error  $\tilde{V}_r \triangleq V_r - V_r^*$  se obtiene:

$$L_r C_r \ddot{\tilde{V}}_r = -k_2 \dot{\tilde{V}}_r + a L_r \hat{i}_0 - L_r C_r \dot{V}_r^* + e - V_r + k_2 \dot{V}_r^* \quad (2.7)$$

la cual también puede escribirse como

$$L_r C_r \ddot{\tilde{V}}_r = \sum_k \rho_k^T \phi_k + e - V_r + k_2 \dot{V}_r^* \quad (2.8)$$

donde se ha definido  $\sum_k \rho_k^T \phi_k \triangleq -k_2 \dot{V}_r^* + a L_r \hat{i}_0 - L_r C_r \dot{V}_r^*$

El sistema en lazo cerrado da el siguiente modelo de error

$$L_r C_r \ddot{\tilde{V}}_r = -k_1 \tilde{V}_r - k_2 \dot{\tilde{V}}_r + \sum_k \rho_k^T \tilde{\phi}_k \quad (2.9)$$

Notar que se trata de un sistema LTI perturbado por una señal periódica desconocida. Ahora, siguiendo el método de Lyapunov para encontrar las leyes de adaptación, se propone la siguiente función cuadrática en término de los errores de voltaje  $\tilde{V}_r$  y su derivada  $\dot{\tilde{V}}_r$ , así como de los errores paramétricos  $\tilde{\phi}_k$ ,  $\forall k \in H$ .

$$W = \frac{L_r C_r}{2} \dot{\tilde{V}}_r^2 + \frac{k_1}{2} \tilde{V}_r^2 + \sum_k \frac{1}{2\gamma_k} \tilde{\phi}_k^T \tilde{\phi}_k \quad (2.10)$$

Su derivada con respecto al tiempo a lo largo de la ecuación del error (2.9) resulta:

$$\dot{W} = -k_2 \dot{\tilde{V}}_r^2 + \dot{\tilde{V}}_r \sum_k \rho_k^T \tilde{\phi}_k + \sum_k \frac{1}{\gamma_k} \dot{\tilde{\phi}}_k^T \tilde{\phi}_k \quad (2.11)$$

Para cancelar los términos asociados a los errores paramétricos se proponen las siguientes leyes de adaptación

$$\dot{\tilde{\phi}}_k^T = -\gamma_k \dot{\tilde{V}}_r \rho_k^T, \quad \forall k \in H \quad (2.12)$$

donde  $\gamma_k$ ,  $\forall k \in H$  son parámetros de diseño, que toman valores constantes positivos.

Dado que  $\phi_k$  es supuesta constante, entonces  $\dot{\tilde{\phi}}_k = \dot{\hat{\phi}}_k$  de donde se obtiene

$$\dot{\hat{\phi}}_k = -\gamma_k \dot{\tilde{V}}_r \rho_k, \quad \forall k \in H \quad (2.13)$$

Con esta ley de adaptación la derivada temporal de la función de almacenamiento  $W$  a lo largo de la trayectoria de  $\tilde{V}_r$  (2.9) se reduce a

$$\dot{W} = -k_2 \dot{\tilde{V}}_r^2 \quad (2.14)$$

De la ecuación (2.14), se obtiene la condición  $k_2 > 0$  para que  $\dot{W}$  sea negativa semidefinida. Esto, sin embargo, no garantiza la convergencia del modelo de error hacia el punto de equilibrio dado por  $\begin{bmatrix} \tilde{V}_r, \dot{\tilde{V}}_r, \tilde{\phi}_k \end{bmatrix} = [0, 0, 0]$ . Para completar la prueba se usa el Teorema de Ljapunov [14]. En este caso,

$$\dot{W} \equiv 0 \Leftrightarrow \dot{\tilde{V}}_r \equiv 0 \quad (2.15)$$

Lo cual implica que  $\ddot{\tilde{V}}_r = 0$ , de aquí se observa que los puntos de equilibrio  $\tilde{V}_r$  y  $\tilde{\phi}_k$  son constantes. Aplicando lo anterior a la ecuación (2.9) se tiene que

$$k_1 \tilde{V}_r = \sum_k \rho_k^T \tilde{\phi}_k \quad (2.16)$$

Como  $k_1 \tilde{V}_r$  es una constante y  $\sum_k \rho_k^T \tilde{\phi}_k$  es una señal variante en el tiempo periódica, entonces el único punto en el cual es válida esta igualdad es en el punto de equilibrio  $\tilde{\phi}_k = 0$  y  $\tilde{V}_r = 0$  que es el máximo invariante. Por lo tanto el punto de equilibrio  $\begin{bmatrix} \tilde{V}_r, \dot{\tilde{V}}_r, \tilde{\phi}_k \end{bmatrix} = [0, 0, 0]$  es asintóticamente estable  $\forall k \in H$ .

El problema de seguimiento del voltaje  $V_r$  hacia su referencia queda entonces resuelto por el controlador formado por las ecuaciones (2.5) y (2.13), más aún, se garantiza la convergencia paramétrica.

Se nota, sin embargo, que la ecuación (2.13) es difícil de implementar a causa del vector  $\rho_k$  cuyas componentes son señales periódicas del tipo senoidal y cuyo cálculo es complejo a nivel de implementación. Para reducir la complejidad de este controlador se definen las siguientes transformaciones:

$$\psi_k^r \triangleq \rho_k^T \hat{\phi}_k \quad (2.17)$$

$$\psi_k^i \triangleq \rho_k^T J \hat{\phi}_k \quad (2.18)$$

donde la matriz  $J \triangleq \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$  es una matriz antisimétrica que ya había sido definida en el capítulo anterior.

Derivando las ecuaciones (2.17) y (2.18) con respecto al tiempo se obtienen las siguientes expresiones

$$\dot{\psi}_k^r = \rho_k^T \dot{\hat{\phi}}_k + \dot{\rho}_k^T \hat{\phi}_k = -\gamma_k \ddot{V}_r \rho_k^T \rho_k + k\omega (J\rho)^T \hat{\phi}_k$$

$$\dot{\psi}_k^i = \rho_k^T J \dot{\hat{\phi}}_k + \dot{\rho}_k^T J \hat{\phi}_k = -\gamma_k \ddot{V}_r \rho_k^T J \rho_k + k\omega (J\rho)^T J \hat{\phi}_k, \quad \rho_k^T J \rho_k = 0$$

Donde se obtienen las siguientes expresiones, las cuales forman un sistema de orden 2, de hecho un oscilador armónico, con la entrada  $\ddot{V}_r$  y la salida de interés  $\psi_k^r$

$$\begin{aligned} \dot{\psi}_k^r &= -k\omega \psi_k^i - \gamma_k \ddot{V}_r \\ \dot{\psi}_k^i &= k\omega \psi_k^r \end{aligned} \quad (2.19)$$

Notar que se trata de un sistema LTI que puede ser escrito en espacio de estados de la manera siguiente:

$$\begin{bmatrix} \dot{\psi}_k^r \\ \dot{\psi}_k^i \end{bmatrix} = \begin{bmatrix} 0 & -k\omega \\ k\omega & 0 \end{bmatrix} \begin{bmatrix} \psi_k^r \\ \psi_k^i \end{bmatrix} + \begin{bmatrix} -\gamma_k \\ 0 \end{bmatrix} \ddot{V}_r \quad (2.20)$$

Expresando la ecuación (2.20) en forma de función de transferencia, considerando que la transformada de Laplace de  $\ddot{V}_r$  está dada por  $s\tilde{V}_r(s)$ , se obtiene la siguiente expresión

$$\psi_k^r(s) = \frac{-\gamma_k s^2}{s^2 + k^2 \omega^2} \tilde{V}_r(s) \quad (2.21)$$

Más aún, esta ecuación también se puede escribir de la siguiente forma:

$$\psi_k^r(s) = -\gamma_k \tilde{V}_r(s) + \frac{\gamma_k k^2 \omega^2}{s^2 + k^2 \omega^2} \tilde{V}_r(s), \quad \forall k \in H \quad (2.22)$$

Se puede observar que el segundo término de la derecha es un filtro paso-bajas de segundo orden en cuya respuesta se observa un pico de resonancia a la frecuencia  $\omega$ . Esta última observación es de gran utilidad al realizar la discretización del controlador.

A partir de las ecuaciones (2.5) y (2.22) queda la siguiente expresión para el lazo interno de control:

$$e = V_r - \left( k_1 + \sum_k \gamma_k \right) \tilde{V}_r - k_2 \dot{V}_r + \sum_k \varphi_k \quad (2.23)$$

donde  $\varphi_k$  esta determinado ahora por

$$\begin{aligned} \begin{bmatrix} \dot{\psi}_k^r \\ \dot{\psi}_k^i \end{bmatrix} &= \begin{bmatrix} 0 & -k\omega \\ k\omega & 0 \end{bmatrix} \begin{bmatrix} \psi_k^r \\ \psi_k^i \end{bmatrix} + \begin{bmatrix} -\gamma_k \\ 0 \end{bmatrix} \tilde{V}_r \\ \varphi_k &= [0 \quad 1] \begin{bmatrix} \psi_k^r \\ \psi_k^i \end{bmatrix} \end{aligned} \quad (2.24)$$

Notar que  $C_r \dot{V}_r = i_{Cr}$ , por lo tanto la ecuación (2.23) también se puede escribir de la siguiente manera:

$$e = V_r - \left( k_1 + \sum_k \gamma_k \right) \tilde{V}_r - k_2' i_{Cr} + \sum_k \varphi_k \quad (2.25)$$

donde  $k_2' = \frac{k_2}{C}$ ,  $k_2'$  es un nuevo parámetro de diseño.

De lo anterior puede apreciarse que no es necesario conocer o estimar la primera derivada del voltaje del capacitor  $\dot{V}_r$ , esta se sustituye por la medición de la corriente en el capacitor  $i_{Cr}$ . Claramente la corriente  $i_{Cr}$  representa una medida mas fiel de la derivada del voltaje  $\dot{V}_r$  en contraste con la implementación de un estimador para  $\dot{V}_r$ .

Otra observación interesante, en el diseño del lazo de control interno, es la reducción de las leyes adaptables a simples filtros resonantes. Esto ratifica el principio del modelo interno, así el banco de filtros resonantes compensa la perturbación periódica formada por componentes armónicos del voltaje de entrada.

## 2.2. Lazo externo

Primeramente se reescribe la dinámica de la ecuación (2.5) como:

$$C \dot{z} = e i_r - \frac{z}{2R} \quad (2.26)$$

donde se ha definido el cambio de variables  $z \triangleq \frac{V_c^2}{2}$ , con este cambio de variable, el objetivo de regulación del voltaje del capacitor es ahora  $z \triangleq \frac{V_c^2}{2} \rightarrow \frac{V_D^2}{2}$ .

Con base en la suposición de desacoplo, es decir, asumiendo que el objetivo de seguimiento del DVR ya se ha cumplido en un tiempo arbitrariamente pequeño, esto es,  $V_r \cong V_r^*$ ,  $\hat{\phi}_k \cong \phi_k$ . En esté caso

$$e \cong V_r^* - a L_r \dot{\hat{i}}_0 + L_r C_r \ddot{V}_r^* \quad (2.27)$$

$$i_r \cong a i_o - C_r \dot{V}_r^* \quad (2.28)$$

donde el control  $e$ , en este caso, corresponde al control equivalente que garantiza el seguimiento de voltaje.

Más aún, de la ecuación (1.4) se tiene que:

$$V_r^* = \frac{V_s - V_0^*}{a} = \frac{V_s - V_s^1 - \rho i_0^1}{a} \quad (2.29)$$

Sustituyendo el valor de  $V_r^*$  de (2.29) en la ecuación (2.25) se obtiene la siguiente expresión que ahora involucra a la variable  $\rho$ :

$$\begin{aligned} e i_r &= (a i_o - C_r \dot{V}_r^*) \left( V_r^* - a L_r \dot{\hat{i}}_0 + L_r C_r \ddot{V}_r^* \right) \\ &= (a i_o - C_r \dot{V}_r^*) \left( \frac{V_s - V_s^1 - \rho i_0^1}{a} - a L_r \dot{\hat{i}}_0 + L_r C_r \ddot{V}_r^* \right) \end{aligned} \quad (2.30)$$

Desarrollando la expresión (2.30) y reagrupando los términos en forma conveniente se obtiene:

$$e i_r = -\rho \left( i_0 i_0^1 - \frac{i_0^1}{a} C_r \dot{V}_r^* \right) + (a i_o - C_r \dot{V}_r^*) \left( \frac{V_s - V_s^1}{a} \right) + (a i_o - C_r \dot{V}_r^*) \left( -a L_r \dot{\hat{i}}_0 + L_r C_r \ddot{V}_r^* \right)$$

Notar que  $C_r \dot{V}_r^* = i_{Cr}^*$ :

$$e i_r = -\rho \left( i_0 i_0^1 - \frac{i_0^1}{a} i_{Cr}^* \right) + (a i_o - C_r \dot{V}_r^*) \left( \frac{V_s - V_s^1}{a} \right) +$$



$$+(ai_o - C_r \dot{V}_r^*) \left( -aL_r \dot{i}_0 + L_r C_r \ddot{V}_r^* \right) \quad (2.31)$$

Dado que en este lazo de control nos interesa regular únicamente el promedio del voltaje del capacitor, entonces se extrae la componente de continua de la expresión (2.26) de aquí se tiene

$$C \dot{z}_0 = \langle ei_r \rangle_0 - \frac{z_0}{2R} \quad (2.32)$$

donde  $\langle \cdot \rangle_0$  es usado para expresar la componente de continua de las señales, también referido como promediado de la señal, y  $z_0 = \langle z \rangle_0$ .

Extrayendo la componente de CD de la ecuación (2.30) se obtiene:

$$\langle ei_r \rangle_0 = -\rho C_1 + C_2 \quad (2.33)$$

donde  $C_2$  es una constante de signo desconocido, y es debida a la componente de directa de los dos últimos productos de (2.30). La constante  $C_1$  es considerada una constante positiva, siempre y cuando la corriente de carga sea suficientemente grande, tal que, la siguiente condición se cumpla:

$$\left( i_{0,RMS}^1 \right)^2 - \left\langle \frac{i_0^1 i_{Cr}^1}{a} \right\rangle_0 = C_1 > 0 \quad (2.34)$$

donde  $i_{0,RMS}^1$  representa el valor cuadrático medio (RMS) de la fundamental de la corriente de carga  $i_0^1$ .

La expresión (2.32) se puede escribir como:

$$C \dot{z}_0 = -\rho C_1 + C_2 - \frac{z_0}{2R} \quad (2.35)$$

Se Observa que se trata de un sistema LTI nuevamente, el cual esta perturbado por una constante, por lo tanto se propone calcular a la variable  $\rho$  usando el siguiente controlador proporcional integral (PI), donde el ancho de banda de la parte proporcional se limita por un LPF. La razón de usar un LPF es para limitar la reinyección de distorsión contenida en el voltaje del capacitor hacia el lazo interno.

$$\begin{aligned} \rho &= -k_p \chi - k_i \xi \\ \dot{\xi} &= \tilde{z}_0 \\ \tau \dot{\chi} &= -\chi + \tilde{z}_0 \end{aligned} \quad (2.36)$$

donde  $\tilde{z}_0 \triangleq z_0 - \frac{V_D^2}{2}$ ,  $k_i$  es la constante de integración,  $k_p$  es la constante de proporcionalidad y  $\tau$  es la constante de tiempo para el LPF.

En el dominio de la frecuencia este sistema se escribe de la siguiente manera:

$$\rho = -\frac{k_p}{\tau s + 1} \tilde{z}_0 - \frac{k_i}{s} \tilde{z}_0 \quad (2.37)$$

Basados en las propiedades de filtrado paso bajas del controlador PI propuesto se puede reemplazar  $\tilde{z}_0$  por  $\tilde{z}$  y entonces el controlador se calcula como

$$\rho = -\left(\frac{k_p}{\tau s + 1} + \frac{k_i}{s}\right) \tilde{z} \quad (2.38)$$

donde, como se había definido anteriormente,  $\tilde{z} = \left(\frac{V_C^2 - V_D^2}{2}\right)$ .

El diagrama a bloques del controlador completo, compuesto por las ecuaciones (2.24)-(2.25) y (2.38), se muestra en la Fig. 3.1.

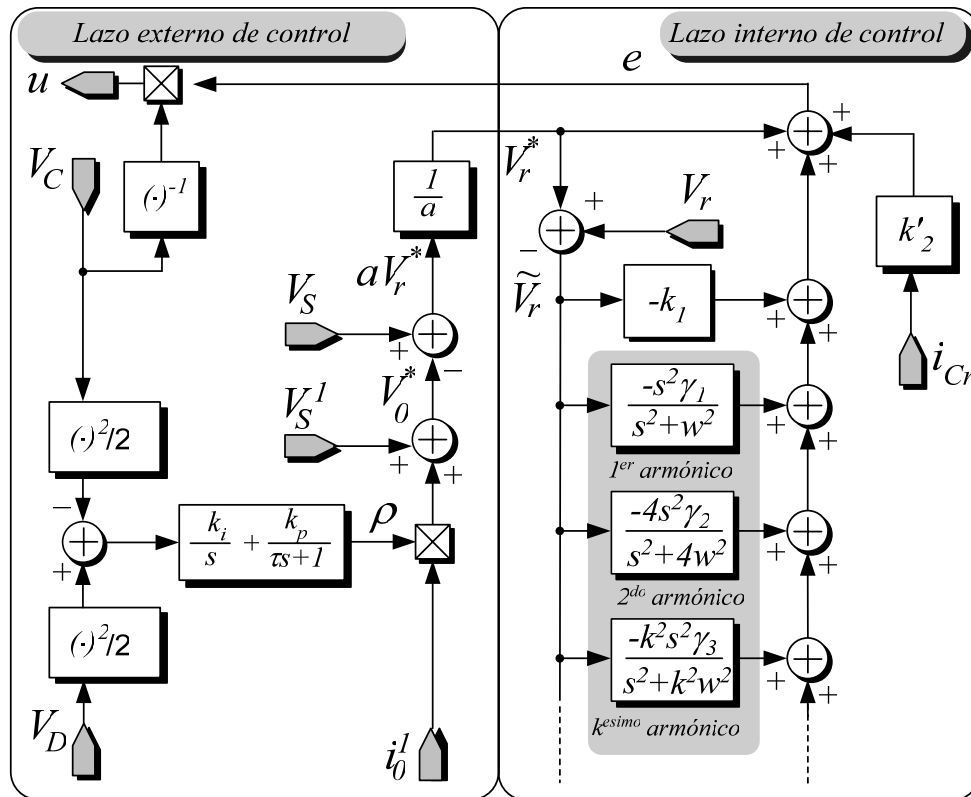


Fig. 3.1. Diagrama de bloques del controlador.

### 2.3. Discretización del controlador

A continuación se hacen algunas consideraciones con el objeto de facilitar la programación del código del controlador:

- a) La obtención de la fundamental de  $V_s$  e  $i_0$  se puede obtener por medio de un filtro pasa-bandas (BPF) sintonizado a la frecuencia de la fundamental  $\omega$ .

$$V_s^1 = \frac{A_{0,1}s\omega / Q_1}{s^2 + s\omega / Q_1 + \omega^2} V_s \quad (2.39)$$

$$i_0^1 = \frac{A_{0,2}s\omega / Q_2}{s^2 + s\omega / Q_2 + \omega^2} i_0 \quad (2.40)$$

donde  $Q_1$  y  $Q_2$  son los factores de calidad;  $A_{0,1}$ ,  $A_{0,2}$  las ganancias de cada filtro, y  $V_s^1$ ,  $i_0^1$  representan las componentes fundamentales del voltaje de línea y corriente de carga, respectivamente.

- b) En los filtros resonantes la ganancia es infinita a la  $k$ -ésima frecuencia de resonancia. Dada las limitaciones del formato de punto fijo, la situación anterior podría causar problemas, por este motivo se remplazan dichos filtros por BPF como se muestra a continuación. En los BPF se tiene control tanto en la ganancia como en el ancho de banda, esto es, se pueden especificar a un cierto factor de calidad y ganancia.

$$\varphi_k = \frac{\gamma_k k^2 \omega^2}{s^2 + k^2 \omega^2} \tilde{V}_r \rightarrow \varphi_k = \frac{\gamma_k k^2 \omega^2}{s^2 + \frac{k\omega s}{Q} + k^2 \omega^2} \tilde{V}_r \quad (2.41)$$

donde  $Q$  es el factor de calidad y  $\gamma_k$  la ganancia. El factor de calidad  $Q$  se selecciona lo más grande posible con el objeto de hacer más selectivo y de mayor ganancia al filtro. Existe sin embargo, un compromiso con la velocidad de respuesta del filtro.

Se puede notar que el filtro mostrado en (3.41) no es en realidad un BPF sino un LPF, el cual presenta un pico de resonancia cerca de la frecuencia de corte por lo que se referiría a ellos como LPF de segundo orden.

- c) Se usará la letra  $l$  como índice de la  $l$ -ésima muestra en la representación discreta de las variable, esto se hace para evitar la confusión con la  $k$  usada para referirse al  $k$ -ésimo armónico.

### 2.3.1. Conversión a espacio discreto

En la presente sección se muestran las conversiones de espacio continuo a espacio discreto de los términos de mayor dificultad en el controlador debido a que computacionalmente involucran más operaciones básicas. Entre estos destacan el integrador, el LPF, los BPFs y algunos LPFs de segundo orden, quienes efectivamente requieren más atención.

### 2.3.2. Discretización del integrador.

Usando la aproximación rectangular hacia adelante, un integrador multiplicado por una constante  $k_i$ , con entrada dada por  $\tilde{z}$  y salida  $m_2$ , como el de la expresión (2.38), se puede discretizar de la manera siguiente:

$$m_2 = \frac{k_i \tilde{z}}{s} \rightarrow m_2[l+1] = m_2[l] + k_i T \tilde{z}[l] \quad (2.42)$$

donde T es el periodo de muestreo.

En la presente aplicación  $T=50 \mu s$ , permitiendo una frecuencia de 20,000 muestras por segundo.

### 2.3.3. Discretización del LPF.

El LPF con constante proporcional  $k_p$ , constante de tiempo  $\tau$ , entrada dada por  $\tilde{z}$  y salida dada por  $m_3$ , usado en el lazo de control externo (ecuación (2.38)), esta dado por:

$$\dot{m}_3 = -\frac{1}{\tau} m_3 + \frac{k_p}{\tau} \tilde{z} \quad (2.43)$$

Discretizando lo anterior con el método de la matriz exponencial [22] se obtiene

$$m_3[l+1] = e^{-\frac{T}{\tau}} m_3[l] + k_p \left(1 - e^{-\frac{T}{\tau}}\right) \tilde{z}[l] \quad (2.44)$$

### 2.3.4. Discretización del BPF.

El BPF, por su capacidad selectiva, tiene la función de discriminar todas las frecuencias fuera de la frecuencia de sintonización. El espacio de estados de este filtro, usado en la extracción de las componentes fundamentales (ecuación (2.39) y (2.40)), es el siguiente:

$$\begin{aligned}\dot{\zeta} &= \begin{bmatrix} \frac{-\omega}{Q} & -\omega \\ \omega & 0 \end{bmatrix} \zeta + \begin{bmatrix} \frac{A_0 \omega}{Q} \\ 0 \end{bmatrix} v \\ y &= [1 \quad 0] \zeta\end{aligned}\quad (2.45)$$

donde la entrada  $v$  puede ser cualesquiera,  $V_s$  o  $i_0$ , mientras que la salida  $y$  puede ser  $V_s^1$  o  $i_0^1$ , respectivamente.

Su discretización usando el método de la matriz exponencial [22] es el siguiente:

$$\begin{aligned}\zeta[l+1] &= \hat{A}\zeta[l] + \hat{B}v[l] \\ y[l] &= [1 \quad 0]\zeta[l]\end{aligned}\quad (2.46)$$

donde las matrices están dadas por

$$\hat{A} = \begin{bmatrix} e^{-\frac{\omega T}{2Q}} \left( \cos\left(\frac{\omega T \sqrt{4Q^2 - 1}}{2Q}\right) - \frac{\text{sen}\left(\frac{\omega T \sqrt{4Q^2 - 1}}{2Q}\right)}{\sqrt{4Q^2 - 1}} \right) & \frac{-2Qe^{-\frac{\omega T}{2Q}} \text{sen}\left(\frac{\omega T \sqrt{4Q^2 - 1}}{2Q}\right)}{\sqrt{4Q^2 - 1}} \\ \frac{2Qe^{-\frac{\omega T}{2Q}} \text{sen}\left(\frac{\omega T \sqrt{4Q^2 - 1}}{2Q}\right)}{\sqrt{4Q^2 - 1}} & e^{-\frac{\omega T}{2Q}} \left( \cos\left(\frac{\omega T \sqrt{4Q^2 - 1}}{2Q}\right) + \frac{\text{sen}\left(\frac{\omega T \sqrt{4Q^2 - 1}}{2Q}\right)}{\sqrt{4Q^2 - 1}} \right) \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} \frac{2Ae^{-\frac{\omega T}{2Q}} \text{sen}\left(\frac{\omega T \sqrt{4Q^2 - 1}}{2Q}\right)}{\sqrt{4Q^2 - 1}} \\ \frac{A}{Q} \left( 1 - e^{-\frac{\omega T}{2Q}} \left( \cos\left(\frac{\omega T \sqrt{4Q^2 - 1}}{2Q}\right) + \frac{1}{\sqrt{4Q^2 - 1}} \text{sen}\left(\frac{\omega T \sqrt{4Q^2 - 1}}{2Q}\right) \right) \right) \end{bmatrix}$$

### 2.3.5. Discretización del LPF de segundo orden

El  $k$ -ésimo filtro, usado en la ecuación (2.41) y que substituye al  $k$ -ésimo filtro resonante, se expresa en espacio de estados de la siguiente manera

$$\dot{x}_k = \begin{bmatrix} 0 & -k\omega \\ k\omega & 0 \end{bmatrix} x_k + \begin{bmatrix} \frac{A_0 k\omega}{Q} \\ 0 \end{bmatrix} \tilde{V}_r \quad (2.47)$$

$$\varphi_k = [0 \quad k\omega] x_k$$

Su discretización por el método de la matriz exponencial [22] resulta en

$$x_k[l+1] = \hat{A}x_k[l] + \hat{B}\tilde{V}_r[l] \quad (2.48)$$

$$\varphi_k[l] = [0 \quad k\omega] x_k[l]$$

donde las matrices están dadas por

$$\hat{A} = \begin{bmatrix} e^{\frac{k\omega T}{2Q}} \left( \cos\left(\frac{k\omega T\sqrt{4Q^2-1}}{2Q}\right) - \frac{\text{sen}\left(\frac{k\omega T\sqrt{4Q^2-1}}{2Q}\right)}{\sqrt{4Q^2-1}} \right) & \frac{-2Qe^{-\frac{k\omega T}{2Q}} \text{sen}\left(\frac{k\omega T\sqrt{4Q^2-1}}{2Q}\right)}{\sqrt{4Q^2-1}} \\ \frac{2Qe^{\frac{k\omega T}{2Q}} \text{sen}\left(\frac{k\omega T\sqrt{4Q^2-1}}{2Q}\right)}{\sqrt{4Q^2-1}} & e^{\frac{k\omega T}{2Q}} \left( \cos\left(\frac{k\omega T\sqrt{4Q^2-1}}{2Q}\right) + \frac{\text{sen}\left(\frac{k\omega T\sqrt{4Q^2-1}}{2Q}\right)}{\sqrt{4Q^2-1}} \right) \end{bmatrix}$$

$$\hat{B} = \begin{bmatrix} \frac{2Ae^{-\frac{k\omega T}{2Q}} \text{sen}\left(\frac{k\omega T\sqrt{4Q^2-1}}{2Q}\right)}{\sqrt{4Q^2-1}} \\ \frac{A}{Q} \left( 1 - e^{-\frac{k\omega T}{2Q}} \left( \cos\left(\frac{k\omega T\sqrt{4Q^2-1}}{2Q}\right) + \frac{1}{\sqrt{4Q^2-1}} \text{sen}\left(\frac{k\omega T\sqrt{4Q^2-1}}{2Q}\right) \right) \right) \end{bmatrix}$$

En resumen el controlador, expresado en su forma discreta, está dado por las siguientes ecuaciones:

$$m_1[l] = \tilde{z}[l] = \frac{V_d^2 - V_C^2[l]}{2} \quad (2.49)$$

$$m_2[l+1] = m_2[l] + k_i T m_1[l] \quad (2.50)$$

$$m_3[l+1] = e^{-\frac{T}{\tau}} m_3[l] + k_p \left( 1 - e^{-\frac{T}{\tau}} \right) m_1[l] \quad (2.51)$$

$$m_4[l] = \rho[l] = m_3[l] + m_2[l] \quad (2.52)$$

$$m_5[l] = V_r^*[l] = \frac{1}{a} \left( V_s[l] - V_s^1[l] - m_4 i_0^1[l] \right) \quad (2.53)$$

$$m_6[l] = \tilde{V}_r[l] = V_r[l] - m_5[l] \quad (2.54)$$

$$e[l] = V_r[l] - \left( k_1 + \sum_k \gamma_k \right) m_6[l] + k_2 \left( i_r[l] - ai_0[l] \right) - \sum_k \varphi_k \quad (2.55)$$

$$u[l] = \frac{e[l]}{V_c[l]} \quad (2.56)$$

## CAPITULO 3

---

### EMULACION DEL LA PLANTA

*En este capítulo se describe la forma de simular la planta usando la tarjeta dSPACE DS1103. La planta está compuesta por el DVR, descrito en el Capítulo 1, la fuente de alimentación de forma distorsionada y la carga no lineal. Para simular la fuente de alimentación distorsionada se introducen algunos armónicos de orden superior, mientras que para la corriente de carga se considera un rectificador no controlado de diodos, alimentando a un capacitor y una resistencia*

#### **3.1. Descripción del sistema de emulación.**

Este capítulo se describe el uso de la tarjeta dSPACE DS1103 [15] para simular a la planta bajo control. Tradicionalmente esta tarjeta juega el papel de controlador, sin embargo, en este trabajo el uso de esta tarjeta se dedica a emular el sistema bajo control. Esto nos permite evaluar en forma confiable las leyes de control, implementadas directamente en un DSP, sin necesidad de arriesgar componentes en un prototipo real, reduciendo así el costo de la implementación. Más aún, las pruebas se realizan sin peligro alguno, pues las posibles altas tensiones y corrientes están representadas simplemente por señales de baja tensión, o bien como valores numéricos dentro de la tarjeta.

Las señales de interés, es decir,  $V_S$ ,  $i_0$ ,  $i_{Cr}$ ,  $V_r$ ,  $V_C$  y  $V_0$  que se generan dentro de la tarjeta dSPACE se hacen accesibles al exterior por medio de convertidores digital-analógico (DAC). Estas mismas señales son ingresadas, a través de convertidores



analógico-digitales (ADC), a la tarjeta de control DSP TMS320LF2407A donde son procesadas para generar la señal de control  $u$ . Cabe aclarar que en esta simulación se implementa el modelo promediado del DVR, es decir, que la planta emulada en la tarjeta dSPACE recibe como señal de control a la señal continua  $u$  proveniente de la tarjeta del DSP. Con esto se pretende que la prueba del controlador propuesto programado en el DSP sea lo más realista posible sin llegar a la construcción del prototipo. En la Fig. 3.1 se muestra el esquema del sistema implementado por la tarjeta dSPACE.

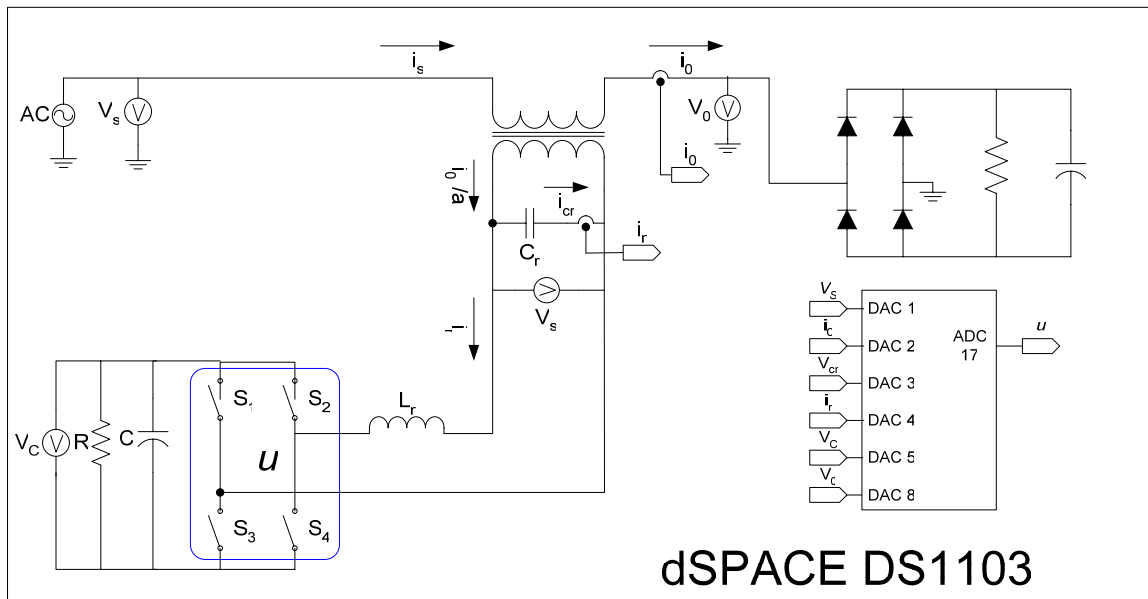


Fig. 3.1. La implementación del sistema a controlar en la tarjeta dSPACE.

### 3.2. Características de la tarjeta dSPACE.

En estos días, gracias al avance de la tecnología se ha hecho posible la existencia de herramientas de hardware especializados para la realización, en el menor tiempo posible, de pruebas de los sistemas y controladores por medio de una PC. Estas herramientas son de una ayuda invaluable ya que reducen el tiempo y costo cuando se trata de diseñar prototipos. Un ejemplo claro de estas herramientas son las tarjetas dSPACE. El costo de estas tarjetas es aún considerable, aunque, con el paso del tiempo, su costo se hace más accesible, además de hacerse mejoras en cuanto a su capacidad y velocidad, pudiendo así, manejar controladores cada vez más complejos.

La tarjeta dSPACE DS1103 toma tradicionalmente el papel del controlador del sistema, ya que ofrece las siguientes características:

- 20 canales de conversión analógica-digital (ADC) para la entrada de señales.
- 8 canales de conversión digital-analógica (DAC) para la salida de señales.
- Puerto digital de 32 bits con la posibilidad de configurar cada bit para ser usado bidireccionalmente.

- Salidas de señal en ancho de pulso (PWM) con frecuencias que van desde los 20 hasta los 100kHz.
- Una interfaz Simulink/MATLAB amigable al usuario.
- “dSPACE Control Desktop”, software que permite controlar todos los procesos de la tarjeta, facilita la depuración del proceso, la medición de señales en la propia pantalla de la PC, así como el manejo de parámetros internos en un ambiente grafico bastante amigable.
- El programa fuente se compila en la computadora, y el ejecutable se descarga en la memoria interna de la tarjeta. Esto hace más rápida la ejecución del programa del controlador.
- Manejo de múltiples usuarios para un misma tarjeta por medio de una red de área local (LAN).

Estas características combinadas permiten una fácil y versátil implementación del controlador.

### 3.3 Implementación del sistema en la tarjeta dSPACE.

La implementación del sistema, usando Simulink/MATLAB, tiene el aspecto (en pantalla) mostrado en la Fig 3.2.

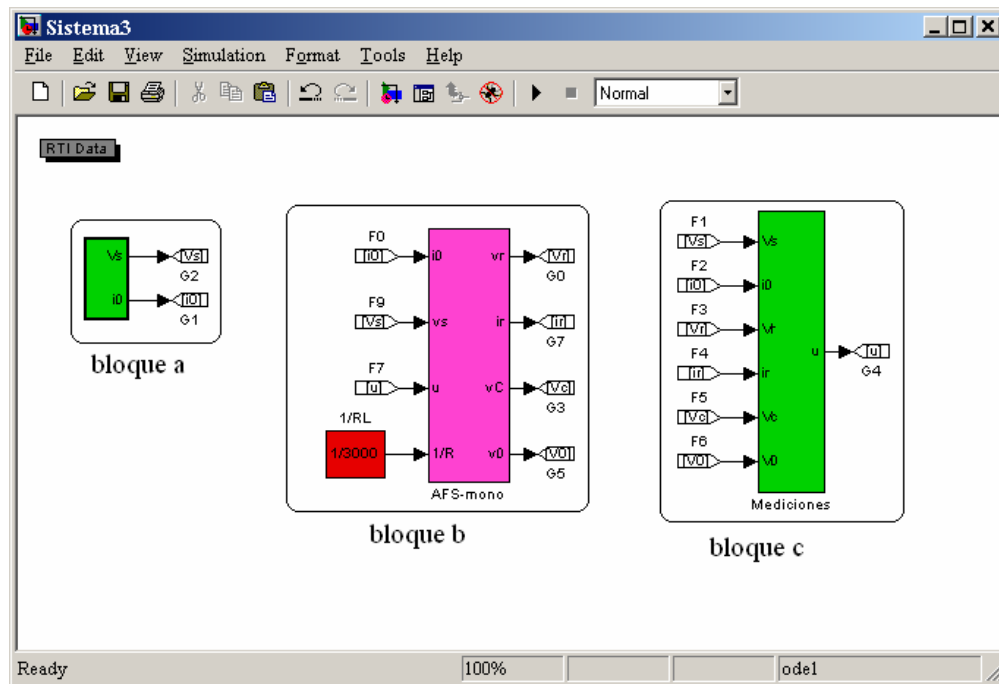


Fig 3.2. Despliegue de la estructura por bloques de la simulación de la planta en Simulink/MATLAB (archivo sistema3.mdl).

En la Fig. 3.2. se aprecian los diferentes bloques que componen al sistema. El “bloque a” contiene al rectificador de diodos no controlado que es el encargado de generar la

señal de tensión de alimentación a compensar  $V_S$ , y la señal de corriente  $i_0$  debida a la carga no lineal.

El “bloque b” contiene las ecuaciones diferenciales que describen el modelo de la planta (ver Capítulo 1). Finalmente, el “bloque c” contiene los bloques necesarios, básicamente convertidores ADC, para hacer accesibles las señales por medio de puertos. A continuación se describe más a detalle cada uno de estos bloques.

### 3.3.1 Bloque a. Implementación de un rectificador no controlado de diodos.

Este bloque permite simular la carga no lineal, y así inyectar al sistema las señales a compensar. Se emuló un rectificador de diodos no controlado cuyo esquema se encuentra en la Fig. 3.3. A este sistema se le alimenta una señal de voltaje distorsionada por armónicos de alto orden.

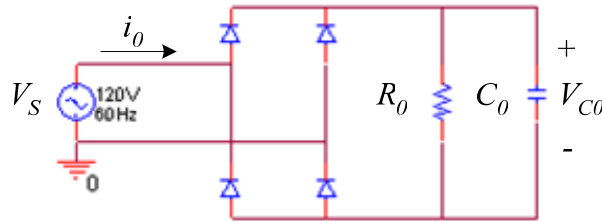


Fig 3.3. Diagrama esquemático de un rectificador de diodos no controlado.

El rectificador de diodos no controlado, dependiendo de las condiciones del voltaje en el capacitor ( $V_{C_0}$ ), conmuta entre dos dinámicas (modos), las cuales se describen a continuación:

#### *Modo 1*

La dinámica llamada modo 1 corresponde a la condición  $|V_S| < V_{C_0}$  en la cual se presenta la siguientes relaciones:

$$i_0 = 0$$

$$\dot{V}_{C_0} = \frac{-1}{R_0 C_0}$$

El sistema se mantiene en este modo mientras  $|V_S| > V_{C_0}$  al tiempo  $t = 1/\tau$ , donde  $\tau = R_0 C_0$

En el instante en que nuevamente ambos voltajes se igualan, i.e.,  $|V_S| = V_{C_0}$ , entonces la dinámica del sistema cambia al modo 2.

*Modo 2*

El sistema se mantiene en este modo mientras la corriente exigida sea mayor a cero, donde se cumple la siguiente relación.

$$i_0 = C_0 \dot{V}_s + \frac{V_s}{R_0}$$

$$V_{C_0} = |V_s|$$

Una vez que la corriente exigida por el rectificador no controlado de diodos cruza por cero, la dinámica del sistema se cambia al modo 1.

La curva típica generada por este sistema, se muestra en la Fig. 3.4.

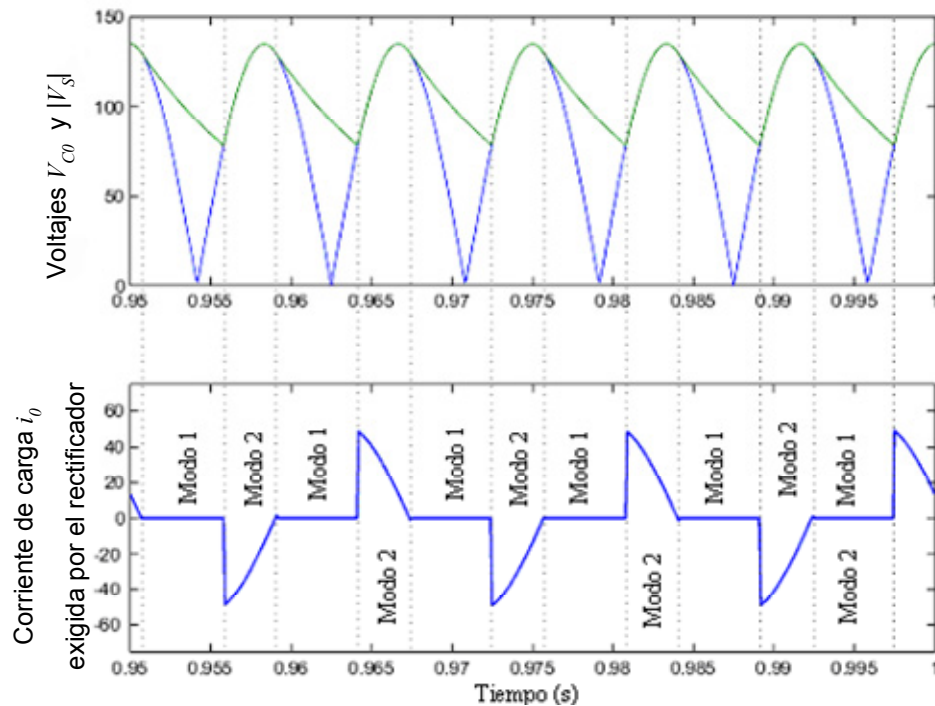


Fig 3.4. Respuesta típica de un rectificador no controlado de diodos.

De lo anterior se puede apreciar que el rectificador no controlado de diodos es una carga que exige una corriente altamente no lineal, cuyo espectro en la frecuencia esta formado por la componente fundamental y armónicos impares de alto orden.

La implementación en la dSPACE del sistema anterior se realiza con base en el entorno grafico del programa Simulink/Matlab como se muestra en la Fig. 3.5 donde se han tomado en cuenta las dinámicas y condiciones de conmutación entre uno y otro modo

descritas anteriormente. Este programa es ejecutado en línea con el resto de la dinámica de la planta.

Otra característica importante del rectificador de diodos no controlado es la posibilidad de cambiar la carga por medio del interruptor en el extremo superior izquierdo (*manual switch*) de la Fig. 3.5. Dependiendo de la posición del *manual switch* el valor de la resistencia se fija en  $10 \Omega$  ó  $50 \Omega$ . La posición del *manual switch* puede ser modificada en tiempo real, usando el paquete dSPACE-Control-Desk acompañando a dicha tarjeta.

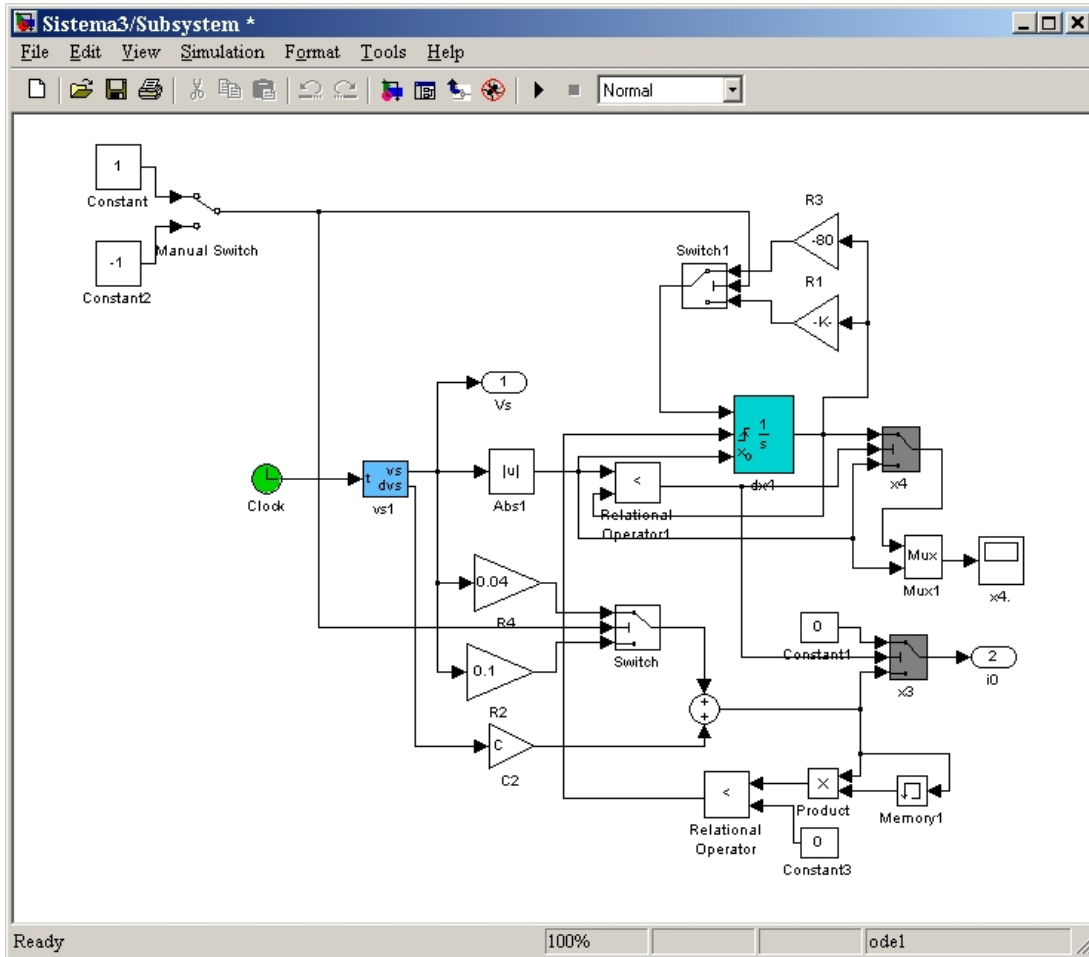


Fig 3.5. Implementación del rectificador no controlado en la dSPACE por medio de Simulink/MATLAB.

### 3.3.2 Bloque b. Realización de la planta.

La planta, contenida en el bloque b de la Fig. 3.2, que comprende las ecuaciones (1.1) a (1.4) se realiza por medio de la Fig. 3.6.

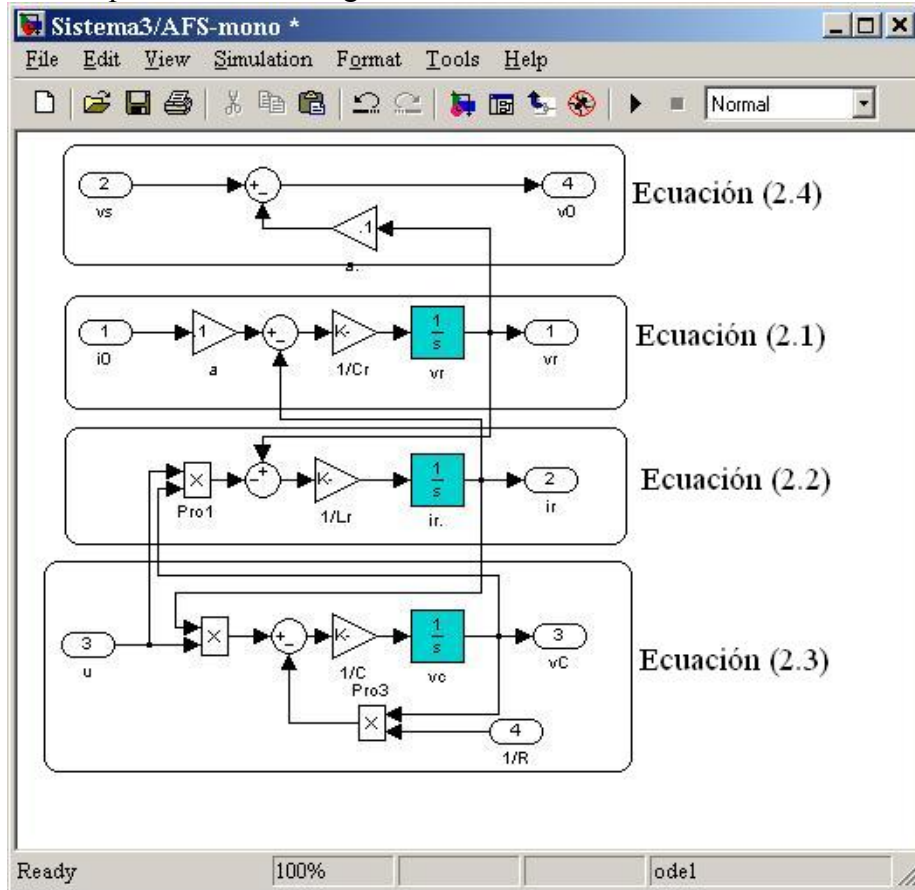


Fig 3.6. Implementación de la planta en la dSPACE

Se hace notar que las ecuaciones diferenciales que representan a la planta son fáciles de construir a modo de bloques. El control representado por la variable  $u$  entra al sistema por medio del puerto 3 de entrada y el único detalle a resaltar es que los bloques integradores deben tener condición inicial igual a cero.

### 3.3.3 Bloque c. Adquisición y muestra de las señales por parte de la tarjeta dSPACE.

La manera de comunicarse al exterior por parte de la tarjeta dSPACE de manera analógica es mediante puertos DACs y ADCs, los cuales son implementados a manera de bloques como se observa en la Fig. 3.7 con salida física a puertos.

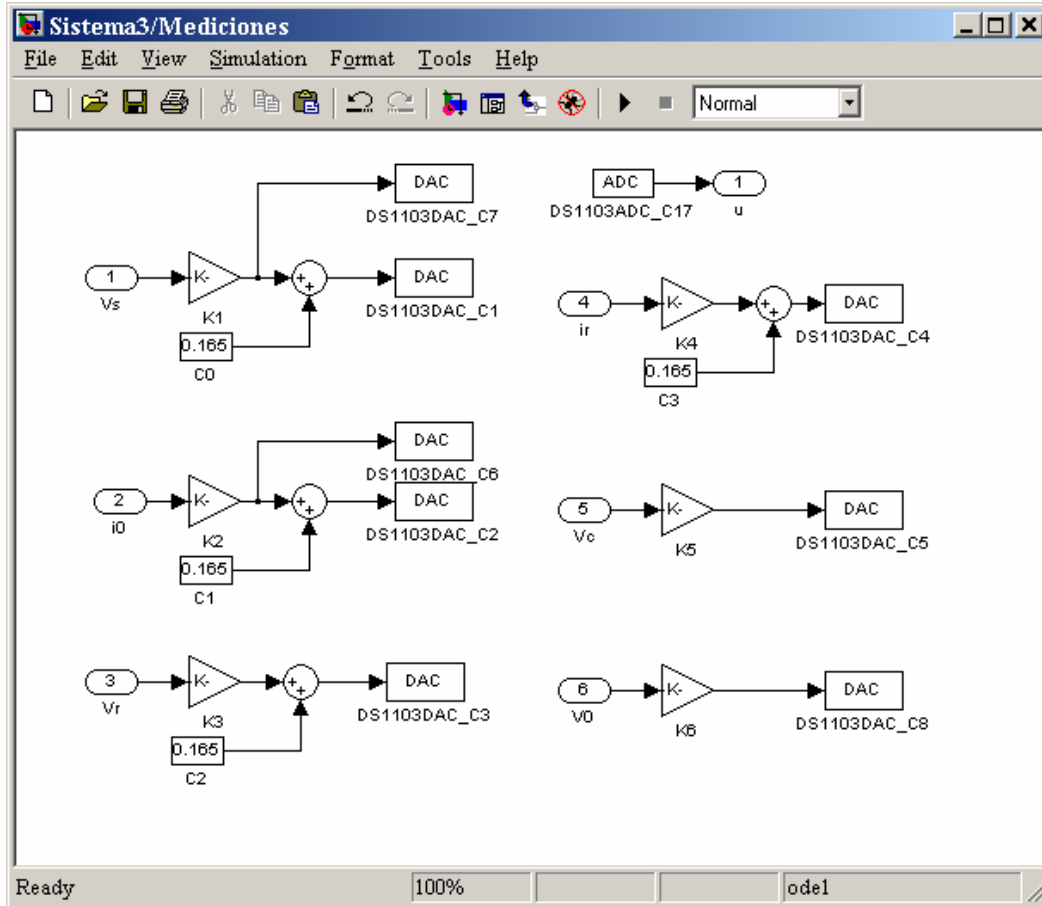


Fig 3.7. Diagrama de los bloques de los DAC y ADC

El voltaje a la entrada de los bloques del DAC se limita numéricamente a  $\pm 1$  para evitar errores de desborde, este valor se aumenta automáticamente a la salida de la interfase física 10 veces.

El valor de salida de los bloques del ADC se limita internamente a  $\pm 1$ , este valor es obtenido al dividir por 10 al voltaje en la entrada correspondiente al ADC, por consiguiente el valor máximo permitido a la entrada del ADC es  $\pm 10V$ . Un valor que sobrepase este rango puede ser potencialmente peligroso para la tarjeta dSPACE.

En este proyecto se requiere que las señales provenientes de la tarjeta dSPACE tengan un rango de 0 a 3.3V, que es el rango de voltaje permitido a la entrada de la tarjeta del DSP. Notar que las señales de entrada  $V_s$ ,  $i_0$ ,  $V_C$  y  $i_C$  son mediciones signadas, donde la mitad de este rango (1.65 V) representa al cero en dichas señales. En cambio la señal proveniente del voltaje del capacitor ( $V_C$ ) en la planta tiene signo exclusivamente positivo. El sistema de ganancias y desplazamientos del voltaje (offsets) requeridos para recuperar los valores reales de las señales al interior de la tarjeta dSPACE se muestran en la Tabla 3.1:

Tabla 3.1. Multiplicadores de la tarjeta

Canal de la tarjeta dSPACE	Señal	Escala máxima	Multiplicador	signado	Desplazamiento de voltaje (Offset)
1	$V_S$	$\pm 260.7$ V	$6.32901e-4$	Sí	1.65V
2	$i_0$	$\pm 77.08$ A	$2.140719e-3$	Sí	1.65V
3	$V_r$	$\pm 650$ V	$2.53846e-4$	Sí	1.65V
4	$i_{Cr}$	$\pm 77.08$ A	$2.140719e-3$	Sí	1.65V
5	$V_C$	491V	$6.720977e-4$	No	0

A fin de visualizar de las señales  $V_S$ ,  $i_0$  y  $V_0$  se implementaron convertidores DACs en la dSPACE cuyas salidas se conectaron directamente al osciloscopio.





## CAPITULO 4

---

### IMPLEMENTACION DEL CONTROLADOR EN EL DSP

*En este capítulo se describe en forma breve la implementación del controlador propuesto. Para ello se introduce la aritmética de punto fijo usada para implementar el algoritmo del controlador. También se describe la conversión de valores realizada en el convertidor analógico-digital y su representación final en formato de punto fijo. Además se presentan los resultados de la evaluación experimental de la ley de control. Esta filosofía de evaluación representa un paso anterior a la implementación real del sistema, y nos permite ahorrar costos, además de proveer seguridad en las pruebas, dado que los valores de voltaje y corriente son variables numéricas dentro de la tarjeta dSPACE.*

#### **4.1. Introducción.**

La programación del controlador se realizó en el entorno del programa de cómputo Code-Composer [18]. Este paquete, permite realizar las tareas siguientes:

1. Servir a manera de editor de notas utilizado especialmente para escribir código del DSP.
2. Generar un archivo en ensamblador de extensión (.asm),
3. Compilar el archivo de ensamblador para realizar el programa
4. Trasladar el código ejecutable hacia el DSP vía puerto paralelo.

En este paquete, los programas ejecutables se manejan como proyectos [19]. Un proyecto es una serie de archivos interconectados por el mismo programa, en el que se distingue un archivo ensamblador principal, el cual apunta a otros archivos de propósito general. Estos últimos pueden tener el contenido de los registros, los vectores de interrupción y más código, inclusive código preprogramado. Todos estos archivos se usan en conjunto para ensamblar el programa completo. Así, el controlador propuesto en la presente tesis es parte de un proyecto.

Existen dos puntos notables, en cuanto a depuración del programa se refiere. El primero consiste en la posibilidad de correr el programa en tiempo real con la frecuencia de reloj del sistema de 29.4912MHz, y al mismo tiempo observar y editar los registros. La otra es la posibilidad de usar macros. Una macro es un símbolo predefinido que se utiliza al interior del programa como sustituto de una secuencia de instrucciones. Su correcta utilización hace al programa muy flexible, ya que se pueden modificar parámetros dentro de la macro con base a los parámetros contenidos en el argumento.

## 4.2. Aritmética de punto fijo.

La aritmética de punto fijo es soportada por la mayoría de las aplicaciones que involucran DSPs, ya que para implementarla se requiere una arquitectura aritmética sencilla, lo cual baja notablemente el costo y el consumo de energía por parte del DSP.

Una de las grandes motivaciones de esta tesis es obtener un programa en tiempo real para la implementación del controlador discutido en el Capítulo 2 en un procesador aritmético de punto fijo. Esto, con el objeto de obtener el mayor poder de procesamiento posible.

Antes de resolver el problema de la implementación, debemos entender que un procesador de punto fijo a tiempo real, deteriora ligeramente, y por naturaleza, la respuesta del controlador, básicamente los problemas que deterioran esta respuesta son:

*Registros de tamaño finito:* Se impone un rango finito de los datos (parámetros y señales) dependiendo del máximo y mínimo número signado representable, también conocido como precisión finita. Cuando el dato cae afuera del rango disponible ocurre un sobreflujo. Cuando los datos son aproximados a uno de los valores representables, entonces un error de cuantización ocurre.

*Tiempos computacionales no cero:* Está dado por la demora de las operaciones computacionales. Como demora computacional se define el tiempo en que las señales se adquieren a la entrada del controlador, y el instante en que se actualiza la señal de salida del controlador.

Para aliviar el problema de los registros de tamaño finito se propone en esta tesis hacer que un número en punto fijo ocupe dos lugares en memoria. Gracias a esto se

reduce la posibilidad de sobreflujo, y por otro lado se minimizan los errores de cuantización.

Así, el formato propuesto, para representar números en punto fijo, ocupa dos lugares de memoria de 16 bits cada uno como se muestra en la Fig. 4.1:

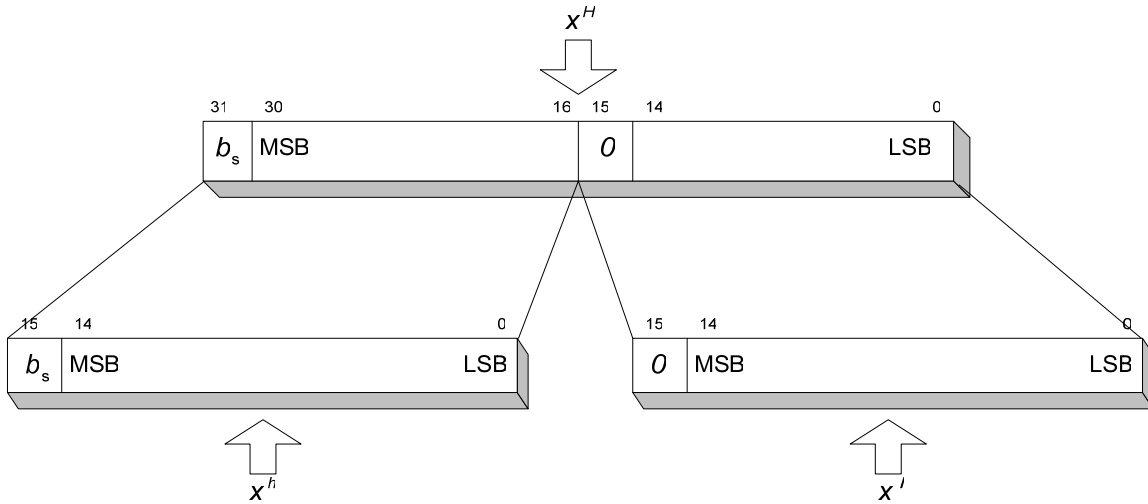


Fig 4.1. Lugares de memoria ocupados por un número en el punto fijo en el formato propuesto

En esta figura  $x^H$  representa el número signado en el formato propuesto de punto fijo. Este se divide en un registro más significativo ( $x^h$ ) que incluye un bit de signo ( $b_s$ ) y un registro menos significativo ( $x^l$ ).

El valor decimal  $x$  de un numero en  $x^H$ , representado como notación de complemento a dos usando 32 bits en formato referido en la literatura como  $Q_\alpha$ , es:

$$x = 2^\alpha \left( -b_s + \sum_{i=0}^{14} x_i^h 2^{i-15} + \sum_{j=0}^{14} x_j^l 2^{j-30} \right) \quad (4.1)$$

donde  $2^\alpha$  representa el máximo valor absoluto que se puede representar, y a  $\alpha$  se le conoce como la base o potencia de ese número. Normalmente la potencia en punto fijo es implícita. Notar que el registro menos significativo lleva un cero en el bit 15, por propósitos de compatibilidad de las operaciones de la DSP, las cuales se hacen comúnmente con registros de solo 16 bits.

El uso de 32 bits para realizar operaciones facilita el manejo de matrices que son usadas para calcular las respuestas de filtros paso-banda (BPF), y filtros paso-bajos de segundo orden (LPF), los cuales son computacionalmente inestables a bajas resoluciones. Así mismo, este formato se extendió al manejo de toda la aritmética del controlador.

### 4.3. Operaciones aritméticas con el formato de 32 bits.

Con el formato de punto fijo propuesto a 32 bits se expandieron y mejoraron operaciones como suma, resta, multiplicación, división a dicho formato. A continuación se dará una breve descripción de las operaciones anteriormente mencionadas.

#### Suma y Resta

Para interpretar los resultados del acumulador de 32 bits, así como para operar un número de punto fijo en el acumulador se procede de la siguiente manera. Se carga el registro de más alto orden de un operando en los bits de mayor orden del acumulador (HACC) y se carga el registro de menor orden en los bits de menor orden del acumulador (LACC), este último, rotado hacia la izquierda un lugar como se especifica en la figura

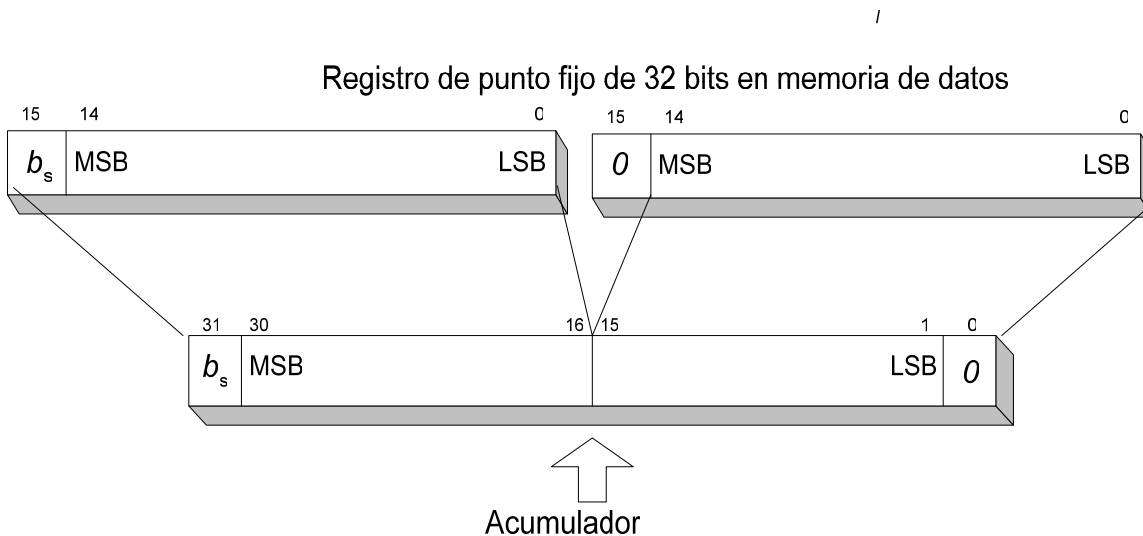


Fig 4.2. Operación de registros de 32 bits en el acumulador.

El segundo número de punto fijo se utiliza directamente en la operación de suma o resta por medio de un proceso similar, esto es, aparecen como parámetros en este mismo formato asociados a las instrucciones suma (ADD) o resta (SUB).

El resultado almacenado en el acumulador se guarda en 2 registros mediante el proceso inverso. Uno de estos registros representando la parte más significativa y el otro la menos significativa. Esto es, los bits más significativos del acumulador se guardan en el registro más significativo que representara al resultado en punto fijo. La parte baja del acumulador se recorre un lugar a la derecha y se guarda en el registro menos significativo que representa al resultado en punto fijo.

## Multiplicación

Una multiplicación de 32 bits por 32 bits genera un resultado de 64 bits, el cual para fines prácticos, se trunca a 32 bits. El resultado truncado se almacena en el acumulador para su posterior operación. Notar que los operandos de la multiplicación pueden ser signados.

En la Fig. 4.3, se muestra el proceso de la multiplicación para dos números en formato de punto fijo a 32 bits,  $x^H$  y  $y^H$  divididos a su vez en  $x^h$  y  $y^h$  representan a los registros más significativos y  $x^l$  y  $y^l$  a los registros menos significativos.

La multiplicación se hace por la suma del producto entrecruzado de  $x^h * y^l$  y  $x^l * y^h$ . Este producto se recorre 16 bits a la izquierda para sumarle  $x^h$  y  $y^h$ , obteniendo así el resultado de la multiplicación. Notar que nunca se realiza el producto entre los registros menos significativos, esto con el fin de agilizar la operación, aunque con un error despreciable en el resultado.

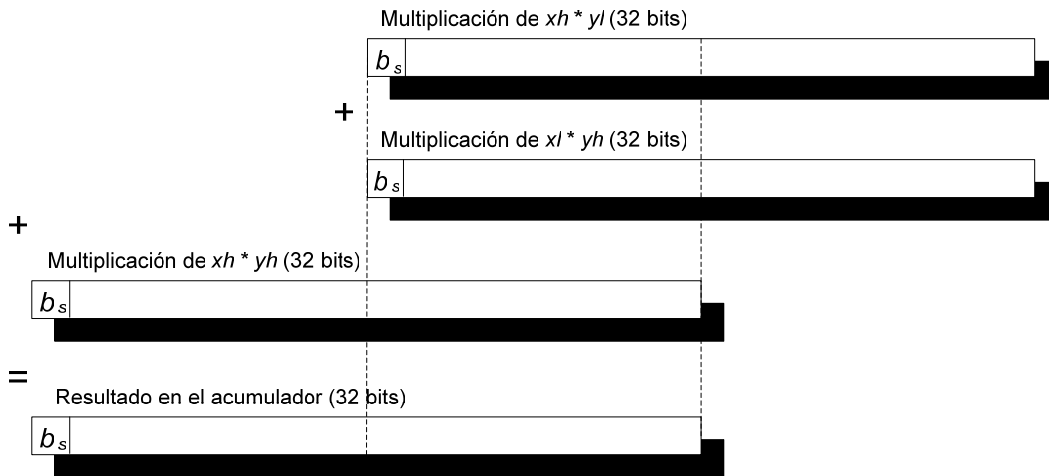


Fig. 4.3. Diagrama de bloques de la multiplicación de 32 bits.

Una multiplicación se ejecuta en el orden de medio microsegundo. Para almacenar el acumulador a registros se realiza el mismo proceso que en la suma o resta.

## División

El DSP efectúa, por medio de la instrucción resta condicionada (SUBC), una forma de división primitiva entre dos operandos positivos.

La división se realiza del modo siguiente:

- Se cargan la parte menos significativos del acumulador con el dividendo
- Con la instrucción repite (RPT), se cicla 16 veces la instrucción SUBC apuntando a la dirección del registro de datos donde se ubica el divisor.

-Al finalizar la división, el residuo queda en la parte alta del acumulador, mientras que en la parte baja del acumulador se guarda el resultado, esto es, el cociente.

La desventaja de la división antes mencionada es que tarda al menos un microsegundo, al ser al menos 16 instrucciones las que se ejecutan. Para obtener una división más exacta el proceso se repite por segunda vez.

#### 4.4. Conversión de los valores adquiridos por el ADC al formato de punto fijo.

El ADC tiene una resolución de 10 bits, que va desde el valor #0h (0V) hasta el valor #FFC0h (3.3V). Cuando se adquiere una señal vía el ADC se debe de tomar en cuenta que hay una ganancia afectando el valor de dicha señal que sirve para escalar su valor al rango permisible. Esta ganancia debe ser invertida y aplicada a la señal para reconvertirla su escala original dentro del DSP como se explica más adelante. Este proceso se muestra en la Fig. 4.4. Cabe señalar que en este proyecto hay 2 clases de señales, signadas y no signadas.

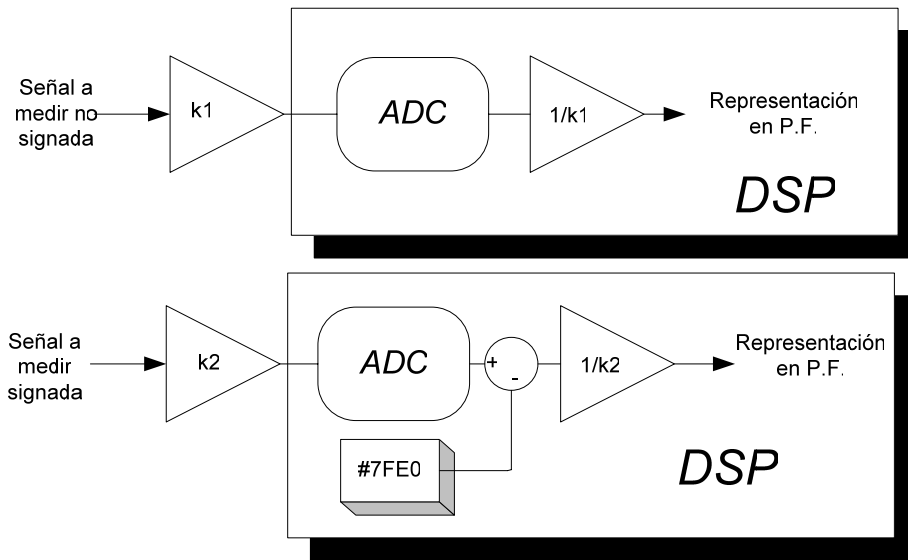


Fig. 4.4. Diagrama del proceso de adquisición de una señal analógica.

##### *Señales signadas.*

Pueden representar tanto a números positivos como negativos. En este caso, se considera como cero al valor medio de la escala completa de voltaje en el ADC, esto es, se toma 1.65 V como el cero de referencia. Así, internamente en el programa, y antes de proceder a transformar dicho valor a su formato en punto fijo se le resta el valor #07FE0h que corresponde al valor del cero de referencia.

Para obtener el inverso de la ganancia que se utilizó para reconvertir la señal de entrada a su escala real dentro del DSP se utiliza la siguiente formula:

$$\text{ganancia inversa} = \frac{\text{rango máximo medible} \left( \frac{2^{25}}{511} \right)}{Q_\alpha} \quad (4.2)$$

donde el *rango máximo medible*, es el valor máximo alcanzable por la medición, de forma tal que a la entrada del ADC se tenga un valor de 3.3V;  $Q_\alpha$  es valor máximo representable en punto fijo de 32 bits. Para fines prácticos este número debe ser potencia de 2 y mayor que el *rango máximo medible*.

*Señales no signadas.*

Representan números positivos exclusivamente.

Para obtener el inverso de la ganancia utilizado para reconvertir la señal de entrada a su escala real en el interior del DSP se utiliza la siguiente formula:

$$\text{ganacia inversa} = \frac{\text{rango maximo medible} \left( \frac{2^{25}}{1023} \right)}{Q_\alpha} \quad (4.3)$$

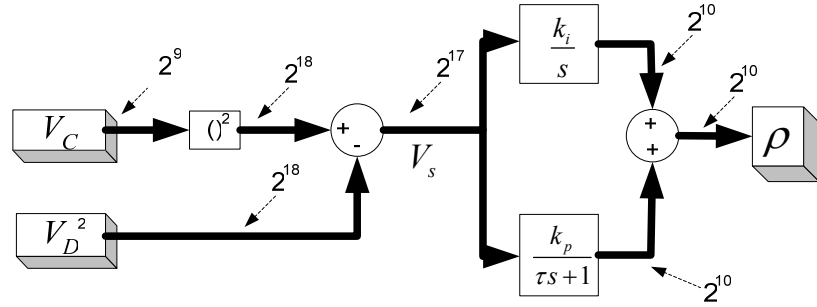
Con las mismas definiciones que en el caso anterior. El definir a conversión por separado para las señales no signadas se hace con el objeto de que las señales representadas al interior del DSP en punto fijo tengan el menor error posible de cuantización.

#### 4.5. Implementación por bloques del algoritmo de control en el DSP

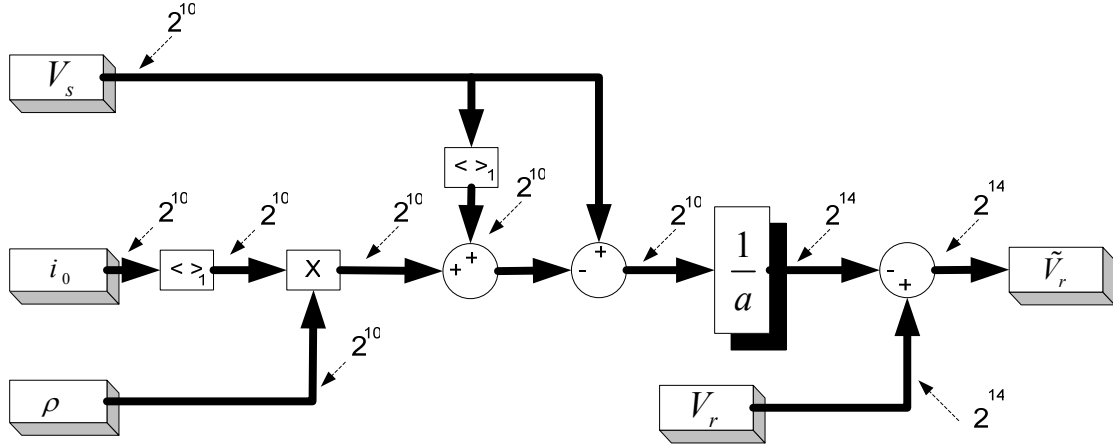
Como se mencionó anteriormente  $2^\alpha$  es el mayor número representable en el formato  $Q_\alpha$ . Por lo que creemos que es conveniente usar diferentes formatos para las diferentes partes del controlador a fin de reducir errores de cuantización. En otras palabras, es conveniente hacer el valor de  $Q_\alpha$  dinámico en cada parte del controlador. Para asignar el formato  $Q_\alpha$  a cada sección de operaciones del controlador se recurrió a simulaciones hechas en el programa Simulink/Matlab, donde se observaron los valores máximos alcanzables de los productos intermedios.

En las Fig 4.5 se pueden observar con más detalle las potencias propuestas al formato  $Q_\alpha$  en cada sección del controlador. En estas figuras los bloques resaltados representan números en formato de punto fijo  $Q_\alpha$  y cuyo valor máximo alcanzable es  $2^\alpha$ , mientras que las figuras en el plano representan simples operaciones.

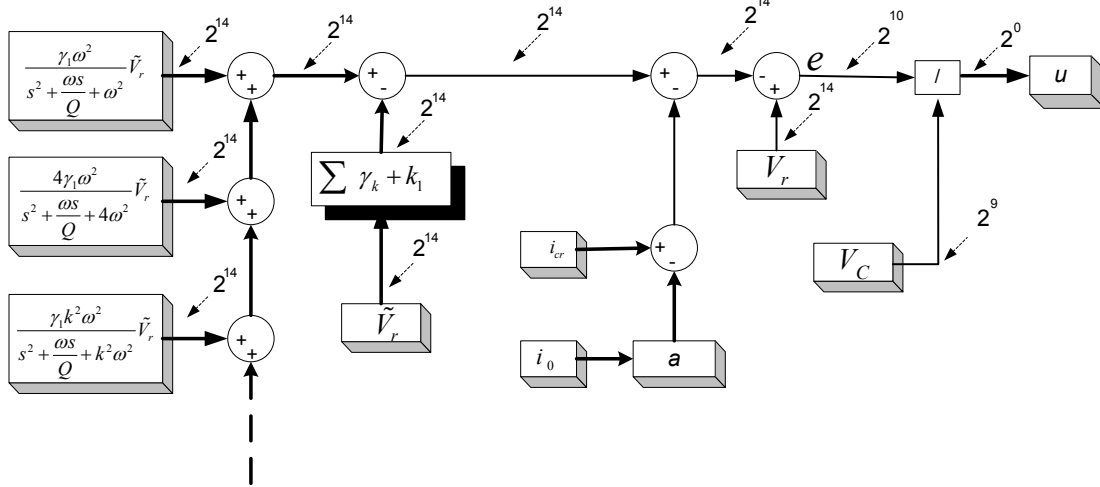




(a) Lazo externo: estimación de  $\rho$ .



(b) Lazo interno: cálculo de  $\tilde{V}_r$ .



(c) Banco de filtro resonantes y cálculo de la señal de control  $u$

Fig. 4.5. Potencias para los formatos de punto fijo propuestas para las operaciones en el controlador.

## 4.6. Descripción del código del programa principal en el DSP

A continuación se describe el código del programa principal que ejecuta el controlador propuesto. El programa principal forma parte de un proyecto, el cual hace referencia a otros archivos auxiliares usados para la construcción del programa ejecutable. Algunas de las macros a las que se hace referencia en este programa principal aparecen en los archivos `macrobib.asm` y `ecu.asm` descritas en el Apéndice C.

El programa principal se compone de los siguientes puntos:

- Inicialización de los dispositivos periféricos internos usados
- Existe un lazo infinito ficticio cuya función es esperar una interrupción del temporizador 2.
- La dirección donde se ubica el código del controlador esta contenida en un vector de interrupción.
- Las dos condiciones anteriores se repiten continuamente hasta el instante en que una interrupción por el pin de interrupción de seguridad es hecha, en este momento se deshabilitan los módulos al interior del DSP.

El encabezado lleva el comando `.title` que contiene el nombre del programa.

```
.title "Michael AFS"      ;Titulo del programa y Autor del mismo
```

En seguida aparecen los comandos `.include` que representan a los archivos a incluir en el proyecto, en este caso, los vectores de interrupción (`vectors.asm`), inicialización (`macrobib.asm`), ecuaciones del controlador (`ecu.asm`) y los registros tanto de la memoria del programa como de RAM (`Registros.asm`).

```
.include 240x.h           ;Archivo con nemonicos
.include vectors.asm     ;Direcciones de los vectores de interrupción
.include macrobib.asm    ;Inicialización de dispositivos periféricos
.include ecu.asm         ;Ecuaciones del controlador
.include Registros.asm   ;Registros del sector de programa y RAM
```

Una de las características peculiares del DSP es que la dirección `#0h` se utiliza como un vector de interrupción, a diferencia de la mayoría de los dispositivos digitales que usan la misma dirección solo como inicialización. Esto permite en la mayoría de los casos determinar si el causante de la reinicialización del sistema fue el guardián (WT), o una caída en el voltaje de alimentación entre otras razones.

```
.def      _c_int0
Start    B    _c_int0      ;La direccion donde se comienza
```

Por medio del comando `.text` se especifica el inicio de compilación del programa.

```
.text
_c_int0:      ;
start:       ;Inicio del programa principal
```

La siguiente línea representa a la macro de inicialización del sistema (mas detalles sobre esta macro en el Apéndice C)

```
Init                ;Inicialización de los registros del DSP
```

Este DSP es sumamente versátil en cuanto a la conexión de dispositivos periféricos, tanto en hardware como la configuración de tiempos de espera (siguientes 2 líneas). En este caso se le especifica al DSP que evite el uso de tiempos de espera.

```
SPLK    #0h,MTRX    ;Se establece el tiempo de espera para
OUT     MTRX,WSGR   ;dispositivos periféricos como el mínimo
```

Es conveniente que el ADC se inicialice para que adquiera de manera ininterumpida las señales  $V_c$ ,  $V_s$ ,  $i_{cr}$ ,  $V_r$  e  $i_0$  sin intervención del programa, para ello se inserta la macro ADCreg.

```
ADCreg            ;Inicializa el ADC
```

La macro Timereg tiene como función principal establecer el tiempo de muestreo del controlador a  $50\mu s$ , para establecer una frecuencia de trabajo de 20kHz.

```
Timereg          ;Inicializa el Timer
```

En este momento se dirige el apuntador de memoria RAM hacia los registros de trabajo

```
LDP     #6h        ;Pagina de los registros de trabajo
```

Una vez encendido el DSP, los registros de la memoria RAM quedan con valores totalmente aleatorios, es importante “limpiarlos” depositando en cada una de las 256 primeras locaciones de los registros de trabajo un cero.

```
Clrblk            ;Limpia desde 300 hasta 3FF
```

Con la siguiente macro se trasladan constantes de memoria de programa hacia la memoria RAM con el fin de reducir los tiempos de algunos cálculos como ya se explicó anteriormente.

```
MKPD             ;Traslada las constantes de PMA a DMA
```

Después de la reinicialización del DSP, los puertos digitales quedan como entradas de datos. La macro Puertos configura los puertos digitales como salidas y los pone a estado bajo.

```
Puertos          ;Inicializa los puertos
```

El modo de desbordamiento debe ser habilitado, a fin de poder manejar el desbordamiento en las operaciones.

```
SETC    OVM                ;Overflow habilitado
```

El modo de corrimiento de producto (SPM) evita el uso de 2 instrucciones para hacer un corrimiento a la izquierda en el resultado de las multiplicaciones, y así escalar el resultado.

```
SPM     #1                 ;Recorre a la izq un lugar los resultados de PREG
```

En la siguiente línea se establece un lazo infinito, a la espera de la interrupción originada por el tiempo de muestreo implementado en el temporizador 2. Notar que el perro guardián (WT) se deshabilita antes de este paso, para evitar que este reinicialice todo el sistema:

```
ende:   B                 ende    ;Solamente se espera a una interrupción
```

Las siguientes líneas muestran como la rutina de interrupción del temporizador 2 se usa para implementar el tiempo de muestreo, reconociendo el tipo de interrupción y dirigiendo al proceso a la subrutina de la ecuación del controlador.

```
ser3:   ;Servicio de interrupción del nivel 3
NOP                                           ;Espera un ciclo
PUERTO  0FFFFh,PBDATDIR ;El puerto B esta en alto
LDP     #PIVR >> 7h   ;Dirige a la pagina de las interrupciones
LACL    PIVR          ;Se obtiene la causa de la interrupción
XOR     #002Dh        ;compara si fue el desbordamiento del Timer 2
BCND    SIS2D,eq      ;si fue afirmativo dirígete a la rutina de Int
RET                                           ;caso contrario, regresa a ende
```

```
SIS2D: ;Causa de la interrupción: Desbordamiento del Timer 2
LDP     #GPTCONA >> 7h ;Pagina del EVM2
LACC    #0004h         ;Se responde afirmativamente al servicio
SACL    EVAIFRB        ;de interrupción del Timer 2
CLRC    INTM           ;Limpiando la llamada a interrupción
```

A continuación se presenta el algoritmo del controlador. Notar que esta rutina llama a otras macros (descritas en el Apéndice C) para completar el proceso.

```
TADC    Vs      ;
Con2m   Vc,i0,Vs,Vr,ir

Eqm1    ;Después  $(V_c^2 - V_d^2) / 2$   $(2^9) \rightarrow (2^{18}) \rightarrow (2^{17})$ 
Eqm2    ;Integra a  $V_z$  tilda  $(2^{17}) \rightarrow (1/s) \rightarrow (2^{10})$ 
Eqm3    ;LPF por  $V_z$  tilda  $(2^{17}) \rightarrow (1/(\tau*s+1)) \rightarrow (2^{10})$ 
Eqm4    ;Rho  $(2^{17})$ 
Eqm5    ;Se calcula  $V_r$  estrella  $(2^{14})$ 
Eqm6    ;Se calcula  $V_r$  tilda  $(2^{14})$ 
Eqm7    ;Se calcula  $e$   $(2^{10})$ 
Eqm8    ;Se hace la división  $e/V_c$   $(2^1)$ 
Eqm9    ;Expulsa el valor del control por una salida analógica
```

Las siguientes líneas despliegan la señal de control  $u$ , el voltaje en el capacitor  $V_c$ , y el parámetro  $\rho$  hacia los puertos E, A y C, respectivamente. Cabe aclarar que el voltaje

que representa al control  $u$  esta multiplicado 10 veces para permitir la máxima resolución por parte de la tarjeta de interfaz.

```
DAC    ukn+1,PEDATDIR,0 ;Muestra el valor del control por el Pto.E
DAC    cVc,PADATDIR,0  ;Muestra Vc por el Puerto A
DAC    m4,PCDATDIR,0   ;Muestra el valor de rho por el pto C
PUERTO 0FF00h,PBDATDIR ;Indica por el pto B que el proceso termino
```

La siguiente línea marca el fin de la interrupción volviendo al lazo infinito

```
RET
```

La rutina de “botón de pánico” deshabilita el proceso, específicamente manda un cero a las salidas analógicas, y deshabilita completamente las interrupciones. Esta situación sigue indefinidamente a menos que se reinicialize el DSP. Esta rutina tiene el máximo nivel de prioridad de interrupción.

```
ser1:    ;Servicio de interrupcion de nivel 1
```

La siguiente instrucción dirige el puntero de datos de memoria hacia la pagina de interrupciones.

```
LDP     #PIVR >> 7h
```

Verifica si la interrupción fue hecha por la activación en alto de la terminal PDPINTA-, cuyo código de interrupción es #20h, de lo contrario, se retorna al programa principal

```
LACL    PIVR                ;Se obtiene el causante de la interrupción
XOR     #0020h              ;Se activo el pin de interrupción A?
BCND    INT120,EQ           ;si, vete a la rutina de interrupción
RET     ;caso contrario regresa
```

Las siguientes líneas de código son utilizadas para responder a la interrupción, y dirigir al programa a la subrutina de desactivación de procesos.

```
LDP     #PIVR >> 7h
LACL    PIVR
XOR     #0020h
BCND    INT120,EQ
RET
```

La siguiente línea marca el inicio de la rutina de desactivación, a la cual es dirigido el contador del programa por las instrucciones anteriores.

```
INT120: ;Se activo el pin de interrupcion A
```

En las siguientes instrucciones se muestra como se deshabilitan las interrupciones limpiando la habilitación de las mismas

```
LDP     #0h                ;Pagina de los vectores de interrupcion
```

```
SPLK    #0h,IMR           ;Deshabilita interrupciones
LDP     #0E1h           ;Pagina de los puertos
```

Finalmente se envía un cero a los puertos en la tarjeta de interfaz.

```
SPLK    #0FF80h,PADATDIR
SPLK    #0FF80h,PCDATDIR
SPLK    #0FF80h,PEDATDIR
```

Al salir de esta rutina de interrupción se retorna al programa con las interrupciones deshabilitadas. Con esto se hace caso omiso a la petición de interrupción del Temporizador 2 encargado de llevar el tiempo de muestreo, deshabilitándose el controlador.

```
RET
```

Vectores de interrupción no usados.

```
ser2: RET
ser4: RET
ser5: RET
ser6: RET
      .end
```

#### 4.7. Evaluación experimental.

El controlador propuesto se programó en la tarjeta del DSP eZDSP320LF2407A la cual genera la señal de control  $u$  que es alimentada a la tarjeta dSPACE donde se emula la planta integrada por el DVR, la carga no lineal y la fuente de voltaje. Para realizar la interconexión entre las tarjetas del DSP y dSPACE, se construyeron tarjetas de interfaz como se muestra en la Fig 4.6. Estas últimas son básicamente limitadoras de voltaje y reforzadoras de corriente. Este conjunto hace un equipo ideal para la evaluación experimental de controladores en sistemas de potencia sin requerir la implementación física de los mismos. Las gráficas obtenidas en la prueba del sistema propuesto se muestran a continuación

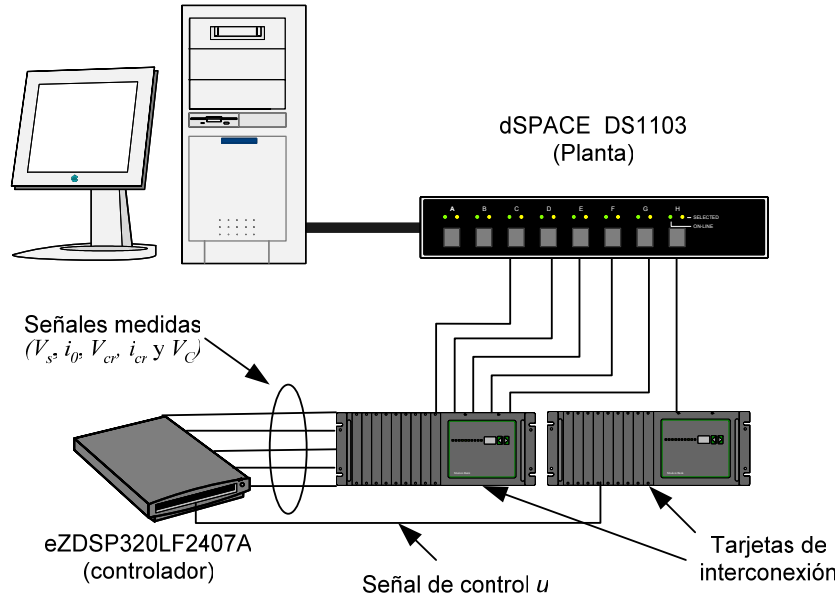


Fig 4.6. Interconexiones entre la tarjeta dSPACE, la tarjeta de evaluación eZDSP a través de las tarjetas de interfaz.

#### 4.8. Gráficas para de los resultados en la salida.

La Fig 4.7 muestra las gráficas de las señales (emuladas en la dSPACE) de voltaje de línea distorsionado y corriente de carga del puente rectificador no controlado al que se ha conectado una carga de  $25 \Omega$ . A continuación, en la Fig 4.8, se muestran las mismas señales pero conectando una carga de  $10 \Omega$  al puente rectificador.

Al voltaje de línea le fue agregado, aparte de la componente fundamental, señales de armónicos de alto orden, esto es,  $3^{er}$ ,  $5^{o}$  y  $7^{o}$  armónico. Por limitantes en la dSPACE fue imposible agregar más componentes, sin embargo, se cree que la adición de estos armónicos es más que suficiente, ya que representan una distorsión armónica de 11.62%, un valor ligeramente superior que en la mayoría de los casos observados en la realidad.

Así, el voltaje de línea tiene una visible distorsión con respecto a una señal senoidal pura, de hecho, su forma tiende a una señal triangular al poseer armónicos impares como se aprecia en las Figs. 4.7 y 4.8.

En la simulación hecha en la tarjeta dSPACE, se tomaron los siguientes valores como parámetros del DVR:  $C = 4400 \mu F$ ,  $C_r = 10 \mu F$ ,  $L_r = 3 \text{ mH}$ ,  $R = 3 \text{ k}\Omega$ . La relación del transformador fue 1 a 10, esto es,  $a=0.1$ . Para el rectificador no controlado de diodos, el capacitor se fijo en  $C_\theta = 330 \mu F$  y el valor de la resistencia, como se menciona, es seleccionable entre  $R_\theta = 10 \Omega$  y  $R_\theta = 25 \Omega$ . En cada grafica se escriben las escalas reales de las señales conforme a los factores de escalamiento usados para representar a dichas señales dentro de la tarjeta dSPACE.





(THD) es de 41%. El espectro en frecuencia de las señales de voltaje de línea y corriente de carga son mostradas en la Fig 4.9 Cabe mencionar que ambas distorsiones, la del voltaje de línea y la de la corriente de carga no lineal, no afectan el funcionamiento básico de nuestro controlador, ya que ambas perturbaciones fueron consideradas en el diseño del mismo como se describió en el Capítulo 2.

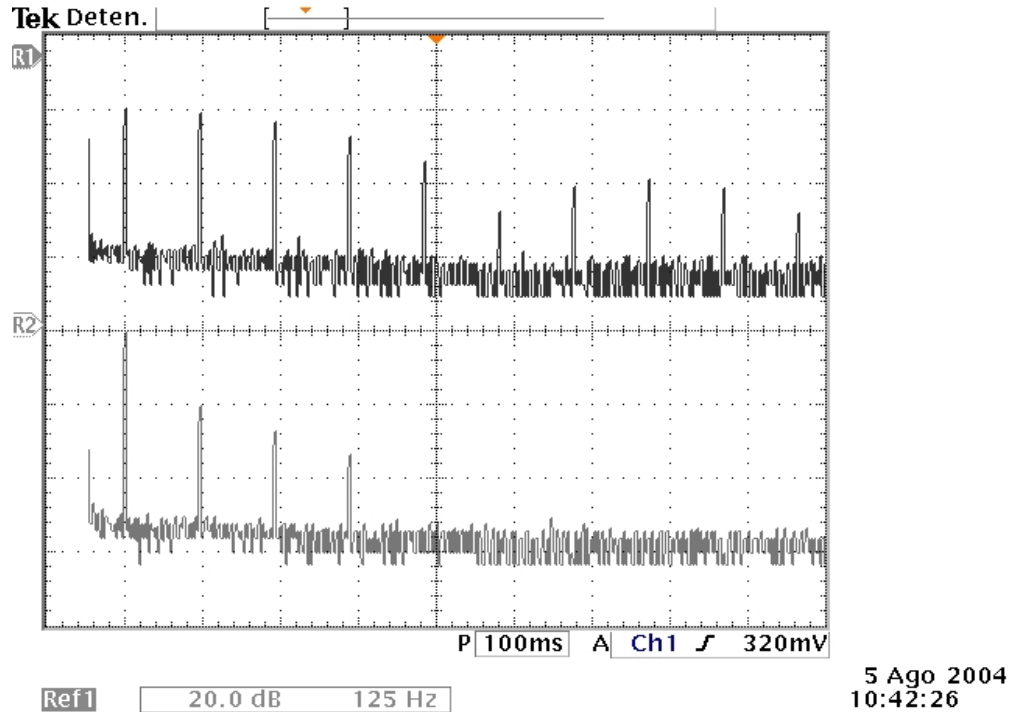


Fig 4.9. Espectro en frecuencia de: **(arriba)** corriente de carga (20 dB/div) y **(abajo)** voltaje de tensión de la línea sin compensar (20 dB/div).

Para el controlador propuesto se desarrollaron 2 tipos de pruebas.

- La primera consiste en observar la respuesta del voltaje de salida en estado estacionario bajo dos condiciones de carga fijas. Para esto, en el rectificador no controlado de diodos, se aplican las resistencias de carga de  $25 \Omega$  y  $10 \Omega$ .
- La segunda prueba consiste en observar las respuestas transitorias ante un cambio repentino en la corriente de carga. Para esto, se inicia con una resistencia de  $50 \Omega$  como carga en el rectificador de diodos y en un instante dado se realiza el cambio a una resistencia de  $10 \Omega$ , finalmente se regresa a la resistencia de  $50 \Omega$ .

En la Fig 4.10 se muestra la respuesta el voltaje de salida del DVR teniendo como carga en el rectificador una resistencia de  $25 \Omega$ . En la Fig. 4.11 se muestran las mismas curvas, pero esta vez para una resistencia de carga de  $10 \Omega$ .

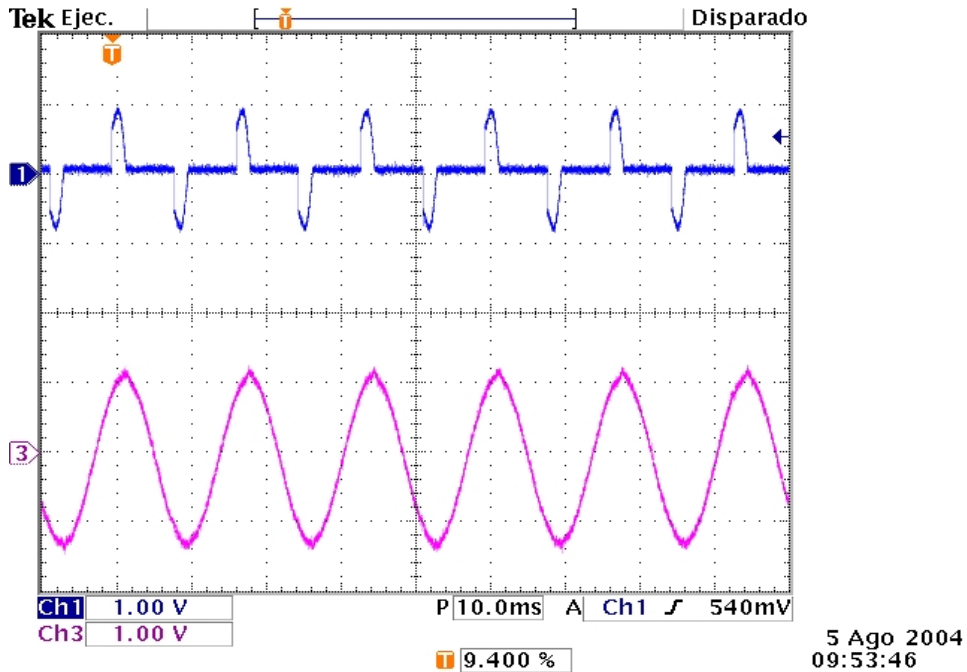


Fig. 4.10. (arriba) Corriente de carga correspondiente a una resistencia de  $25\Omega$  en el rectificador (47 A/div) y (abajo) respuesta del voltaje de salida ya compensado (158 V/div).

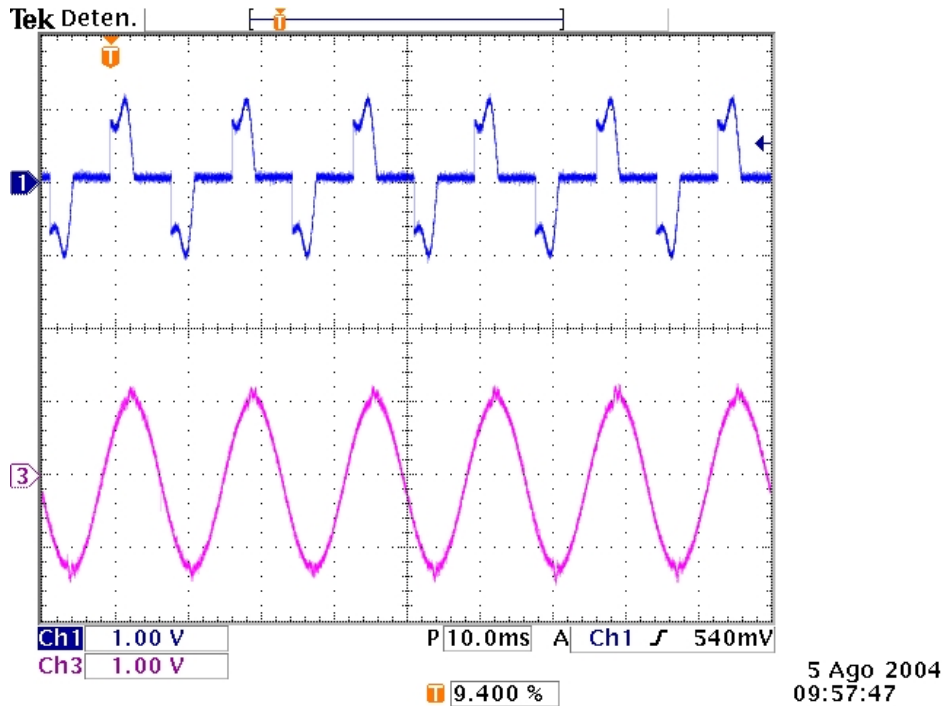


Fig. 4.11. (arriba) Corriente de carga correspondiente a una resistencia de  $10\Omega$  en el rectificador (47 A/div) y (abajo) respuesta del voltaje de salida ya compensado (158 V/div).

En las gráficas anteriores se observa que no importando la forma de la corriente de carga, ni su magnitud, se podrá compensar un voltaje de entrada distorsionado. Cabe

notar que la única condición a cumplir es que, dicha corriente de carga sea suficientemente superior a la corriente del capacitor del filtro de alisamiento, esto es, se requiere cumplir con la condición (2.34). El espectro de frecuencias correspondiente al caso en que  $R=10\ \Omega$ , esto es, el asociado a las respuestas de la Fig 4.11, se muestra en la Fig 4.12.

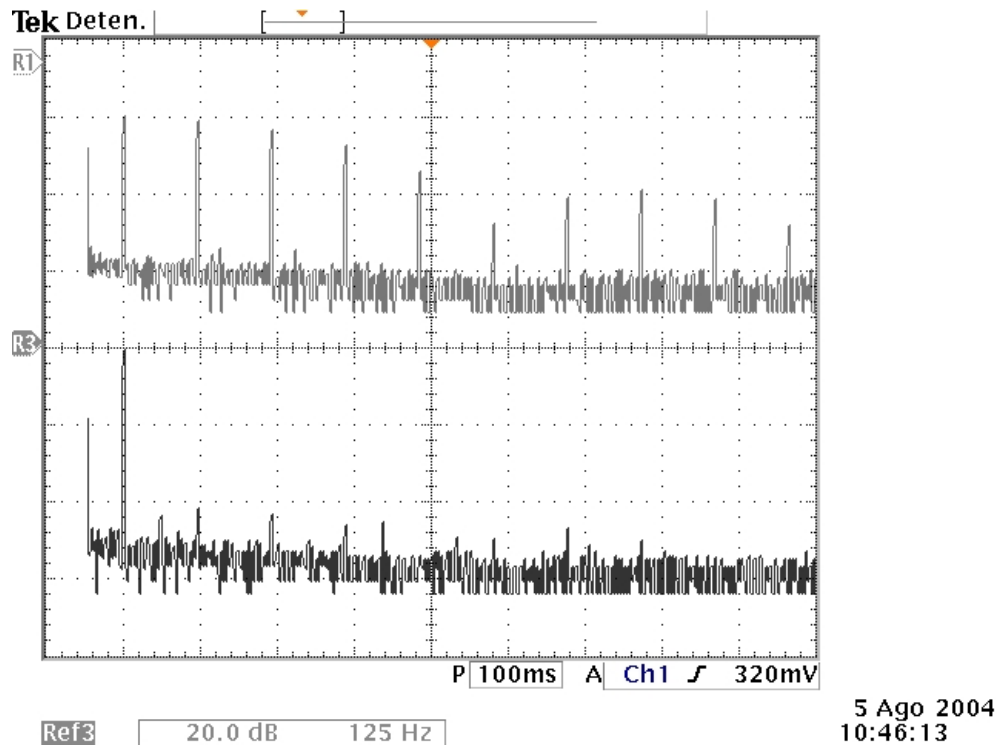


Fig. 4.12. Espectro en frecuencia de: **(arriba)** las señales de corriente de carga (20 dB/div), y **(abajo)** la señal de voltaje compensado (20 dB/div).

También de lo anterior se confirma que la señal de voltaje compensado ha quedado formada principalmente por su componente fundamental, esto es, los componentes armónicos de orden superior han disminuido su intensidad de manera contundente. El aspecto de la señal de entrada y de salida del voltaje se muestran, a manera de comparación, en la Fig 4.13 En la señal de voltaje compensado se puede observar una ligera deformidad en los extremos de la señal, esto se debe a la resolución del controlador analógico-digital utilizado para suministrar la señal de control a la tarjeta dSPACE. Así mismo en la Fig 4.14. se realiza una comparación entre el espectro en frecuencia de ambas señales de voltaje, donde se puede apreciar la disminución drástica del contenido armónico de la señal de salida respecto de la señal de entrada. Concluimos así que, el controlador implementado en la DSP logra resultados notables en la compensación de armónicos en la tensión de la línea bajando la distorsión armónica de un 11.62% a un 1.22% como se puede apreciar en la Tabla 4.1.

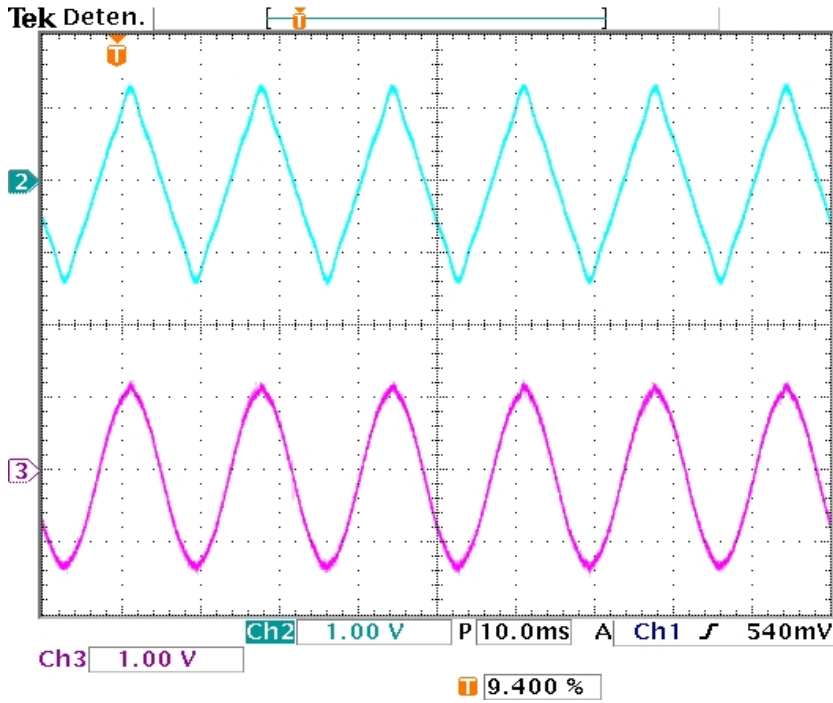


Fig 4.13. (arriba) Comparación en entre el voltaje de la fuente antes del DVR (158 V/div) y (abajo) el voltaje de salida después de ser compensado por el DVR (158 V/div).

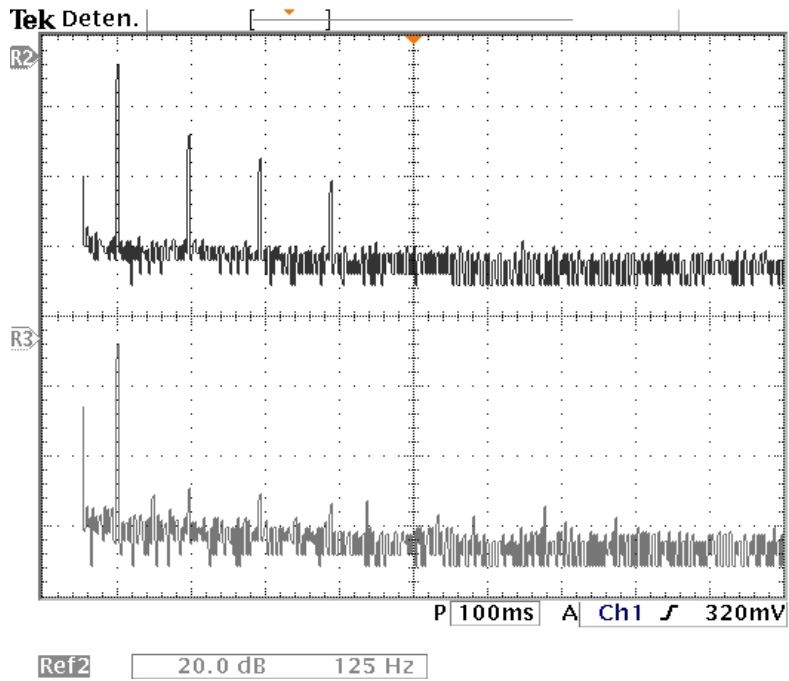


Fig. 4.14. Comparación del espectro en frecuencia de (arriba) voltaje de de la fuente antes del DVR (20 dB/div) y (abajo) voltaje de salida una vez compensado por el DVR (20 dB/div).

Tabla 4.1 Distorsión armónica total del voltaje de entrada y el voltaje compensado

Voltaje	Armónico				Distorsión armónica total (THD)
	3°	5°	7°	9°	
Voltaje de Entrada	10%	5%	2.5%	0%	11.62%
Voltaje compensado	0.853%	0.618%	0.413%	0.098%	1.22%

A continuación se muestran los resultados obtenidos cuando aparece un cambio en la corriente de carga. La resistencia en el rectificador tiene originalmente un valor de  $50 \Omega$  y es sustituida por una resistencia de  $10 \Omega$ . El objetivo es mostrar los comportamientos del voltaje en el capacitor de la planta y del parámetro  $\rho$  ante este tipo de perturbación.

En la Fig 4.15 se observan los transitorios de la corriente de carga y el voltaje compensado ante dicho cambio en la corriente de carga. Se observa que el voltaje en la entrada, y el voltaje de salida no presentan cambios perceptibles durante esta transición. En la Fig 4.16 se muestran las respuestas del voltaje del capacitor y del parámetro  $\rho$ . Se puede observar que durante el cambio de carga, el valor del capacitor cae ligeramente, y en un tiempo relativamente corto se recupera, mientras que  $\rho$  se establece en un nuevo valor. Se observó que ambas variables se estabilizan en menos de 10 segundos, que es muy deseable en casos prácticos.

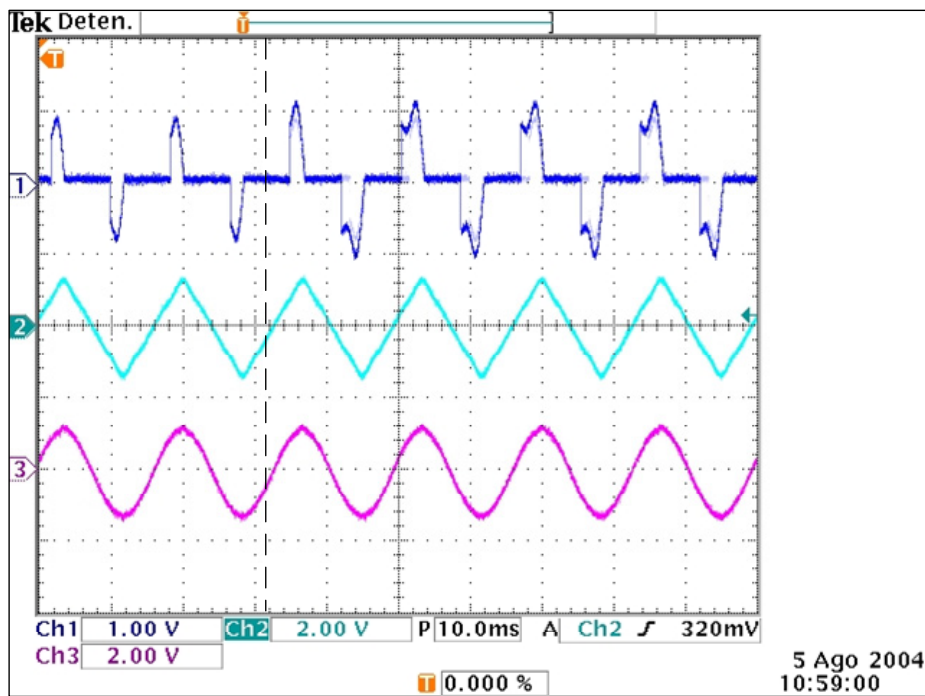


Fig. 4.15. Respuestas transitorias ante un cambio en la resistencia de carga del rectificador no controlado de diodos al reducirla de  $50 \Omega$  a  $10 \Omega$ . (de arriba a abajo) corriente de carga ( $47 \text{ A/div}$ ), señal de voltaje de entrada sin compensar ( $158 \text{ V/div}$ ) y señal de voltaje de salida compensada ( $158 \text{ V/div}$ ).

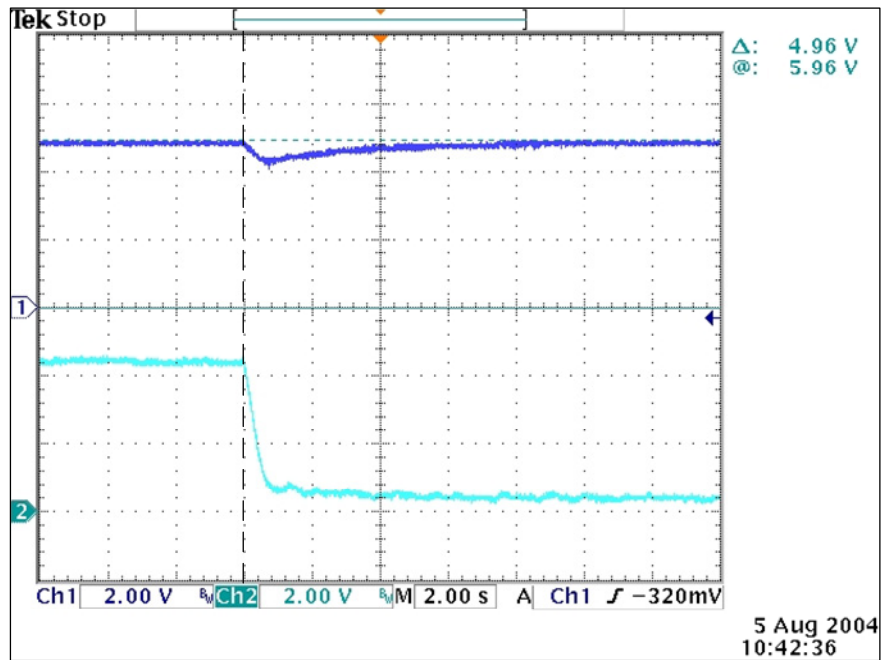


Fig 4.16 Respuestas transitorias ante un cambio en la resistencia de carga del rectificador no controlado de diodos al reducirla de  $50 \Omega$  a  $10 \Omega$ . **(arriba)** voltaje del capacitor (102 V/div), y **(abajo)** parámetro  $\rho$  (205 1/div).

En las Figs 4.17 y 4.18 se observan las respuestas transitorias obtenidas al regresar la resistencia de carga del rectificador de  $10 \Omega$  a  $50 \Omega$ . De estas figuras se obtienen similares conclusiones que en el caso anterior. Esto es, el voltaje de salida compensado no se modifica perceptiblemente a pesar de modificar la corriente de carga, sea incrementando o disminuyendo la resistencia de carga del rectificador no controlado de diodos, y además, el voltaje del capacitor mantiene su valor constante después de un transitorio relativamente pequeño. De hecho la recuperación del voltaje del capacitor y el establecimiento del nuevo valor de  $\rho$ , se efectúa en menos de 10 segundos. Cabe mencionar que el sobretiro de voltaje no va más allá del 5% del valor de referencia.

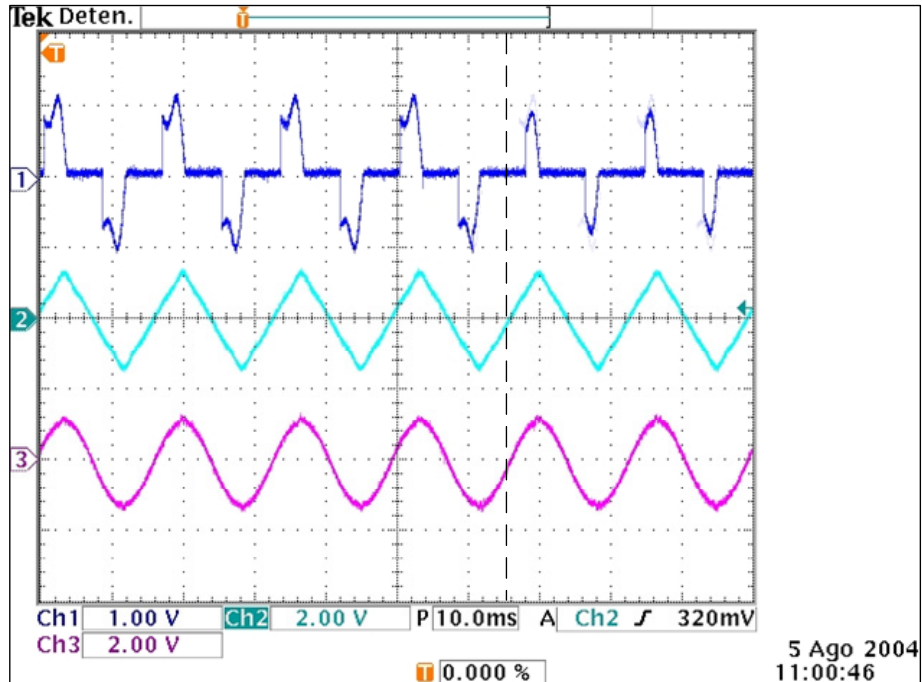


Fig. 4.17 Respuestas transitorias ante un cambio en la resistencia de carga del rectificador no controlado de diodos al aumentarla de  $10 \Omega$  a  $50 \Omega$ . **(de arriba a abajo)** Corriente de carga (47 A/div), señal de voltaje de entrada sin compensar (158 V/div) y señal de voltaje de salida compensada (158 V/div).

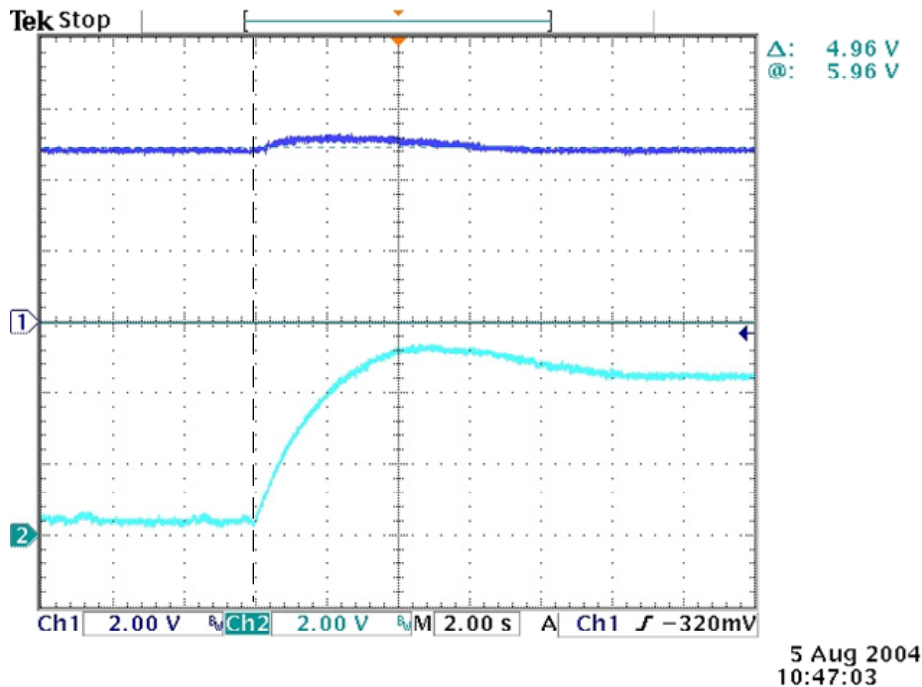


Fig. 4.18. Respuestas transitorias ante un cambio en la resistencia de carga del rectificador no controlado de diodos al aumentarla de  $10 \Omega$  a  $50 \Omega$ . **(arriba)** Voltaje del capacitor (102 V/div), y **(abajo)** parámetro  $\rho$  (205 1/div).

Las gráficas anteriores nos permiten comprobar que la ley de control propuesta cumple muy cercanamente con los objetivos establecidos. A saber, seguimiento de voltaje, y regulación del voltaje del capacitor de continua hacia un valor constante de referencia. Siendo el primer objetivo muy ligado al rechazo de perturbaciones armónicas, entonces podemos decir que el controlador propuesto logra compensar los armónicos no deseados en el voltaje de línea.

Cabe mencionar que el controlador propuesto puede compensar armónicos impares de orden superior hasta el 17<sup>o</sup> componente armónico. La razón es que cada filtro resonante tarda en ejecutarse aproximadamente 3.2  $\mu$ s, esto es 25.6  $\mu$ s, que sumados a 6.4  $\mu$ s que toman los dos filtros paso-banda en ejecutarse, se obtiene un total de 32  $\mu$ s. Más aún, se añaden 15  $\mu$ s extras para ejecutar las demás operaciones, con lo que el total del tiempo de ejecución del controlador es de alrededor de 47  $\mu$ s, siempre que el DSP este corriendo con un reloj de 30 MHz. En el caso en que el DSP tenga un reloj de 40 MHz, la compensación podría hacerse hasta el 25<sup>o</sup> armónico, en este caso, el filtro resonante así como el filtro paso-banda tardarían en ejecutarse 2.4  $\mu$ s, y se darían 12  $\mu$ s extras para ejecutar las demás operaciones. Con esto se obtiene un total de 48  $\mu$ s para el proceso del controlador.





## APENDICE A

---

### ARQUITECTURA INTERNA DEL DSP TMS320LF2407A

*En este apéndice se describe la estructura del DSP TMS320LF2407A, así como la tarjeta de evaluación eZDP320LF2407A utilizada para la realización de este proyecto.*

#### **A.1. Introducción.**

En la actualidad los DSPs juegan un papel preponderante en el control digital, los campos de aplicación son variados, pueden ir desde la industria química, pasando por las telecomunicaciones hasta la electrónica de potencia, prueba de ello es la creciente demanda de DSPs en todo el mundo.

Esta preferencia radica en la capacidad de implementar un controlador con DSP en forma más sencilla que con un sistema analógico. Estos últimos presentan ciertas desventajas, por ejemplo, la variación de las características de los componentes ante cambios de temperatura, la gran cantidad de componentes que forman al sistema y su alta susceptibilidad ante variaciones electromagnéticas.

Por el contrario el DSP puede implementar un controlador en un solo circuito integrado. Dado los pocos componentes externos que se conectan, el costo es reducido, además tiene la ventaja extra de que el propio DSP puede hacer una labor de supervisión en el proceso más compleja que un controlador análogo.

En este trabajo se utiliza la tarjeta de evaluación eZDSP2407b [16] de Spectrum Digital Incorporated, cuyo corazón es el DSP TMS320LF2407A [17] de Texas Instrument que posee las siguientes características:

- Arquitectura Harvard.
- 14 convertidores analógico a digital (ADC) de 10 bits de resolución.
- Salidas PWM con ciclos de trabajo independientes.
- Suma, resta y multiplicación signadas implementadas por hardware.
- Puede trabajar hasta 40 Millones de instrucciones por segundo (MIPS).
- Permite la conexión hacia la PC vía puerto paralelo, lo que facilita la escritura y depuración de programas.

## **A.2. La Arquitectura Interna.**

El corazón de la tarjeta de evaluación eZDP320LF2407A es el procesador digital de señales (DSP) TMS320LF240A, el cual es un DSP de 16 bits especializado en la implementación de algoritmos de control. Está basado en una modificación de la arquitectura Harvard la cual soporta diferentes canales para la transmisión de información tanto para el espacio de programa como para el espacio de datos.

Un canal por separado es reservado para soportar el uso de una gran cantidad de periféricos. El espacio de memoria destinado al uso de periféricos es direccionado en el espacio de memoria de datos a través de un sistema especial. Esto permite que todas las instrucciones, accedando la memoria de datos, puedan ser utilizadas para manipular los datos provenientes de dispositivos periféricos como aparece en la Fig. A.1 (tomada de [17] página 2-4).

El uso de canales separados para la memoria de programa y de datos permiten la ejecución simultanea de operaciones, por ejemplo, mientras un dato se esta multiplicando, el anterior producto esta siendo sumado al acumulador, y al mismo tiempo, la siguiente instrucción se carga para ser decodificada a fin de dar mayor rapidez al DSP.

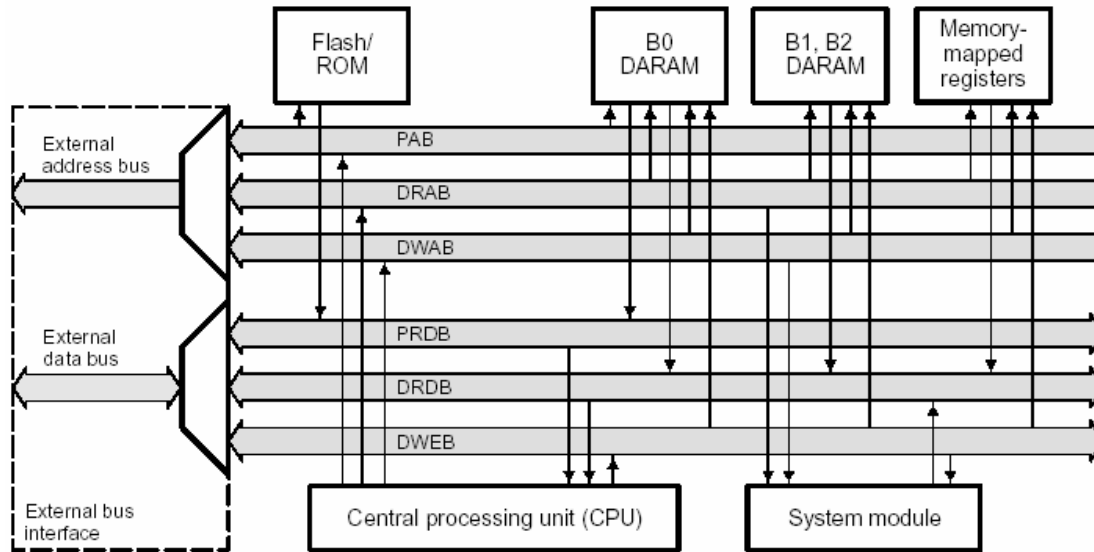


Fig. A.1. Estructura de los canales de transmisión del sistema dentro del DSP.

Tal paralelismo se traduce en que, instrucciones aritméticas, lógicas, y de manipulación de bits, se puedan hacer en un solo ciclo de máquina. Además el DSP incluye mecanismos de control para el soporte de interrupciones, repetición de operaciones, y llamadas a funciones o subrutinas.

Cabe agregar que mientras el DSP esta ejecutando el programa, algunos módulos periféricos, como lo son el convertidor analógico-digital (ADC), salidas de modulación de ancho de pulso (PWM), manejador de eventos (EVM) (responsable de los temporizadores), las salidas seriales y de red de área local (LAN), funcionen independientemente.

El reloj del sistema es de 29.4912MHz, de hecho, el DSP cuenta con un multiplicador de frecuencia interno, que dobla la frecuencia dada por un cristal de 14.7456Mhz, el fabricante no recomienda que el reloj del DSP este por arriba de los 40Mhz.

### A.3. Unidad de Procesamiento aritmético.

La unidad de procesamiento del DSP TMS2407 contiene:

- Una unidad central aritmética lógica de 32 bits (CALU)
- Un acumulador de 32 bits (ACC).
- Unidad de corrimientos de entrada y de salida para el CALU.
- Multiplicador/acumulador (MAC) de 16bits x 16bits.
- Unidad de corrimiento para el resultado de la multiplicación (PREG).
- Unidad lógica de generación de direcciones, que incluye ocho registros auxiliares y una unidad auxiliar de registros aritméticos (ARAU).
- Generación lógica de direccionamiento de programa.

A continuación se describen las unidades utilizadas en el presente proyecto.

### **A.3.1 Unidad central aritmética lógica (CALU) y acumulador.**

La CALU realiza una gran variedad de operaciones aritméticas (suma y resta, con o sin signo, valor absoluto, complemento, normalización), así como operaciones lógicas (AND, OR, NOT, XOR), muchas de las cuales se ejecutan en un solo ciclo de reloj. Estas operaciones se dividen en 4 categorías.

- Suma de 16 bits.
- Resta de 16 bits.
- Operaciones de lógica booleana.
- Prueba de bits, corrimientos y rotaciones.

La CALU tiene siempre como operando al acumulador, el otro operando puede ser provisto por una localidad de memoria de datos, o bien, por el registro de producto (PREG). Dado que la CALU realiza operaciones booleanas, se puede implementar la manipulación de bits con operaciones, más aún, la rotación y el corrimiento se realizan también en el acumulador.

Una vez que la CALU termina una operación aritmética o lógica, el resultado es puesto en el acumulador de 32 bits. Para algunas instrucciones, la bandera de extensión de signo (SXM) determina si la extensión de signo esta siendo usada ( $SXM = 1$ ) o si ha sido deshabilitada ( $SXM=0$ ) por la CALU durante la ejecución de estas instrucciones.

Se le llama unidad central ya que existe, dentro de la arquitectura del DSP, otra unidad aritmética-lógica (ALU) usada para los registros auxiliares (AR), la cual será descrita mas adelante.

### **A.3.2. Acumulador**

El acumulador se divide en 16 bits mas significativos (bits 31 – 16) (ACCH), y 16 bits menos significativos (bits 15-0) (ACCL) como se muestra en la Fig. A.2 (tomada de [17] página 4-8). Cada parte del acumulador puede ser cargada individualmente por los registros en memoria, y a su vez, cada parte del acumulador puede ser almacenada en un registro de memoria.

La CALU, como se mencionó, guarda el resultado en el acumulador al realizar una operación, por otro lado, se generan banderas (flags) de estado que reflejan, según la operación realizada, propiedades asociadas al resultado en el acumulador. Notar que no todas las instrucciones asociadas al acumulador afectan a las mismas banderas, incluso existen instrucciones que no cambian el estado de estas. Las banderas asociadas a las operaciones, cuyo resultado se almacena en el acumulador, modifican los bits de los registros de estado.

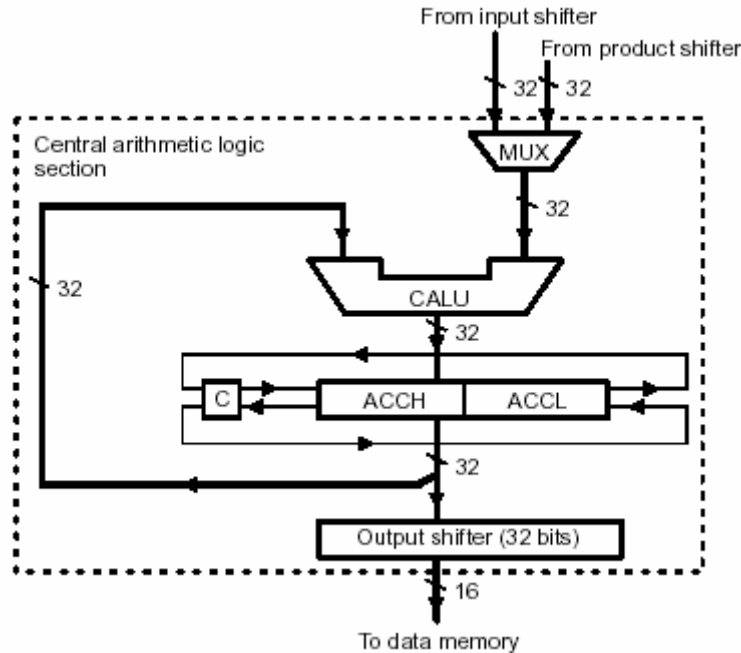


Fig. A.2 Diagrama de bloques del acumulador.

### Registros de estado (ST0 y ST1)

El DSP utilizado contiene dos registros de estado, el ST0 y el ST1, quienes contienen bits de estado y de control. Una de las ventajas del DSP, es que estos registros pueden ser guardados o cargados directamente a registros en memoria, lo que permite manipular o guardar el estado de las banderas durante las subrutinas.

A continuación se describen brevemente algunas de las banderas usadas en este trabajo.

**Corrimiento (C):** (Bit 9 del registro de estado). Esta bandera es afectada por sumas y restas. Denota el desbordamiento de una suma no signada, y en el caso de la resta, determina el signo del resultado, o bien, la existencia de un desbordamiento durante la ejecución de dicha operación.

Cabe notar que el bit de corrimiento es parte del operando durante los corrimientos o rotaciones, esto es, si alguna de estas operaciones es a la izquierda, el bit de corrimiento pasará al bit más significativo del acumulador, pero si es a la derecha el bit de corrimiento pasará al bit menos significativo del acumulador.

**Modo de desborde (OVM):** (Bit 11 del registro de estado). Refleja el resultado del desborde en el acumulador tras una operación. Si OVM=1, el resultado de la CALU se almacenará en el acumulador en las siguientes condiciones:

- Si el desborde fue en dirección positiva, el acumulador se carga con el valor más positivo (#7FFFFFFh).

- Si el desborde fue en dirección negativa, el acumulador se carga con el valor más negativo (#800000000h).

En el caso en que OVM=0, el acumulador se desbordará, lo cual no es conveniente para operaciones en punto fijo.

**Detección de modo desborde (OV):** Esta bandera se activa cuando un desbordamiento ha sido detectado en una operación en la CALU (OV = 1), en caso contrario, la bandera de desbordamiento se mantendrá en cero (OVM = 0).

**Prueba de control de bit (TC):** Es puesta a 0 ó 1 según el valor de bit a probar. Notar que esta instrucción no es exclusiva del acumulador dado que también se puede realizar con localidades de memoria.

### A.3.3. Estructura del Multiplicador.

El DSP cuenta con un multiplicador por hardware de 16 bits x 16 bits, que produce un producto de 32 bits, signado o no signado, en un solo ciclo de maquina. La estructura del bloque multiplicador se muestra en la Fig. A.3 (tomada de [17] página 4-5), y consiste en:

- Un registro temporal de 16 bits (TREG), el cual contiene a uno de los multiplicandos.
- El multiplicador, quien multiplica el contenido de TREG por un registro en memoria de programa o memoria de datos.
- Un registro de producto de 32 bits (PREG) que recibe el resultado de la multiplicación.
- El corredor (shifter) del producto, que escala el contenido del registro PREG antes de pasar al acumulador.

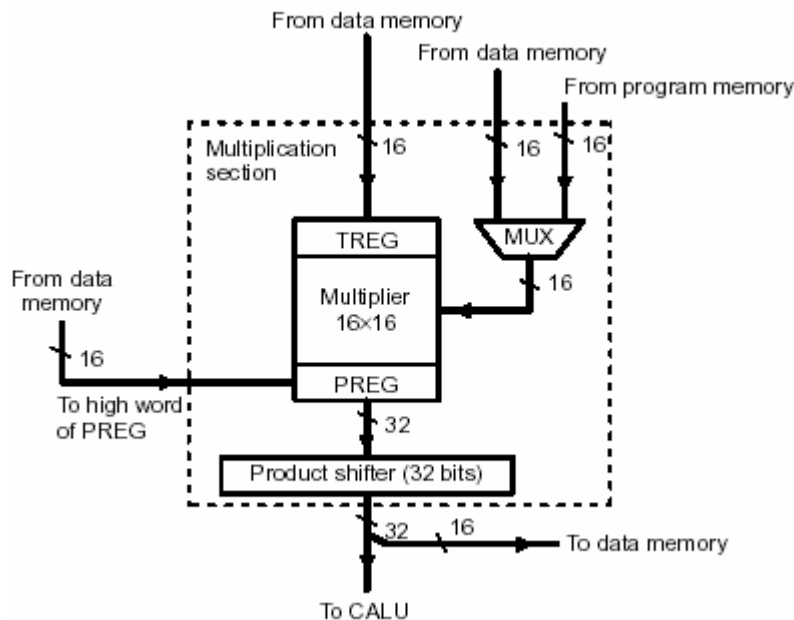


Fig. A.3. Estructura del multiplicador del DSP.

En una multiplicación existen 2 operandos, uno de ellos es siempre el registro TREG, el cual debe ser cargado antes de la operación, el otro operando puede ser el contenido de un registro en memoria de datos o de programa. Tanto los operandos como el resultado son tratados en una multiplicación como números de complemento a 2, exceptuando el caso en que se utiliza la instrucción MPYU (Multiplicación sin signo).

Después que la multiplicación fue realizada, el resultado de 32 bits es guardado en el registro de producto (PREG), donde se puede correr antes de su transferencia al acumulador.

#### **A.3.4 Unidad de escalamiento por corrimiento.**

El DSP posee tres corredores (shifters) de 32 bits para escalamiento, extracción de bits, manipulación de la aritmética de los números, y prevención del desbordamiento.

**Corrimiento a la entrada.** Se aplica a la unidad que hace los corrimientos de 16 bits para alinear los registros antes de entrar a la CALU (registro de 32 bits).

**Corrimiento de salida.** Este corrimiento de 1 a 7 bits es realizado por hardware a la salida del acumulador para almacenarlo en registros de la memoria de datos. Notar que el acumulador permanece sin cambios.

**Corrimiento del Producto de la Multiplicación.** Se hace el corrimiento sobre el registro PREG, quien recibe el producto de la multiplicación con la finalidad de manejar la suma de productos sucesivos en el acumulador. Este corrimiento esta determinado por los 2 bits que en conjunto se les llama “modo de corrimiento del producto” (SPM). Este último recorre 1 bit a la izquierda (SPM=#01b), 2 bits a la izquierda (SPM=#10b), 6 bits a la derecha (SPM=#11b), o bien inhabilita el corrimiento SPM=(#00b).

#### **A.4. Registros Auxiliares.**

Los ocho registros auxiliares (AR7–AR0) proveen un flexible y poderoso direccionamiento indirecto. Así, cualquier localidad en los 64KB de memoria puede ser accesado mediante el uso de los 16 bits contenidos en los registros auxiliares.

Las operaciones que realiza el DSP van dirigidas al manejo de un solo registro auxiliar a la vez. Para seleccionar un registro auxiliar específico se debe cargar el número (entre 0 y 7) de dicho registro en los 3 bits correspondientes al apuntador del registro auxiliar (ARP) dentro del registro de estado (ST).

Al registro apuntado por ARP se le puede llamar “registro auxiliar seleccionado” (AR seleccionado). Durante el procesamiento de una instrucción, el contenido del AR



seleccionado es tomado para apuntar el lugar en memoria de datos donde será seleccionado el operador.

El DSP tiene una unidad lógica-aritmética para la manipulación de los registros auxiliares conocida como ARAU. Esta última funciona en forma paralela con la CALU. La ARAU se encarga de las siguientes operaciones.

- Incrementa o decrementa el registro auxiliar por un valor especificado. Esta es una manera de direccionar registros consecutivos en memoria.
- Suma o resta un valor constante de 8 bits al registro auxiliar especificado en la instrucción.
- Compara el contenido del registro auxiliar seleccionado con el contenido del registro auxiliar cero (AR0). El resultado es puesto en el bit de prueba (TC) del registro de estado.

El diagrama de bloques que representa el manejo de los registros auxiliares se muestra en la Fig. A.4 (tomada de [17] página 4-12).

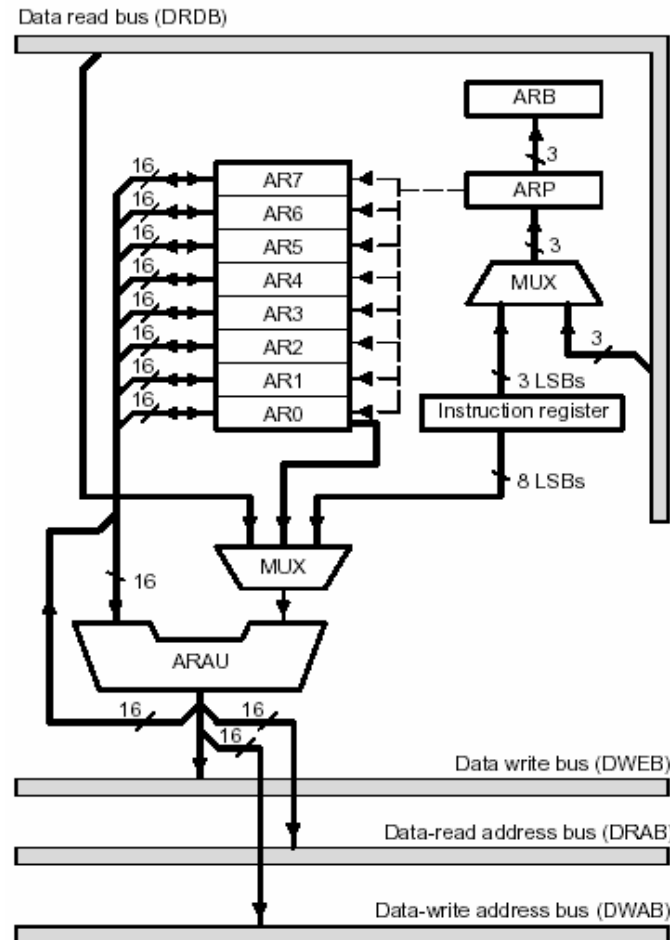


Fig. A.4. Diagrama de bloques para los registros auxiliares.

## **A.5. Espacios de Memoria.**

El DSP tiene 16 líneas de dirección que le permiten acceder 4 espacios de memoria individuales (224K en total) enumerados a continuación. Cabe hacer la aclaración de que el tamaño usado para la medición de memoria es la palabra de 16 bits (WORD) y no el octeto de 8 bits (BYTE):

- 64K de espacio de programa. Contiene las instrucciones a ejecutarse y los datos inamovibles usados durante el desarrollo del programa.
- 64K de espacio local de datos. Contiene datos usados por las instrucciones.
- 32K de espacio global de datos. Aquí se comparten datos con otros dispositivos, o bien, puede servir como espacio de memoria adicional.
- 64K de espacio de entrada-salida (I/O). Sirve para la interfaz con periféricos externos.

Como se menciono anteriormente el DSP esta diseñado con base en una modificación mejorada de la Arquitectura Harvard. Cada espacio de memoria es accesado por un canal de datos individual. Estos canales son independientes entre si, lo cual permite acceder al mismo tiempo a la memoria de datos y a la memoria de programa.

La CALU puede hacer operaciones con registros provenientes de 3 espacios de memoria distintos, los cuales se accesan en diferentes ciclos de la decodificación de una instrucción.

Las ventajas de usar la memoria integrada al DSP son:

- Mejor desempeño que las memorias externas (ya que la velocidad de la transmisión de datos esta a la medida del DSP evitando memoria lentas).
- Menor costo que la implementación de memorias externas.
- Consumo menor de energía que usando una memoria externa.

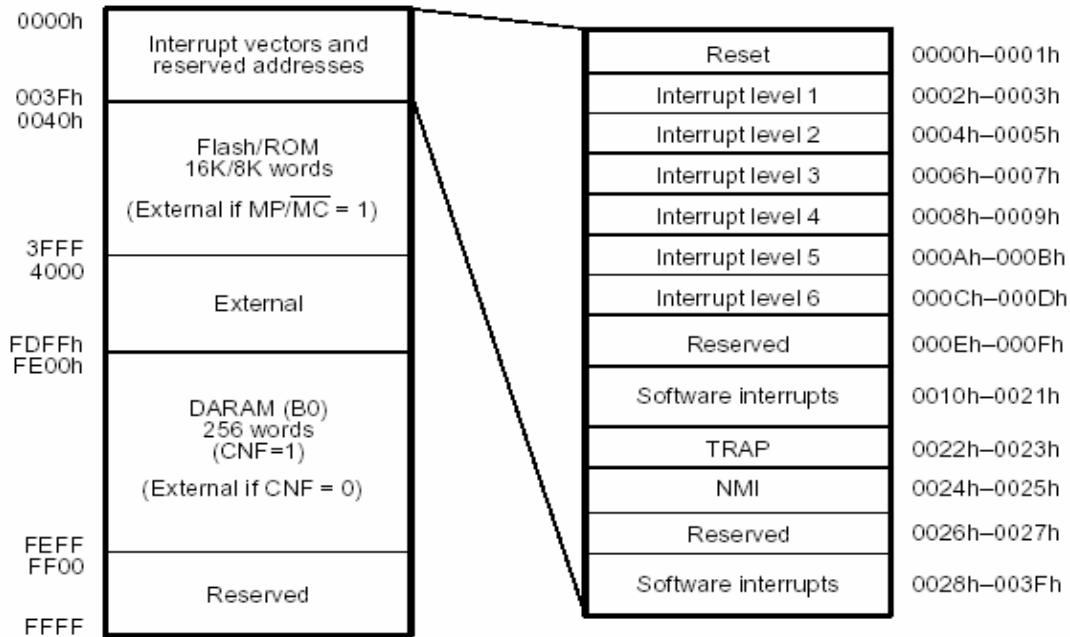
La principal desventaja es la imposibilidad expandir el espacio de memoria total que puede manejar el DSP a tamaños mayores a los 86KB para la memoria de datos.

Por razones de espacio solo se describirá el espacio de memoria correspondiente al de programa y de datos, ya que son los únicos necesarios para el desarrollo de la presente tesis.

### **A.5.1 Memoria de Programa**

En esta parte de la memoria reside el código del programa, también se pueden colocar tablas de datos y operandos inmediatos que no se modifican continuamente ya que residen aun después de apagar el DSP. El DSP puede manejar el direccionamiento de más

de 64 K de memoria de programa con tamaño de 16 bits cada una, como se observa en la Fig. A.5 (tomada de [17] página 3-4).



**Note:** Flash/ROM memory includes the address range 0000h-003Fh.

Fig. A.5. Mapa de Memoria de programa en el TMS320LF2407A.

Este espacio incluye a la memoria Flash (ROM eléctricamente borrable) y DRAM (RAM de acceso dual), ambas integradas en el DSP. Cuando se genera una dirección, que va mas allá del espacio que ocupa la memoria integrada, el DSP automáticamente configura el acceso externo por medio de las señales adecuadas de comunicación con el dispositivo periférico (si este es disponible).

### A.5.2. Memoria de Datos.

La memoria de datos puede direccionarse a más de 64K de localidades de memoria de 16 bits cada una. Físicamente, la memoria esta dividida en 3 bloques, a saber, B0, B1 y B2. El bloque B0 puede ser configurado como memoria de datos o de programa, y los bloques B1 y B2 son usados exclusivamente como memorias de datos.

Se puede direccionar la memoria de manera directa o indirecta. Cuando el direccionamiento directo es usado, la memoria se divide en bloques de 128 WORD's llamados páginas de datos. Los 64K se dividen en 512 paginas de 128 registros cada una. El direccionamiento a la pagina actual depende del valor de los nueve bits del apuntador de datos (DP) contenido en el registro de estado (ST0).

El direccionamiento se realiza de la siguiente manera:

- Cada uno de los 128 espacios, en una pagina de memoria, es referenciado por 7 bits, los cuales se obtienen directamente de la instrucción, y están representados por los bits menos significativos de esta.
- A los 7 bits anteriores se les agrega el valor del apuntador de datos (DP) como su parte más significativa. Así, cuando se realiza un direccionamiento indirecto se debe tener claro en que pagina esta el apuntador de datos, a fin de no cometer errores en la posición del registro al que deseamos operar.

El direccionamiento indirecto, por su parte, usa los registros auxiliares como apuntadores absolutos del registro de datos, pudiendo manipular con un registro toda la longitud de la memoria de datos (#0000h - #FFFFh). El inconveniente con este modo resalta en el tiempo de demora, ya que el direccionamiento directo toma 2 ciclos de reloj del DSP, en cambio el direccionamiento indirecto toma 3 ciclos por contener un argumento más.

## **A.6. Periféricos Internos.**

El DSP TMS320LF2407A incluye varios periféricos internos para el manejo de señales y la comunicación de datos, entre estos destacan:

- Manejador de Eventos (EVM).
- Controlador de área local (CAN).
- Puertos de comunicación serial (SCI, SPI).
- Convertidor analógico-digital (ADC).
- Características de seguridad como guardián (WT) y protección para casos anormales en sistemas de potencia (PDP).

Cabe mencionar que estos dispositivos perifericos funcionan de manera independiente a la CALU, es decir, están implementados totalmente por hardware, ahorrando tiempo de ejecución del programa, y con la posibilidad de ampliar la versatilidad del controlador implementado.

Por razones de espacio sólo se detallara el funcionamiento de los periféricos internos EVM, ADC y del guardián (WT). Cuando se conecta o reinicializa el sistema, los módulos internos se encuentran deshabilitados, para que estos no consuman energía. Esta característica es especialmente útil para el uso eficiente de la alimentación del DSP por medio de baterías.

Los periféricos internos se habilitan mediante la puesta en uno de ciertos bits del registro de configuración 1 (SCSR1).

Los registros de configuración SCSR1 y SCSR2, entre cuyas funciones destacan las siguientes: habilitar los periféricos internos, determinar el multiplicador de la frecuencia

del cristal externo para su uso de reloj de sistema, avisar de direccionamiento ilegal, determinar del lugar de la memoria física de datos, etc.

### **A.6.1 Guardián (WT).**

El guardián (WT) es un contador reinicializable de forma ascendente cuya longitud de registro es de 8 bits y cuyo reloj es un submúltiplo del reloj del sistema.

El WT protege contra fallas de software del sistema y fallas de la arquitectura del DSP por medio de una reinicialización general del sistema. Esta condición también sucede por un intento fallido al reinicializar el contador del WT, esto es, cuando se carga el registro llave de la reinicialización del WT (WDKEY) con un patrón incorrecto de valores. La manera correcta de realizar el proceso anterior es la sucesiva carga en cualquier orden de los valores #0AAAAh y #05555h.

El control del WT se logra por medio del registro de configuración del WT (WDCR), este registro incluye bits para indicar si la reinicialización anterior del sistema fue a causa del modulo de WT, este registro puede deshabilitar al WT por medio de la carga correcta de los bits en el, de otra forma, se reinicializará el sistema.

Dado que, en el proyecto mostrado, se necesita un lazo infinito para realizar la espera entre los ciclos de control, el WT se debe de deshabilitar para evitar cualquier reinicialización del sistema en medio del proceso. Esto se logra por medio la carga del valor #06Fh al registro WDCR.

### **A.6.2 Puertos Digitales de Entrada y Salida (D I/O).**

El modulo de los puertos digitales de entrada y salida proveen un método flexible para controlar tanto los puertos digitales como las funciones compartidas provenientes de otros módulos, a esta últimas se les llama funciones primarias.

La opción de entrada-salida y la función primaria en los puertos digitales esta controlada por nueve registros de 16 bits. Estos registros están divididos en dos tipos:

- Registros de control de multiplexado (MCRx). Usados para el control que multiplexa el uso del pin entre la función primaria o una entrada-salida de datos de propósito general.
- Registros de control de datos y dirección (PxDATDIR). Usados para controlar los datos y dirección de los mismos cuando el pin esta en la opción de puerto digital de entrada-salida.

Donde x es cualquiera de los puertos digitales A, B, C, D, E y F. El control y los datos de los puertos digitales están mapeados en la memoria de datos.

### A.6.3 Temporizadores de uso general (GPT).

Son usados para el manejo exacto del tiempo. El GPT contiene varias funciones que lo hacen muy versátil. En este proyecto se utilizó una de las más importantes, a saber, la ejecución de rutinas a intervalos exactos de tiempo. La implementación del proceso en la DSP requiere la ejecución de la rutina del controlador a intervalos exactos de tiempo, ya que, como se observó en el Capítulo 3, la discretización del proceso requiere de un tiempo exacto de muestreo (T).

Esto último se realizó por medio de la interrupción generada por el inicio del periodo del Temporizador 2. El temporizador 2, uno de los 4 GPT, se habilita siguiendo los pasos a continuación:

- Se reinicializa el conteo del temporizador 2 cargando el valor #0000h al registro contador del mismo (T2CNT).
- Se habilita el modo en que se opera el Temporizador 2 por medio del registro del control del temporizador 2 (T2CON), cargando hacia este registro el valor #0842h.
- Dado que el periodo de muestreo del controlador se desea a 20KHz, el periodo de ciclo del controlador debe ser de 50us, esto se logra cargando el valor obtenido de la ecuación (5.1) al registro de comparación del temporizador 2 (T2CMPR).

$$T2CMP = \frac{(29.4912MHz)(50\mu s)}{2} = 737.28 = \#2E1h \quad (5.1)$$

- Después se habilita la interrupción por periodo del comparador al cargar un #4h al registro de interrupción por banderas B (EVAIFRB).

Después de estos pasos, la rutina de interrupción ha quedado habilitada. Es en esta rutina que ha sido colocado el código correspondiente al controlador.

Cabe notar, que los otros 3 temporizadores no se habilitaron, permitiendo así, usar el modulo de PWM de manera independiente al CPU. Esto permite habilitar de manera conveniente el uso de los comparadores, los tiempos muertos y las funciones primarias para así poder generar las salidas del PWM, las cuales serán accesibles en los pines ubicados en los puertos digitales.

### A.6.4 Convertidor analógico-digital (ADC).

El ADC integrado le da al DSP un rendimiento substancial en el procesamiento de señales, ya que no se necesitan periféricos externos para la conversión de señales analógicas, además, los resultados están mapeados en la memoria de datos, por lo que los tiempos de acceso son mínimos.

De entre las características del ADC podemos resaltar las siguientes:

- ADC de 10 bits con la habilidad de muestrear y retener (S/H) la señal de entrada.
- 16 entradas analógicas multiplexadas (ADCIN0 – ADCIN15).
- Capacidad de autosecuencia, la cual se puede programar para seguir la secuencia de canales de conversión específicos.
- 16 registros de resultados diseccionados de manera independiente para guardar los resultados del convertidor (RESULT0 – RESULT15).

El ADC tiene dos registros de control ADCTRL1 y ADCTRL2, con los cuales se habilita y se configura al mismo tiempo el modo de operación del ADC.

En nuestro proyecto se requiere que el ADC realice múltiples conversiones para adquirir el valor de las 5 señales  $V_s$ ,  $i_0$ ,  $i_{cr}$ ,  $V_r$  y  $V_c$ . Estas conversiones se deben realizar en paralelo con la ejecución del programa, de tal manera que cuando se requiera el valor de dichas señales, solo se lean los resultados ya disponibles provenientes del ADC.

Para este propósito el ADC se configura de la siguiente manera:

- Se reinicializa y se detiene inmediatamente el modulo del ADC cargando el valor de #04000h al ADCTRL1.
- Se carga al registro ADCTRL1 el valor de #31D0h para habilitar al ADC en el modo de conversión ininterrumpida.
- Se especifica el número de canales a convertir por medio del registro de número de conversiones (MAXCONV).
- El valor especificado en los registros secuenciadores de canal (CHSELSEQ1 y CHSELSEQ2) determina el orden en el cual los canales se convertirán. En este caso, se carga #3210h para el CHSELSEQ1 y #0004h para CHSELSEQ2, a fin de convertir los canales 0, 1, 2, 3 y 4 (en total 5) en este orden.
- Se carga al ADCTRL2 el valor #2000h para iniciar las conversiones por medio del ADC.

Después de estos pasos el ADC convertirá continuamente y sin interrumpir los 5 canales. Los resultados de cada canal se almacenaran en los registros (RESULT0 – RESULT4). El diagrama de bloques del ADC se muestra en la Fig. A.6 (tomada de [20] página 7-6).

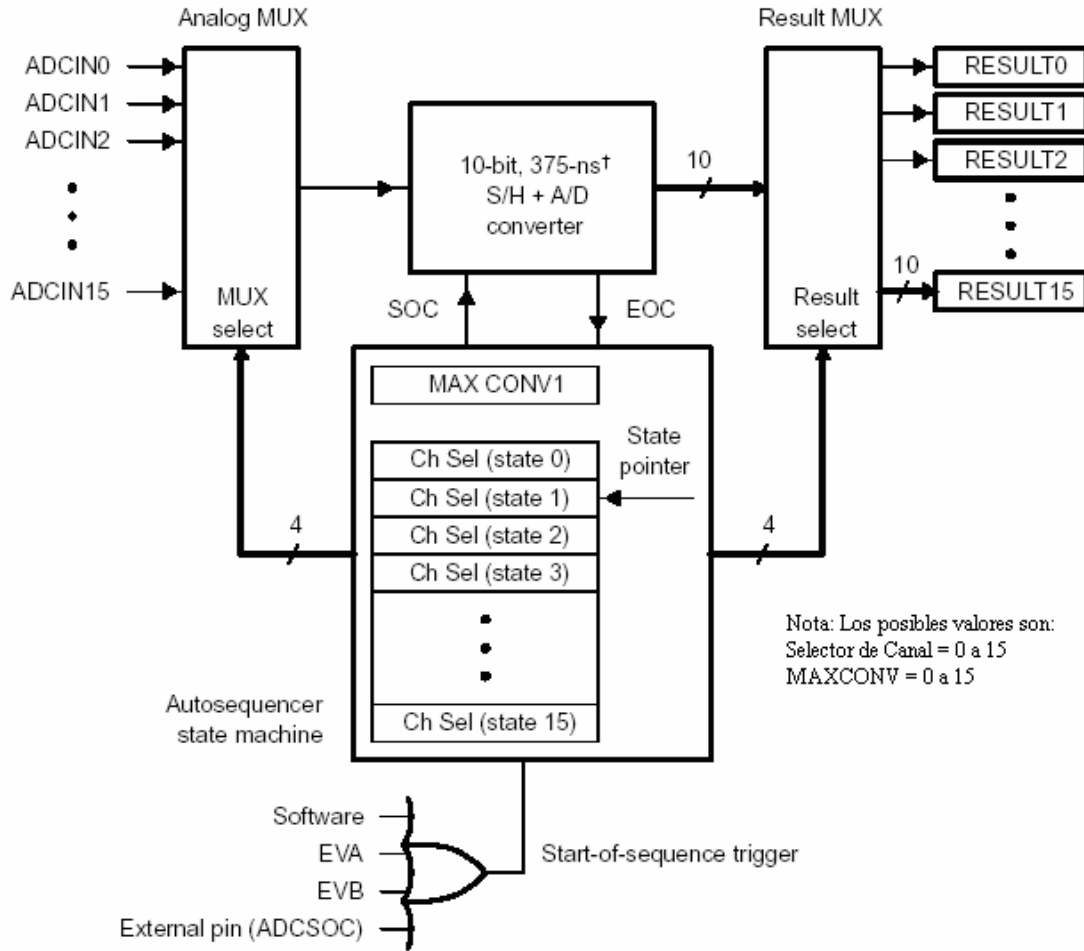


Fig. A.6. Diagrama de bloques del modulo de ADC.





## APENDICE B

---

### CODIGO DEL PROGRAMA

*Este apéndice contiene los archivos usados para la programación del algoritmo de control usado en el DSP TMS320LF2407A, tanto como las macros de operaciones, rutinas de interrupción, constantes en la memoria de programa, así como las locaciones reservadas en RAM para el manejo de números en formato Q30*

#### B.1. Programa Principal: Michael.asm

```
;Proceso Final de Para el Filtro Activo Serie

        .title "Michael AFS"      ;Titulo del programa y Autor del mismo
        .include 240x.h           ;Archivo con nemonicos
        .include vectors.asm      ;Direcciones de los vectores de interrupción
        .include macrobib.asm     ;Inicialización de dispositivos periféricos
        .include ecu.asm          ;Ecuaciones del controlador
        .include Registros.asm    ;Registros del sector de programa y RAM

        .def      _c_int0
Start    B      _c_int0      ;La direccion donde se comienza

        .text
_c_int0:      ;
start:       ;Inicio del programa principal
        Init      ;Inicialización de los registros del DSP
        SPLK      #0h,MTRX    ;Se establece el tiempo de espera para
        OUT      MTRX,WSGR    ;dispositivos periféricos como el mínimo
;En esta lineas se inicializa el ADC
        ADCreg    ;Inicializa el ADC
```

## APENDICE B: CODIGO DEL PROGRAMA

```

    Timereg          ;Inicializa el Timer
;En estas lineas se limpian los registros a usar
    LDP      #6h          ;Pagina de los registros de trabajo
    Clrblk          ;Limpia desde 300 hasta 3FF
    MKPD          ;Traslada las constantes de PMA a DMA
    Puertos        ;Inicializa los puertos
    SETC      OVM        ;Overflow habilitado
    SPM        #1        ;Recorre a la izq un lugar los resultados de PREG

ende:   B           ende      ;Solamente se espera a una interrupción

;Servicio de interrupción del controlador
ser3:   ;Servicio de interrupción del nivel 3
    NOP          ;Espera un ciclo
    PUERTO 0FFFFh,PBDATDIR ;El puerto B esta en alto
    LDP      #PIVR >> 7h ;Dirige a la pagina de las interrupciones
    LACL     PIVR        ;Se obtiene la causa de la interrupción
    XOR      #002Dh      ;compara si fue el desbordamiento del Timer 2
    BCND    SIS2D,eq     ;si fue afirmativo dirígete a la rutina de Int
    RET          ;caso contrario, regresa a ende

;Comienzo del lazo de control
SIS2D:  ;Causa de la interrupción: Desbordamiento del Timer 2
;Limpia las interrupciones de el timer2
    LDP      #GPTCONA >> 7h ;Pagina del EVM2
    LACC     #0004h        ;Se responde afirmativamente al servicio
    SACL     EVAIFRB      ;de interrupción del Timer 2
    CLRC     INTM         ;Limpiando la llamada a interrupción

;Operaciones del lazo de control
    TADC     Vs          ;
    Con2m    Vc,i0,Vs,Vr,ir

    Eqm1     ;Después  $(Vc^2 - Vd^2) / 2$   $(2^9) \rightarrow (2^{18}) \rightarrow (2^{17})$ 
    Eqm2     ;Integra a Vz tilda  $(2^{17}) \rightarrow (1/s) \rightarrow (2^{10})$ 
    Eqm3     ;LPF por Vz tilda  $(2^{17}) \rightarrow (1/(\tau * s + 1)) \rightarrow (2^{10})$ 
    Eqm4     ;Rho  $(2^{17})$ 
    Eqm5     ;Se calcula Vr estrella
    Eqm6     ;Se calcula Vr tilda
    Eqm7     ;Se calcula e
    Eqm8     ;Se hace la división e/Vc
    Eqm9     ;Expulsa el valor del control por una salida analógica

    DAC      ukn+1,PEDATDIR,0 ;Muestra el valor del control por el Pto.E
    DAC      cVc,PADATDIR,0   ;Muestra Vc por el Puerto A
    DAC      m4,PCDATDIR,0    ;Muestra el valor de rho por el pto C
    PUERTO  0FF00h,PBDATDIR ;Indica por el pto B que el proceso termino
    RET

;Rutina de interrupción del "boton de panico" el cual interrumpe el proceso
ser1:   ;Servicio de interrupción de nivel 1
    NOP
    LDP      #PIVR >> 7h
    LACL     PIVR
    XOR      #0020h
    BCND    INT120,EQ
    RET

INT120: ;Se activo el pin de interrupcion A
    LDP      #GPTCONA >> 7h ;Pagina del EVM 2

```

## APENDICE B: CODIGO DEL PROGRAMA

```
LACC    #0001h           ;Se responde a la interrupcion
SACL    EVAIFRA         ;De power pin protection A
CLRC    INTM            ;Se limpia las interrupciones
;Se deshabilita todo
LDP     #0h             ;
SPLK    #0h,IMR         ;Deshabilita interrupciones
LDP     #0E1h          ;Pagina de los puertos
SPLK    #0FF80h,PADATDIR ;Se reinician las salidas analógicas
SPLK    #0FF80h,PCDATDIR ;con cero
SPLK    #0FF80h,PEDATDIR
RET

ser2:   RET             ;Servicios de rutina no usado
ser4:   RET
ser5:   RET
ser6:   RET
        .end
;Fin del programa principal
```

### B.2. Macros de ecuaciones: ecu.asm

```
;Proyect Thorax
;Macro para inicializar los puertos
Puertos .macro
    LDP     #0E1h           ;Pagina de los puertos
    SPLK    #03F00h,MCRA    ;solo Puerto A, PB6 y PB7 como IO
    SPLK    #00000h,MCRC    ;Puerto E como salida
    SPLK    #0FF00h,MCRB    ;Puerto C como salida
    SPLK    #0FF80h,PADATDIR ;Puerto A como salidas
    SPLK    #0FF80h,PCDATDIR ;Puerto C como salidas
    SPLK    #0FF80h,PEDATDIR ;Puerto E como salidas
.endm

;Puerto por los puertos el valor analogico
Puerto .macro valor,puerto
    LDP     #0E1h           ;Pagina de los puertos
    SPLK    #valor,puerto   ;Puerto = #valor
    LDP     #06h            ;Regresa a al pagina de trabajo
.endm

;Pon la variable Q30 especificada al acumulador
LACQ .macro reg
    LAC     reg,16          ;Carga la parte alta al HACC
    ADD     reg+1,1        ;Suma la parte baja al LACC
.endm

;Suma la variable Q30 especificada al acumulador
ADDQ .macro reg
    ADD     reg+1,1        ;Suma a la parte baja al LACC
    ADD     reg,16         ;Suma a la parte alta al HACC
.endm

;Resta la variable Q30 especificada al acumulador
SUBQ .macro reg
    SUB     reg+1,1        ;Resta a la parte baja al LACC
    SUB     reg,16         ;Resta a la parte alta al HACC
.endm

;Limpia con ceros el bloque #300h -> #3FFh
Clrblk .macro
```

## APENDICE B: CODIGO DEL PROGRAMA

```

LAR   AR7,#300h      ;Carga 300h en AR7 (Apuntador)
MAR   *,AR7         ;Cambia el ARP a AR7
LAC   #0h           ;Carga el acumulador con cero
RPT   #0FFh        ;repite 256 veces la operacion, de cargar
SACL  *+           ;El cero al acumulador incrementando AR7
.endm                ;Cada vez

;Mueve las Variables constantes de PMA a DMA
MKPD  .macro
LAR   AR7,#RK1a     ;Mueve al Registro de Datos
RPT   #06h         ;Repite siete veces
BLPD  #K1a,*+      ;Mueve las constantes hacia memoria
.endm

;Lanza los datos al DAC0800 un registro Q15 o Q30
DAC   .macro reg,puerto,n ;
LACC  reg,n+8      ;Carga el registro en el acumulador
ADD   #0800h,8     ;suma con para hacer el producto para el DAC
OR    #0FF00h,16   ;Sacalo como puertos del DAC
LDP   #0E1h       ;Pagina de el DAC y los Puertos
SACH  puerto       ;Expulsa el resultado del HACC en el puerto
LDP   #06h       ;Regresa a la pagina de los registros de trabajo
.endm

;Convierte el acumulador a al formato de MQ30 (Michael's Q30)
;variable dada por nxq
Cmff: .macro nxq ;
SACH  nxq         ;Almacena la parte alta del acumulador en nxq
SFR   ;           ;Recorre a la derecha para eliminar el signo
AND   #7FFFh     ;del LACC por medio de una operacion AND
SACL  nxq+1      ;almacena el resto en nxq
.endm

;basicamente es la multiplicacion de una matriz 2x2 dada por Maa, por un
;vector Max y una entrada en Mau, los numeros deben estar en formato
;MQ30, tarda 3.25us en 24.8Mhz.
; [ Max1[k+1] ] = [ Maa(1,1) Maa(1,2) ] [ Max1[k] ] + [ Mab(1) ]* Mau
; [ Max2[k+1] ] = [ Maa(2,1) Maa(2,2) ] [ Max2[k] ] [ Mab(2) ]
pPA101 .macro Maa,Max,Mau ;SPM = 1
;Para las operaciones de la primera fila
MAC   Maa,Max+1    ;Multiplica Ah[1,1]*x1l
LAC   #0h         ;Limpia el acumulador
MAC   Maa+1,Max    ;MAC Al[1,1]*x1h
MAC   Maa+2,Max+3  ;MAC Ah[1,2]*x2l
MAC   Maa+3,Max+2  ;MAC Al[1,2]*x2h
MAC   Maa+8,Mau+1  ;MAC Bh[1]*U1
MAC   Maa+9,Mau    ;MAC Bl[1]*Uh
APAC  ;           ;Suma el ultimo producto al acumulador
SACH  MTRX        ;Almacena la parte alta de todo el producto
LAC   #0         ;En MTRX (auxiliar) y limpia el acumulador
LT    MTRX        ;y ponla en la parte baja del registro del
MPY   #1         ;producto con su signo correspondiente
MAC   Maa,Max     ;MAC Ah[1,1]*x1h
MAC   Maa+2,Max+2 ;MAC Ah[1,2]*x2h
MAC   Maa+8,Mau   ;MAC Bh[1]*Uh
APAC  ;           ;Suma el ultimo producto al acumulador
Cmff  MTRX+1     ;almacenalo como MQ30 en una locacion auxiliar
;Para las operaciones de la segunda fila
MAC   Maa+4,Max+1 ;Multiplica Ah[2,1]*x1l
LAC   #0h         ;Limpia el acumulador
MAC   Maa+5,Max   ;MAC Al[2,1]*x1h

```

## APENDICE B: CODIGO DEL PROGRAMA

```

MAC      Maa+6,Max+3      ;MAC Ah[2,2]*x2l
MAC      Maa+7,Max+2      ;MAC Al[2,2]*x2h
MAC      Maa+10,Mau+1     ;MAC Bh[2]*U1
MAC      Maa+11,Mau       ;MAC Bl[2]*Uh
APAC                               ;Suma el ultimo producto al acumulador
SACH     MTRX              ;Almacena la parte alta de todo el producto
LAC      #0                ;En MTRX (auxiliar) y limpia el acumulador
LT       MTRX              ;y ponla en la parte baja del registro del
MPY      #1                ;producto con su signo correspondiente
MAC      Maa+4,Max        ;MAC Ah[2,1]*x1h
MAC      Maa+6,Max+2     ;MAC Ah[2,2]*x2h
MAC      Maa+10,Mau      ;MAC Bh[2]*Uh
APAC                               ;Suma el ultimo producto al acumulador
Cmfff    Max+2            ;almacenalo como MQ30 en x2
;Mueve los bloques de datos
BLDD     #MTRX+1,Max      ;Pasa de la locacion auxiliar de x1
BLDD     #MTRX+2,Max+1   ;al vector correspondiente m1
.endm

;Low pass filter en Q30, es una discretizacion de LPF, bastante precisa, tarda
;1.5 us en 29.4Mhz, aqui T es el periodo de muestreo, todo debe estar en MQ30
;  ___kp___*u
; tau*s+1 = x[k+1]= exp(-T/tau)*x[k] + kp*(1-exp(-T/tau))*u[k]
Delta .macro LP,Max,Mau
MAC      LP,Max+1         ;Aqui se hace la multiplicacion de los
LAC      #0h              ;productos entrecruzados
MAC      LP+1,Max        ;
MAC      LP+2,Mau+1     ;
MAC      LP+3,Mau       ;
APAC                               ;
SACH     MTRX              ;almacena el producto en una variable auxiliar
;Se hizo la multiplicacion de los productos intercalados
LAC      #0h              ;Limpia el acumulador
;Aqui se hace la multiplicacion de los operandos altos
LT       MTRX              ;inserta lo que haya en MTRX en PREG
MPY      #1                ;con su signo
MAC      LP,Max          ;Multiplicacion de los operandos altos o
MAC      LP+2,Mau       ;de los registros mas significativos
APAC                               ;
Cmfff    Max              ;almacenalo en Max[k] -> Max[k+1]
.endm

;Integrador en Q30, es una discretizacion de integradore, bastante precisa,
tarda
;1 us en 29.4Mhz, aqui T es el periodo de muestreo, todo debe estar en MQ30
;  _ki_ * u
; s      = x[k+1]= x[k] + ki*T*u[k]
Sigma .macro Sm,Max,Mau
MAC      Sm,Mau+1        ;Multiplica la entrada por ki*T y
LAC      #0h              ;acumula en la variable dada por Max
MAC      Sm+1,Mau        ;
APAC                               ;Es una multiplicacion para hacer la
SACH     MTRX              ;integral
LACQ     Max              ;x[k+1]= x[k] + ki*T*u[k]
LT       MTRX              ;
MPY      #1                ;
MAC      Sm,Mau          ;
APAC                               ;
Cmfff    Max              ;
.endm

```

## APENDICE B: CODIGO DEL PROGRAMA

;Multiplica dos variables en datos de Q30 y almacena el producto en una variable  
;Que tambien debe ser formato Q30

```
XQ30 .macro a,b,c
    LAC #0h ;Limpia el acumulador
    LT a+1 ;Pon a a+1 en TREG
    MPY b ;Multiplica por b
    LT a ;Pon a en TREG
    MPYA b+1 ;Multiplica por b+1
    APAC

    SACH MTRX
    LT MTRX
    MPY #1h
    LAC #0
    LT a
    MPYA b
    APAC
    Cmff c
.endm
```

;Macro para hacer la multiplicación de una constante de longitud  
;de 1 registro en memoria a=cte, b=registro, c=resultado

```
XQ15 .macro a,b,c
    LT a
    MPY b+1 ;Multiplica por b
    SPH MTRX
    LAC MTRX
    MPY b
    APAC
    Cmff c
.endm
```

;Macro para hacer la división de una constante de longitud (todo positivos)  
;de 1 registro en memoria a,b=registro c=resultado c=a/b

```
DQ16 .macro a,b,c
    LAC a ;registro a como dividendo
    RPT #14 ;el registro b es divisor
    SUBC b
    NOP ;la parte "entera" se almacena
    SACL c ;directamente en la MSByte de c
    AND #0FFFFh,16 ;Y se limpia la parte baja del acumulador
    ADD a+1,1 ;Se complementa con lo que sigue del dividendo
    RPT #14 ;Se hace 15 veces mas
    SUBC b ;Para la siguiente división
    NOP
    AND #07FFFh ;Solo nos interesa los bits 15-0 del acumulador
    SACL c+1 ;en c se tiene el resultado positivo
    LAC c
    AND #07FFFh ;Solo nos interesa los bits 15-0 del acumulador
    SACL c ;en c se tiene el resultado positivo
.endm
```

;ADCn --> Rn Trae desde el ADC los valores leidos

```
TADC: .macro Rgst
    LDP #6h ;
    LAR AR7,#RESULT0 ;Se apunta a RESULT0 con AR7
    MAR *,AR7 ;Este proceso se hace para traer
    RPT #6 ;Desde RESULT0 hasta R5 todos
    BLDD *+,#Rgst ;los resultados
.endm
```

## APENDICE B: CODIGO DEL PROGRAMA

```

;Convierte los registros del ADC en los valores terminales
CADC .macro areg,mul,creg,cor,off
    LACC areg,10 ;Se recorre el acumulador el rADC
    SUB #off,10 ;Se tiene el nivel de offset
    SACH creg,cor ;Se pone la variable en ci0 signada
    MAC mul,creg ;Se pone el multiplicador para convertir
    PAC ;a un factor de conversion
    Cmff creg ;Y se pone el valor en creg
.endm

;Conversion de todos los valores del ADC a sus formatos de 2^n en si
;todos son signados excepto el valor de Vc
Con2m: .macro aVc,ai0,aVs,aVr,air
    LDP #6h ;Pagina de todos los registros
;Se convierte Vc(491)*0.9599=cVc(512)
    LACC aVc,10 ;Se obtiene el Vc del ADC
    SACH cVc,5 ;Y se recorre 1 lugar a la derecha
    MAC bVc,cVc ;Se multiplica por el factor de
    PAC ;conversion =.9599 * 2^15
    SACH cVc ;Para una escala completa de 491V
;Se convierte i0(77.07692)*0.60216=ci0(1024)
    CADC ai0,bi0,ci0,3,7FC0h
;Se convierte Vs(260.704)*0.49132=cVs(1024)
    CADC aVs,bVs,cVs,5,7FC0h
;Se convierte Vcr(650)*valor=cVcr(16384)
    CADC aVr,bVr,cVr,2,7FC0h
;Se convierte ir(77.07692)*0.60216=cir(1024)
    CADC air,bir,cir,3,7FC0h
.endm

;Ecuaciones intermedias para m1 el numero Vc se convierte a
;formato 2^9, luego al elevarlo al cuadrado se pone en formato
;2^17 que es valor a la salida.
Eqm1: .macro
;Vc al cuadrado menos Vd al cuadrado
    SQRA cVc ;Se eleva Vc al cuadrado
    LACC #03D08h,15 ;Se pone al acumulador Vd^2 directamente
    ADD #04000h ;Para luego hacerlo negativo y efectuar
    NEG ;La resta
    APAC
    Cmff m1 ;Por el DSP, y listo
.endm

;Ecuacion intermedia para m2 que es un integrador
Eqm2: .macro
    sigma Im2,m2,m1 ;Integra con un factor de escala kiT
.endm

;Ecuacion para m3 que es un LPF
Eqm3: .macro
    Delta Lm3,m3,m1 ;Es un LPF de la forma 1/(Tau*s+1)
.endm

;Ecuacion de la suma m3 y m2 para formar m4 = rho
Eqm4: .macro
    LACQ m2 ;Se suman los productos de m2 y m3
    ADDQ m3 ;Para hacer rho
    Cmff m4 ;m4 esta en formato 2^10
.endm

;m5 = (1/a) (Vs-<Vs>-k<i0>)=Vr*

```



APENDICE B: CODIGO DEL PROGRAMA

```

Eqm5      .macro
pPA101    Ma11,MVs1,cVs      ;Se aplica la matriz <Vs>
pPA101    Ma11,Mi01,ci0      ;y tambien <i0(1024)>
XQ30      Mi01,m4,m51        ;es rho*<i0(1024)>
LACQ      cVs                ;Se pone Vs(1024) en el ACC
SUBQ      MVs1               ;Luego se le resta la fundamental de Vs
SUBQ      m51                ;En este punto se tiene (Vs-<Vs>-k<i0>)
Cmff      m52                ;Pon el resultado en m52
XQ15      RK1a,m52,m5        ;Guarda el resultado de Vr* en 2^14
.endm

;m6 = Vr - m5(k) = Vr - Vr*
Eqm6      .macro              ;Macro para la ecuacion 6
LACQ      cVr                ;Se pone en el acumulador Vr(16384)
SUBQ      m5                 ;Se resta Vr* (16384)
Cmff      m6                 ;El resultado se almacena en m6(16384)=Vr~
.endm

;m7 = e(k) = Vr - (k1+Egn)*m6 + k2*ir - k2*a*i0 - E N(m6)
Eqm7      .macro
XQ15      RK1,m6,m71         ;Multiplicacion de (k1+Egn)*m6=m71 (16384)
XQ15      RK2,cir,m72        ;Multiplica k2*ir = m72 (16384)
XQ15      RK2a,ci0,m73       ;Multiplica k2*a*i0 = m73 (16384)
pPA101    Ra11,NRa1,m6       ;Resonante para 60Hz          1
pPA101    Sa11,NRb1,m6       ;Resonante para 180Hz         3
pPA101    Ta11,NRc1,m6       ;Resonante para 300Hz         5
pPA101    Ua11,NRd1,m6       ;Resonante para 420Hz         7
pPA101    Va11,NRe1,m6       ;Resonante para 540Hz         9
pPA101    Wa11,NRf1,m6       ;Resonante para 660Hz        11
pPA101    Xa11,NRg1,m6       ;Resonante para 780Hz        13
LACQ      NRa2               ;Resta el resonante de 60Hz como LPF
ADDQ      NRb2               ;Resta el resonante de 180Hz como LPF
ADDQ      NRc2
ADDQ      NRd2
ADDQ      NRe2
ADDQ      NRf2
ADDQ      NRg2
Cmff      m74                ;La suma de los resonantes
XQ15      RKd,m74,m75        ;Se divide entre 32

LACQ      cVr                ;Pon en el acumulador a Vr
SUBQ      m71                ;Resta (k1+Egn)*m6=m71 (16384)
ADDQ      m72                ;Suma k2*ir = m72 (16384)
SUBQ      m73                ;Resta k2*a*i0 = m73 (16384)
ADDQ      m75                ;Suma el producto de los resonantes
Cmff      ek                 ;almacena ek en formato 2^14
.endm

Eqm8      .macro              ;m8 = u(k)=e(k)/cVc
LACQ      ek                 ;Pon en acumulador ek
ABS       ;Saca el valor absoluto
Cmff      aek                ;Ponlo en absolute ek
DQ16      aek,cVc,uk         ;Divide aek/Vc = uk
.endm

Eqm9      .macro              ;Normaliza el contenido del acumulador
LACQ      uk                 ;Pon a uk en el acumulador
SUB       #3FFh              ;Es mayor a 1 <2^21>?
BCND     E91,GT              ;Salta a E81 si es mayor
LACC     uk+1,6              ;Recorre en uk para el MSB
SACL     ukn                 ;Pon el valor en uk normalizado

```

## APENDICE B: CODIGO DEL PROGRAMA

```

B      E92      ;Vete a E82
E91: SPLK    #7FFFh,ukn      ;Pon el valor maximo a ukn
E92:      ;Sigue el codigo
      BIT      ek,0          ;Compara el signo de ek
      BCND    E93,NTC
      LACC    ukn,16        ;Pon uk en el acumulador
      NEG
      SACH    ukn+1
      B      E94
E93:      LACC    ukn,16        ;Pon uk en el acumulador
      SACH    ukn+1
E94:
      .endm

```

### B.3. Archivo de Inicialización: macrobib.asm

;Librería de macros de inicialización para la ezDSP 2407

```

Rdog      .macro          ;Macro para reinicializar el Watchdog
LDP      #00E0h          ;DP -> 7000h - 707Fh
SPLK     #05555h,WDKEY   ;Es una regla un 5555 seguido de
SPLK     #0AAAAh,WDKEY   ;un AAAA como seguro contra accidentes
LDP      #0h             ;DP -> 0000h - 007Fh
      .endm

Init      .macro          ;Macro para inicializar el sistema
LDP      #0h             ;DP = 0
SETC     INTM            ;Deshabilita las interrupciones
CLRC     SXM             ;Deshabilita la extensión de signo
SPLK     #0000h,IMR      ;Enmascara todas las int de núcleo
LACC     IFR             ;Lee las banderas de interrupción
SACL     IFR             ;Limpia todas las banderas de interrupción
LDP      #WDKEY>> 7h     ;(E0 = 224) (E0*80=7000)
SPLK     #006Fh,WDCR     ;Inhabilita el Perro Guardián
SPLK     #008Ch,SCSR1    ;Habilita el reloj al ADC, EVA, EVB
Rdog
      .endm

Timereg   .macro          ;Se habilitan los registros del EVM
LDP      #GPTCONA >> 7h ;Pagina para los EVM
SPLK     #00000h,GPTCONA ;
;Inicializa el timer 2
SPLK     #00000h,T2CNT   ;Reinicializa al Timer 2
SPLK     #00842h,T2CON   ;Inicializa el timer2 control reg
SPLK     #002E1h,T2PR    ;Se establece el periodo de muestreo
SPLK     #00170h,T2CMPR  ;a 50us
SPLK     #00004h,EVAIMRB ;Habilita Underflow Int del timer2
SPLK     #00004h,EVAIFRB ;Limpia las interrupciones
;Inicializamos el timer 1 para PWM
SPLK     #00000h,T1CNT   ;Cuenta el timer 1 desde cero
SPLK     #00842h,T1CON   ;Inicializa el timer1 control reg
SPLK     #002E1h,T1PR    ;Timer 1 Period register
;Para el PWM
SPLK     #00555h,ACTRA
SPLK     #08200h,COMCONA
SPLK     #00170h,T1CMPR  ;Timer 1 compare period
;Para habilitar la interrupcion en timer 1

```

## APENDICE B: CODIGO DEL PROGRAMA

```

    SPLK    #00001h,EVAIMRA ;Habilita PDPA
    SPLK    #00001h,EVAIFRA ;Limpia las interrupciones
;Para luego habilitar las interrupciones
    LDP     #0h
    SPLK    #07h,IMR          ;Habilita el nivel 1,2 y 3 de interrupciones
; que es la interrupción de comparación del Timer 1
    CLRC    INTM
    .endm

ADCreg .macro
;Para habilitar el ADC
    LDP     #0E1h              ;Pagina para el ADC
    SPLK    #4000h,ADCTRL1     ;Resetea el modulo de ADC
    SPLK    #31D0h,ADCTRL1     ;Convierte de manera continua

    SPLK    #0004h,MAXCONV     ;Cinco conversiones
    SPLK    #3210h,CHSELSEQ1   ;Convierte el canal 0,1,2,3 y 4
    SPLK    #0004h,CHSELSEQ2

;Se inician las conversiones del ADC
    SPLK    #4000h,ADCTRL2     ;Resetea a SEQ1.
    SPLK    #2000h,ADCTRL2     ;Comienza la conversion
    .endm

```

### B.4. Registros de constantes y utilización de memoria: Registros.asm

```

;Registros auxiliares de proposito general
    .bss    MTRX,4             ;Son usados para almacenar resultados intermedios

;Registros mapeados de variables a medir
    .bss    Vs,1              ;El Vc del ADC
    .bss    i0,1              ;El i0 del ADC
    .bss    Vr,1              ;El Vs del ADC
    .bss    ir,1              ;El Vcr del ADC
    .bss    Vc,1              ;El icr del ADC

;Registros mapeados de variables a medir convertidas a formato
    .bss    cVc,1             ;El Vc convertido a formato 512 positivo
    .bss    ci0,2             ;El i0 convertido a formato ? signado
    .bss    cVs,2             ;El Vs convertido a formato ? signado
    .bss    cVr,2             ;El Vcr convertido a formato ? signado
    .bss    cir,2             ;El icr convertido a formato ? signado

;Paisajes intermedios de las variables
    .bss    ek,2              ;La variable para e[k]
    .bss    aek,2             ;Valor absoluto de ek
    .bss    uk,2              ;el resultado de e/Vc = u
    .bss    ukn,2             ;El valor normalizado de uk
    .bss    m1,2              ;Resultado de (Vc^2-Vd^2)/2
    .bss    m2,2              ;Resultado del integrador
    .bss    m3,2              ;Resultado del LPF
    .bss    m4,2              ;m2+m3 = rho
    .bss    m5,2              ;Vr estrella
    .bss    m51,2             ;es el producto k*<i0>
    .bss    m52,2             ;antes de multiplicar por 1/a
    .bss    m53,2             ;Rho por una cte menor que 1
    .bss    m6,2              ;Es Vr tilda
    .bss    m71,2             ;Registros usados para los resultados
    .bss    m72,2             ;del LPF de segundo orden

```

## APENDICE B: CODIGO DEL PROGRAMA

```

.bss    m73,2
.bss    m74,2
.bss    m75,2
.bss    pwm,2

;Registro para las matrices
.bss    Mi01,2    ;Fundamental de i0
.bss    Mi02,2
.bss    MVs1,2    ;Fundamental de Vs
.bss    MVs2,2

;Registro de los LPF de segundo orden
.bss    NRa1,2    ;60Hz, fundamental
.bss    NRa2,2
.bss    NRb1,2    ;180Hz, tercer armónico
.bss    NRb2,2
.bss    NRc1,2    ;300Hz, quinto armónico
.bss    NRc2,2
.bss    NRd1,2    ;420Hz, sexto armónico
.bss    NRd2,2
.bss    NRe1,2    ;540Hz, noveno armónico
.bss    NRe2,2
.bss    NRf1,2    ;660Hz, onceavo armónico
.bss    NRf2,2
.bss    NRg1,2    ;780Hz, treceavo armónico
.bss    NRg2,2

;Registro para las "constantes" almacenadas en RAM
.bss    RK1a,1    ;Constante igual a 1/a
.bss    RK1,1     ;Constante igual a k1+sigma(gn)
.bss    RK2a,1    ;Constante igual a a
.bss    RK2,1     ;Constante igual a k2
.bss    RKPWM,1   ;Constante para el PWM
.bss    RKd,1     ;Constante de mult
.bss    RKe,1

;Registros de constantes para el PMA

.data

;Integrador Ki = 2e-2, para m1
Im2     .word    #00004h, #018DFh

;Filtro paso-bajos Kp= 5e-2, Tau= 0.01
Lm3     .word    #07F5Ch, #048D2h
Bm3     .word    #0082Bh, #0757Ch

;Filtro paso-bajos Kp= 1, Tau= 0.025
LP      .word    #07FBEh, #043C6h
BP      .word    #00005h, #01D6Ah

;LPF Kp=1, Tau= 2.6526e-3
LP1     .word    #07FF9h, #03939h
BP1     .word    #00006h, #046C7h

;Registros de las matrices de conversion para el BPF

;A=1, Q=4, l=94.2477796078 para m5
Ma11    .word    #07FDBh, #0282Fh
Ma12    .word    #0FD96h, #05528h
Ma21    .word    #00269h, #02AD8h

```

APENDICE B: CODIGO DEL PROGRAMA

```
Ma22 .word #07FFAh, #01720h
Mb1 .word #0001Eh, #06EF1h
Mb2 .word #00000h, #0253Eh
```

;Filtro Resonante a 60Hz, Q=40, A=30

```
Ra11 .word # 07FEAh, # 05F13h
Ra12 .word # 0FD96h, # 0428Ah
Ra21 .word # 00269h, # 03D76h
Ra22 .word # 07FFAh, # 01702h
Rb1 .word # 001CFh, # 00E19h
Rb2 .word # 00004h, # 02EBEh
```

;Filtro resonante a 180Hz, Q=40, A=80

```
Sa11 .word # 07F9Dh, # 03090h
Sa12 .word # 0F8C5h, # 02792h
Sa21 .word # 0073Ah, # 0586Eh
Sa22 .word # 07FCBh, # 052C6h
Sb1 .word # 01697h, # 03457h
Sb2 .word # 000A3h, # 04D55h
```

;Filtro Resonante a 300Hz, Q=40, A=50

```
Ta11 .word # 07F21h, # 057C9h
Ta12 .word # 0F3F7h, # 071C7h
Ta21 .word # 00C08h, # 00E39h
Ta22 .word # 07F6Eh, # 05824h
Tb1 .word # 01E14h, # 0238Fh
Tb2 .word # 0016Bh, # 023A7h
```

;Filtro Resonante a 420Hz, Q=40, A=80

```
Ua11 .word # 07E77h, # 070D0h
Ua12 .word # 0EF30h, # 00038h
Ua21 .word # 010CFh, # 07FC8h
Ua22 .word # 07EE3h, # 03D9Bh
Ub1 .word # 03F0Bh, # 07F2Ch
Ub2 .word # 0042Ah, # 078FBh
```

;Filtro Resonante a 540Hz, Q=40, A=100

```
Va11 .word # 07DA0h, # 02014h
Va12 .word # 0EA6Fh, # 030BFh
Va21 .word # 01590h, # 04F41h
Va22 .word # 07E2Ah, # 0220Fh
Vb1 .word # 035E9h, # 04622h
Vb2 .word # 00496h, # 02ADBh
```

;Filtro Resonante a 660Hz, Q=40, A=100

```
Wa11 .word # 07C9Bh, # 0124Fh
Wa12 .word # 0E5B7h, # 05FADh
Wa21 .word # 01A48h, # 02053h
Wa22 .word # 07D43h, # 02CB8h
Wb1 .word # 041B4h, # 050D0h
Wb2 .word # 006D7h, # 05034h
```

;Filtro Resonante a 120Hz, Q=40, A=100

```
Xa11 .word # 07FC9h, # 06E87h
Xa12 .word # 0FB2Dh, # 04663h
Xa21 .word # 004D2h, # 0399Dh
Xa22 .word # 07FE8h, # 05CC5h
Xb1 .word # 00134h, # 04E67h
Xb2 .word # 00005h, # 068CFh
```

;Constantes conversión para los valores leídos

## APENDICE B: CODIGO DEL PROGRAMA

```
bVc    .word    #07ADFh
bi0    .word    #04D27h
bVs    .word    #0413Dh
bVr    .word    #05154h
bir    .word    #04D27h
```

;Constantes de uso para el algoritmo

```
K1a    .word    #05000h           ;Cte  $1/a = 10 = 0.625 \times 2^4$  10->14
K1     .word    #05000h           ;Cte igual a  $k1=0.25$  10->14
K2a    .word    #00800h           ;Cte  $a = .1 = .8 \times 2^{-3}$  10->14
K2     .word    #05000h           ;Cte igual a  $K2 = 10$ 
PWM    .word    #005C2h           ;Cte de multiplicacion para el PWM
Kd     .word    #00400h           ;Cte igual
Ke     .word    #07000h
```

APENDICE B: CODIGO DEL PROGRAMA

## APENDICE C

---

### DESCRIPCION DE LOS MACROS UTILIZADOS

*Este apéndice contiene todas los macros usadas para el programa del controlador. Estos macros están clasificadas de la siguiente manera:*

- 1. Macros de inicialización del sistema*
- 2. Macros del manejo de memoria de programa, RAM y puertos*
- 3. Macros para realizar operaciones con memoria de programa y de datos*
- 4. Macros del código del controlador.*

*Para cada una de ellas se presenta una breve explicación de su funcionamiento.*

#### C.1 Macros de inicialización del sistema.

##### **ADCreg**

Este macro habilita características de la conversión del ADC como la conversión ininterrumpida y la conversión en secuencia de los canales 0, 1, 2, 3, 4 de manera repetitiva.

```
ADCreg .macro
;Para habilitar el ADC
    LDP    #0E1h                ;Pagina para el ADC
    SPLK   #4000h,ADCTRL1       ;Resetea el modulo de ADC
    SPLK   #31D0h,ADCTRL1       ;Convierte de manera continua
    SPLK   #0004h,MAXCONV        ;Cinco conversiones
    SPLK   #3210h,CHSELSEQ1      ;Convierte el canal 0,1,2,3 y 4
    SPLK   #0004h,CHSELSEQ2
    SPLK   #4000h,ADCTRL2       ;Resetea a SEQ1.
```



## APENDICE C: DESCRIPCION DE LAS MACROS UTILIZADAS

```
SPLK    #2000h,ADCTRL2    ;Comienza la conversion
.endm
```

### Clrblk

Este macro permite establecer las condiciones iniciales para los registros de la página de trabajo iguales a cero. La dirección de inicio se carga en el registro auxiliar 7 (AR7), y por medio de direccionamiento indirecto. Usando este registro auxiliar es posible modificar secuencialmente los valores de los registros direccionados.

```
Clrblk .macro
LAR     AR7,#300h        ;Carga 300h en AR7 (Apuntador)
MAR     *,AR7           ;Cambia el ARP a AR7
LAC     #0h             ;Carga el acumulador con cero
RPT     #0FFh          ;repite 256 veces la operacion, de cargar
SACL    *+             ;El cero al acumulador incrementando AR7
.endm    ;Cada vez
```

### Init

Macro para la inicialización del sistema. Deshabilita las posibles interrupciones erróneas al encendido del sistema. Además, habilita el uso de los temporizadores, el ADC y establece el multiplicador de la frecuencia de reloj del sistema al doble de la frecuencia entregada por el cristal externo.

```
Init .macro ;Macro para inicializar el sistema
LDP     #0h ;DP = 0
SETC    INTM ;Deshabilita las interrupciones
CLRC    SXM ;Deshabilita la extensión de signo
SPLK    #0000h,IMR ;Enmascara todas las int de núcleo
LACC    IFR ;Lee las banderas de interrupción
SACL    IFR ;Limpia todas las banderas de interrupción
LDP     #WDKEY>> 7h ;(E0 = 224) (E0*80=7000)
SPLK    #006Fh,WDCR ;Inhabilita el Guardián
SPLK    #008Ch,SCSR1 ;Habilita el reloj al ADC, EVA, EVB
Rdog
.endm
```

### MKPD

Este macro tiene como función mover todas las constantes de memoria de programa a memoria de datos (RAM). Esto se hace con el fin de agilizar el tiempo de ejecución de algunas multiplicaciones.

```
MKPD .macro
LAR     AR7,#RK1a ;Mueve al Registro de Datos
RPT     #06h ;Repite siete veces
BLPD    #K1a,*+ ;Mueve las constantes hacia memoria
.endm
```

### Puertos

Inicializa el valor de las salidas analógicas a cero, a fin de establecer condiciones iniciales en el encendido o la reinicialización.

```
Puertos .macro
LDP     #0E1h ;Pagina de los puertos
```

## APENDICE C: DESCRIPCION DE LAS MACROS UTILIZADAS

```

SPLK    #03F00h,MCRA      ;solo Puerto A, PB6 y PB7 como IO
SPLK    #00000h,MCRC      ;Puerto E como salida
SPLK    #0FF00h,MCRB      ;Puerto C como salida
SPLK    #0FF80h,PADATDIR  ;Puerto A como salidas
SPLK    #0FF80h,PCDATDIR  ;Puerto C como salidas
SPLK    #0FF80h,PEDATDIR  ;Puerto E como salidas
.endm

```

### Rdog

Este macro limpia el contador del guardián, quien es el encargado de vigilar que el DSP no caiga en lazos infinitos.

```

Rdog    .macro              ;Macro para reinicializar el Watchdog
LDP     #00E0h              ;DP -> 7000h - 707Fh
SPLK    #05555h,WDKEY       ;Es una regla un 5555 seguido de
SPLK    #0AAAAh,WDKEY       ;un AAAA como seguro contra accidentes
LDP     #0h                  ;DP -> 0000h - 007Fh
.endm

```

### Timereg

Habilita los registros del Temporizador 1 y el 2. Establece y habilita interrupción por periodo del Temporizador 2, dando como tiempo de muestreo 50µs. Notar que, el código que configura el Temporizador 1 habilita el uso del PWM, el cual, sin embargo, no es utilizado en el presente proyecto.

```

Timereg .macro              ;Se habilitan los registros del EVM
LDP     #GPTCONA >> 7h     ;Pagina para los EVM
SPLK    #00000h,GPTCONA ;
;Inicializa el timer 2
SPLK    #00000h,T2CNT      ;Reinicializa al Timer 2
SPLK    #00842h,T2CON      ;Inicializa el timer2 control reg
SPLK    #002E1h,T2PR       ;Se establece el periodo de muestreo
SPLK    #00170h,T2CMPR     ;a 50us
SPLK    #00004h,EVAIMRB    ;Habilita Underflow Int del timer2
SPLK    #00004h,EVAIFRB    ;Limpia las interrupciones
;Inicializamos el timer 1 para PWM
SPLK    #00000h,T1CNT      ;Cuenta el timer 1 desde cero
SPLK    #00842h,T1CON      ;Inicializa el timer1 control reg
SPLK    #002E1h,T1PR       ;Timer 1 Period register
;Para el PWM
SPLK    #00555h,ACTRA
SPLK    #08200h,COMCONA
SPLK    #00170h,T1CMPR     ;Timer 1 compare period
;Para habilitar la interrupcion en timer 1
SPLK    #00001h,EVAIMRA    ;Habilita PDPA
SPLK    #00001h,EVAIFRA    ;Limpia las interrupciones
;Para luego habilitar las interrupciones
LDP     #0h
SPLK    #07h,IMR           ;Habilita el nivel 1,2 y 3 de interrupciones
; que es la interrupción de comparación del Timer 1
CLRC    INTM
.endm

```

## C.2 Macros del manejo de memoria de Programa, RAM y Puertos.

### DAC

Manda la variable originalmente en formato de punto fijo de 32 bits a una salida analógica en el formato apropiado para el DAC externo.

```
DAC      .macro reg,puerto,n      ;
LACC    reg,n+8      ;Carga el registro en el acumulador
ADD     #08000h,8    ;suma con para hacer el producto para el DAC
OR      #0FF00h,16   ;Sacalo como puertos del DAC
LDP     #0E1h        ;Pagina de el DAC y los Puertos
SACH    puerto       ;Expulsa el resultado del HACC en el puerto
LDP     #06h         ;Regresa a la pagina de los registros de trabajo
.endm
```

### Puerto

Muestra por el puerto especificado un valor entre # 0h y # FFh dado por el parámetro *valor*. Este macro es especialmente útil cuando se realiza la depuración del programa.

```
Puerto  .macro valor,puerto
LDP     #0E1h        ;Pagina de los puertos
SPLK    #valor,puerto ;Puerto = #valor
LDP     #06h         ;Regresa a al pagina de trabajo
.endm
```

### TADC

Macro para copiar los registros (RESULT0... RESULT5) hacia la pagina de memoria de datos donde están los registros de trabajo.

```
TADC:   .macro Rgst
LDP     #6h          ;
LAR     AR7,#RESULT0 ;Se apunta a RESULT0 con AR7
MAR     *,AR7        ;Este proceso se hace para traer
RPT     #6           ;Desde RESULT0 hasta R5 todos
BLDD   *+,#Rgst     ;los resultados
.endm
```

## C.3 Macros para realizar operaciones con memoria de programa y de datos.

### ADDQ

Suma la variable de formato de punto fijo de 32 bits especificada por el argumento *reg* al acumulador.

```
ADDQ    .macro reg
ADD     reg+1,1      ;Suma a la parte baja al LACC
ADD     reg,16       ;Suma a la parte alta al HACC
.endm
```

## CADC

Convierte los valores de la señal medida provenientes del ADC a su valor real y los deja en el formato de punto fijo de 32 bits.

```
CADC .macro areg,mul,creg,cor,off
LACC areg,10 ;Se recorre el acumulador el rADC
SUB #off,10 ;Se tiene el nivel de offset
SACH creg,cor ;Se pone la variable en ci0 signada
MAC mul,creg ;Se pone el multiplicador para convertir
PAC ;a un factor de conversion
Cmff creg ;Y se pone el valor en creg
.endm
```

## Cmff

Convierte la variable en el acumulador al formato de punto fijo de 32 bits para salvarla a un registro.

```
Cmff: .macro nxq ;
SACH nxq ;Almacena la parte alta del acumulador en nxq
SFR ;Recorre a la derecha para eliminar el signo
AND #7FFFh ;del LACC por medio de una operacion AND
SACL nxq+1 ;almacena el resto en nxq
.endm
```

## Con2m:

Las siguientes líneas de código realizan la conversión de formato de todos los valores provenientes de la conversión del ADC a valores en formato de punto fijo. Recordar que, todos los valores proveniente dicha conversión son signados, a excepción de  $V_c$ , el cual es siempre positivo. Los valores de los resultados se almacenan en los registros llamados  $cVc$ ,  $ci0$ ,  $cVs$ ,  $cir$ , y en el caso de  $V_{cr}$ , se almacena en  $cVr$ .

```
Con2m: .macro aVc,ai0,aVs,aVr,air
LDP #6h ;Pagina de todos los registros
;Se convierte  $V_c(491) * 0.9599 = cVc(512)$ 
LACC aVc,10 ;Se obtiene el  $V_c$  del ADC
SACH cVc,5 ;Y se recorre 1 lugar a la derecha
MAC bVc,cVc ;Se multiplica por el factor de
PAC ;conversion = .9599 * 2^15
SACH cVc ;Para una escala completa de 491V
;Se convierte  $i0(77.07692) * 0.60216 = ci0(1024)$ 
CADC ai0,bi0,ci0,3,7FC0h
;Se convierte  $V_s(260.704) * 0.49132 = cVs(1024)$ 
CADC aVs,bVs,cVs,5,7FC0h
;Se convierte  $V_{cr}(650) * valor = cVcr(16384)$ 
CADC aVr,bVr,cVr,2,7FC0h
;Se convierte  $ir(77.07692) * 0.60216 = cir(1024)$ 
CADC air,bir,cir,3,7FC0h
.endm
```

## Delta

Implementa a un LPF de primer orden que se usa para el calculo de  $\rho$  y tiene como entrada a  $\tilde{z}$ . Su tiempo de ejecución es 1.5 $\mu$ s.

## APENDICE C: DESCRIPCION DE LAS MACROS UTILIZADAS

```

Delta .macro LP,Max,Mau
    MAC    LP,Max+1    ;Aqui se hace la multiplicacion de los
    LAC    #0h        ;productos entrecruzados
    MAC    LP+1,Max    ;
    MAC    LP+2,Mau+1 ;
    MAC    LP+3,Mau    ;
    APAC                   ;
    SACH    MTRX        ;almacena el producto en una variable auxiliar
;Se hizo la multiplicacion de los productos intercalados
    LAC    #0h        ;Limpia el acumulador
;Aqui se hace la multiplicacion de los operandos altos
    LT     MTRX        ;inserta lo que haya en MTRX en PREG
    MPY    #1          ;con su signo
    MAC    LP,Max      ;Multiplicacion de los operandos altos o
    MAC    LP+2,Mau    ;de los registros mas significativos
    APAC                   ;
    Cmff   Max         ;almacenalo en Max[k] -> Max[k+1]
.endm

```

### DQ16

Macro que realiza la división un número en punto fijo positivo de 32 bits por un número en punto fijo positivo de 16 bits (solo se toma el registro mas significativo), y el resultado se almacena en un número de punto fijo especificado por el argumento *c*. Notar que si uno de los operandos es negativo la división dará por resultado un valor erróneo.

```

DQ16 .macro a,b,c
    LAC    a           ;registro a como dividendo
    RPT    #14        ;el registro b es divisor
    SUBC   b
    NOP                   ;la parte "entera" se almacena
    SACL   c           ;directamente en la MSByte de c
    AND    #0FFFFh,16 ;Y se limpia la parte baja del acumulador
    ADD    a+1,1      ;Se complementa con lo que sigue del dividendo
    RPT    #14        ;Se hace 15 veces mas
    SUBC   b           ;Para la siguiente división
    NOP
    AND    #07FFFh    ;Solo nos interesa los bits 15-0 del acumulador
    SACL   c+1        ;en c se tiene el resultado positivo
    LAC    c
    AND    #07FFFh    ;Solo nos interesa los bits 15-0 del acumulador
    SACL   c           ;en c se tiene el resultado positivo
.endm

```

### LACQ

Carga la variable de punto fijo de 32 bits especificada por el argumento *reg* al acumulador.

```

LACQ .macro reg
    LAC    reg,16     ;Carga la parte alta al HACC
    ADD    reg+1,1    ;Suma la parte baja al LACC
.endm

```

### pPA101

Macro que realiza un sistema de 2º orden previamente discretizado. Este macro se utiliza para implementar los BPF's que extraen las componentes fundamentales de las

señales  $i_0$  y  $V_s$ , así como los LPF de segundo orden. La ejecución de este código tarda  $3.25\mu s$ .

```
pPA101 .macro Maa,Max,Mau ;SPM = 1
;Para las operaciones de la primera fila
MAC Maa,Max+1 ;Multiplica Ah[1,1]*x1l
LAC #0h ;Limpia el acumulador
MAC Maa+1,Max ;MAC Al[1,1]*x1h
MAC Maa+2,Max+3 ;MAC Ah[1,2]*x2l
MAC Maa+3,Max+2 ;MAC Al[1,2]*x2h
MAC Maa+8,Mau+1 ;MAC Bh[1]*U1
MAC Maa+9,Mau ;MAC Bl[1]*Uh
APAC ;Suma el ultimo producto al acumulador
SACH MTRX ;Almacena la parte alta de todo el producto
LAC #0 ;En MTRX (auxiliar) y limpia el acumulador
LT MTRX ;y ponla en la parte baja del registro del
MPY #1 ;producto con su signo correspondiente
MAC Maa,Max ;MAC Ah[1,1]*x1h
MAC Maa+2,Max+2 ;MAC Ah[1,2]*x2h
MAC Maa+8,Mau ;MAC Bh[1]*Uh
APAC ;Suma el ultimo producto al acumulador
Cmff MTRX+1 ;almacénalo como formato de pf en un reg auxiliar
;Para las operaciones de la segunda fila
MAC Maa+4,Max+1 ;Multiplica Ah[2,1]*x1l
LAC #0h ;Limpia el acumulador
MAC Maa+5,Max ;MAC Al[2,1]*x1h
MAC Maa+6,Max+3 ;MAC Ah[2,2]*x2l
MAC Maa+7,Max+2 ;MAC Al[2,2]*x2h
MAC Maa+10,Mau+1 ;MAC Bh[2]*U1
MAC Maa+11,Mau ;MAC Bl[2]*Uh
APAC ;Suma el ultimo producto al acumulador
SACH MTRX ;Almacena la parte alta de todo el producto
LAC #0 ;En MTRX (auxiliar) y limpia el acumulador
LT MTRX ;y ponla en la parte baja del registro del
MPY #1 ;producto con su signo correspondiente
MAC Maa+4,Max ;MAC Ah[2,1]*x1h
MAC Maa+6,Max+2 ;MAC Ah[2,2]*x2h
MAC Maa+10,Mau ;MAC Bh[2]*Uh
APAC ;Suma el ultimo producto al acumulador
Cmff Max+2 ;almacenalo como formato de pf en x2
;Mueve los bloques de datos
BLDD #MTRX+1,Max ;Pasa de la locacion auxiliar de x1
BLDD #MTRX+2,Max+1 ;al vector correspondiente m1
.endm
```

## Sigma

Macro que integra a la señal  $\tilde{z}$ . Representa la parte integral del controlador para el calculo de  $\rho$ . Este macro tarda en ejecutarse  $1\mu s$ .

```
Sigma .macro Sm,Max,Mau
MAC Sm,Mau+1 ;Multiplica la entrada por ki*T y
LAC #0h ;acumula en la variable dada por Max
MAC Sm+1,Mau ;
APAC ;Es una multiplicación para hacer la
SACH MTRX ;integral
LACQ Max ;x[k+1]= x[k] + ki*T*u[k]
LT MTRX ;
MPY #1 ;
```

## APENDICE C: DESCRIPCION DE LAS MACROS UTILIZADAS

```
MAC      Sm,Mau      ;
APAC                                          ;
Cmfff    Max         ;
.endm
```

### SUBQ

Resta la variable, en formato de punto fijo de 32 bits, especificada por el argumento *reg* del acumulador.

```
SUBQ     .macro      reg
SUB      reg+1,1     ;Resta a la parte baja al LACC
SUB      reg,16      ;Resta a la parte alta al HACC
.endm
```

### XQ15

Macro que realiza una multiplicación de un número en formato de punto fijo *a* por una constante *b* de longitud 16 bits en memoria de datos, almacenando el resultado en el registro *c* especificado.

```
XQ15     .macro      a,b,c
LT       a
MPY      b+1         ;Multiplica por b
SPH      MTRX
LAC      MTRX
MPY      b
APAC
Cmfff    c
.endm
```

### XQ30

Es el macro que realiza la multiplicación de dos números en formato de punto fijo, *a* y *b*, de 32 bits cada uno, y almacena el resultado en el registro *c* especificado.

```
XQ30     .macro      a,b,c
LAC      #0h        ;Limpia el acumulador
LT       a+1        ;Pon a a+1 en TREG
MPY      b           ;Multiplica por b
LT       a           ;Pon a en TREG
MPYA     b+1        ;Multiplica por b+1
APAC
SACH     MTRX
LT       MTRX
MPY      #1h
LAC      #0
LT       a
MPYA     b
APAC
Cmfff    c
.endm
```

## C.4 Macros del código del controlador.

Los siguientes macros contienen el código de programa para la implementación de las ecuaciones (3.49) a (3.56), esto es, una macro por ecuación. Notar que se ha dado nombre a cada macro de acuerdo al nombre de la variable de salida de cada ecuación.

### Eqm1 (ecuación (3.49))

Calcula  $\tilde{z}$ , para ello eleva al cuadrado  $V_c$ , y le resta el valor del cuadrado de la referencia  $V_d^2$ . El resultado de esta operación se almacena en  $m1$ .

```
Eqm1: .macro
;Vc al cuadrado menos Vd al cuadrado
  SQRA    cVc          ;Se eleva Vc al cuadrado
  LACC    #03D08h,15   ;Se pone al acumulador Vd^2 directamente
  ADD     #04000h      ;Para luego hacerlo negativo y efectuar
  NEG
  APAC
  Cmff    m1           ;Por el DSP, y listo
.endm
```

### Eqm2 (ecuación (3.50))

En la siguiente línea se llama a el macro **sigma** quien integra el valor de  $\tilde{z}$  como. Este término integral forma parte del calculo del parámetro  $\rho$ . El resultado se almacena en  $m2$ .

```
Eqm2: .macro
  sigma   Im2,m2,m1    ;Integra con un factor de escala kiT
.endm
```

### Eqm3 (ecuación (3.51))

En la siguiente línea, se llama a el macro **delta** para aplicar un LPF a  $\tilde{z}$ . Esto forma parte del cálculo del parámetro  $\rho$ . El resultado se almacena en  $m3$ .

```
Eqm3: .macro
  Delta   Lm3,m3,m1    ;Es un LPF de la forma 1/(Tau*s+1)
.endm
```

### Eqm4 (ecuación (3.52))

Este macro calcula el parámetro  $\rho$ . Para ello, suma los resultados provenientes de las 2 macros anteriores. El resultado se almacena en  $m4$ .

```
Eqm4: .macro
  LACQ    m2           ;Se suman los productos de m2 y m3
  ADDQ    m3
  Cmff    m4           ;m4 esta en formato 2^10
.endm
```

### Eqm5 (ecuación (3.53))

Este macro calcula el valor de  $V_r^*$ . Se obtiene la fundamental de las señales  $V_s$  e  $i_0$  por medio de la implementación de BPF a través de las macros **pPA101**. Estas fundamentales



se almacenan en  $MVs1$  y  $Mi01$ . Se multiplica el valor del parámetro  $\rho$  por la fundamental de  $i_0$ , y el resultado se almacena en  $m51$ . Se realiza la operación,  $V_s - V_s^1 - \rho i_0^1$  por medio de  $cVs - MVs1 - m4 * Mi01$  el cual se guarda en  $m52$ . Finalmente, se multiplica  $m52$  por el valor de  $1/a$  y el resultado de esto se guarda en  $m5$ , que es el registro que contiene al valor de  $V_r^*$ .

```
Eqm5      .macro
pPA101    Ma11,MVs1,cVs      ;Se aplica la matriz <Vs>
pPA101    Ma11,Mi01,ci0      ;y tambien <i0(1024)>
XQ30      Mi01,m4,m51        ;es rho*<i0(1024)>
LACQ      cVs                ;Se pone Vs(1024) en el ACC
SUBQ      MVs1               ;Luego se le resta la fundamental de Vs
SUBQ      m51                ;En este punto se tiene (Vs-<Vs>-k<i0>)
Cmff      m52                ;Pon el resultado en m52
XQ15      RK1a,m52,m5        ;Guarda el resultado de Vr* en 2^14
.endm
```

### Eqm6 (ecuación (3.54))

Obtiene el valor de  $\tilde{V}_r$ . Para esto resta  $m5$  de  $cVr$  y almacena el resultado en  $m6$ .

```
Eqm6      .macro                ;Macro para la ecuacion 6
LACQ      cVr                  ;Se pone en el acumulador Vr(16384)
SUBQ      m5                    ;Se resta Vr* (16384)
Cmff      m6                    ;El resultado se almacena en m6(16384)=Vr~
.endm
```

### Eqm7 (ecuación (3.55))

Este macro calcula el valor de la señal de control auxiliar  $e$ . Primeramente se calcula el valor de los términos intermedios  $(\sum_k \gamma_k + k_1) \tilde{V}_r$ ,  $k_2 i_r$  y  $a k_2 i_0$  almacenando estos resultados en  $m71$ ,  $m72$  y  $m73$ , respectivamente. Las siguientes líneas corresponden al cálculo de los LPF de 2° orden implementados en el programa, así como la sumatoria de los resultados de estos. Cabe resaltar el hecho de, con base en la capacidad del DSP, se puede compensar hasta el 13° armónico. El resultado de la anterior suma se almacena en  $m74$ . Finalmente, se realiza la suma de los términos anteriores y el resultado se almacena en el registro  $ek$ .

```
Eqm7      .macro
XQ15      RK1,m6,m71            ;Multiplicación de (k1+Egn)*m6=m71 (16384)
XQ15      RK2,cir,m72           ;Multiplica k2*ir = m72 (16384)
XQ15      RK2a,ci0,m73          ;Multiplica k2*a*i0 = m73 (16384)
pPA101    Ra11,NRa1,m6           ;Resonante para 60Hz          1
pPA101    Sa11,NRb1,m6           ;Resonante para 180Hz         3
pPA101    Ta11,NRc1,m6           ;Resonante para 300Hz         5
pPA101    Ua11,NRd1,m6           ;Resonante para 420Hz         7
pPA101    Va11,NRe1,m6           ;Resonante para 540Hz         9
pPA101    Wa11,NRf1,m6           ;Resonante para 660Hz        11
pPA101    Xa11,NRg1,m6           ;Resonante para 780Hz        13
LACQ      NRa2                  ;Resta el resonante de 60Hz como LPF
ADDQ      NRb2                  ;Resta el resonante de 180Hz como LPF
ADDQ      NRc2
ADDQ      NRd2
```

APENDICE C: DESCRIPCION DE LAS MACROS UTILIZADAS

```

ADDQ    NRe2
ADDQ    NRf2
ADDQ    NRg2
Cmff    m74                ;La suma de los resonantes
XQ15    RkD,m74,m75        ;Se divide entre 32
LACQ    cVr                ;Pon en el acumulador a Vr
SUBQ    m71                ;Resta (k1+Egn)*m6=m71 (16384)
ADDQ    m72                ;Suma k2*ir = m72 (16384)
SUBQ    m73                ;Resta k2*a*i0 = m73 (16384)
ADDQ    m75                ;Suma el producto de los resonantes
Cmff    ek                ;almacena ek en formato 2^14
.endm

```

### Eqm8

El macro Eqm8 realiza la división  $e/V_c$  obteniendo como resultado la salida de control  $u$ . Notar que se obtiene el valor absoluto de  $e$  para garantizar que los operandos en la división sean positivos.

```

Eqm8    .macro    ;m8 = u(k)=e(k)/cVc
LACQ    ek                ;Pon en acumulador ek
ABS     ek                ;Saca el valor absoluto
Cmff    aek                ;Ponlo en absolute ek
DQ16    aek,cVc,uk        ;Divide aek/Vc = uk
.endm

```

### Eqm9

Este macro limita el valor  $u$  al rango de -1 a 1. En el caso en que este controlador se utilice para armar un prototipo físico, en este macro se declara el ciclo de trabajo de las salidas de PWM por medio de los registros de comparación (CMPR),

```

Eqm9    .macro    ;Normaliza el contenido del acumulador
LACQ    uk                ;Pon a uk en el acumulador
SUB     #3FFh              ;Es mayor a 1 <2^21>?
BCND    E91,GT            ;Salta a E81 si es mayor
LACC    uk+1,6            ;Recorre en uk para el MSB
SACL    ukn                ;Pon el valor en uk normalizado
B       E92                ;Vete a E82
E91:    SPLK    #7FFFh,ukn ;Pon el valor maximo a ukn
E92:    BIT     ek,0        ;Compara el signo de ek
BCND    E93,NTC
LACC    ukn,16            ;Pon uk en el acumulador
NEG
SACH    ukn+1
B       E94
E93:    LACC    ukn,16        ;Pon uk en el acumulador
SACH    ukn+1
E94:    .endm

```



## APENDICE D

---

### PUERTOS Y TARJETAS DE INTERFAZ

*En este apéndice se hace una revisión de los conectores de la dSPACE y las terminales de la tarjeta eZDSP320LF2407A. También se presentan las tarjetas de interfaz usadas para la interconexión de ambas tarjetas.*

#### **D.1. TARJETA dSPACE RTI-1103**

El panel de conector CP1103 proporciona una conexión fácil de usar entre la tarjeta dSPACE RTI-1103 y los dispositivos que se conectan a esta. Los dispositivos pueden ser individualmente conectados, desconectados o intercambiados sin la necesidad de ser soldados. Esta característica simplifica la puesta en marcha del sistema.

Se muestra en la Tabla D.1 los conectores BNC hembra, en donde la coraza está conectada a tierra GND (CP1...CP28) como se muestra en la Fig. D.1.

**Tabla D.1.**  
Señales proporcionadas por el conector CPL1103.

<i>Conector</i>	<i>Señal</i>	<i>Conector</i>	<i>Señal</i>
CP1	ADCH1	CP15	ADCH15
CP2	ADCH2	CP16	ADCH16
CP3	ADCH3	CP17	ADCH17
CP4	ADCH4	CP18	ADCH18
CP5	ADCH5	CP19	ADCH19
CP6	ADCH6	CP20	ADCH20
CP7	ADCH7	CP21	DACH21
CP8	ADCH8	CP22	DACH22
CP9	ADCH9	CP23	DACH23
CP10	ADCH10	CP24	DACH24
CP11	ADCH11	CP25	DACH25
CP12	ADCH12	CP26	DACH26
CP13	ADCH13	CP27	DACH27
CP14	ADCH14	CP28	DACH28

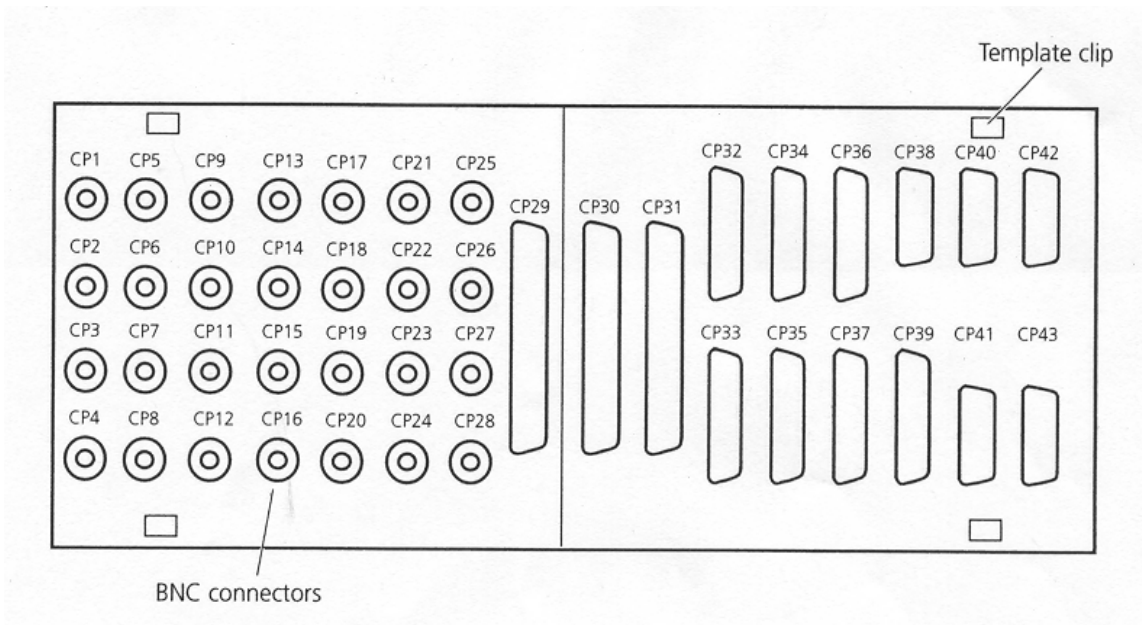


Fig. D.1 Conector analógico dado por la tarjeta dSPACE DS1103.

## D.2. TARJETA DE EVALUACION DEL DSP

La tarjeta eZDSP320LF2407A mostrada en la Fig. D.2. es un sistema de evaluación del DSP TMS320LF2407A (procesador principal). Los pines del DSP son accesibles gracias a conectores especiales dispuestos en la tarjeta eZDSP320LF2407A.

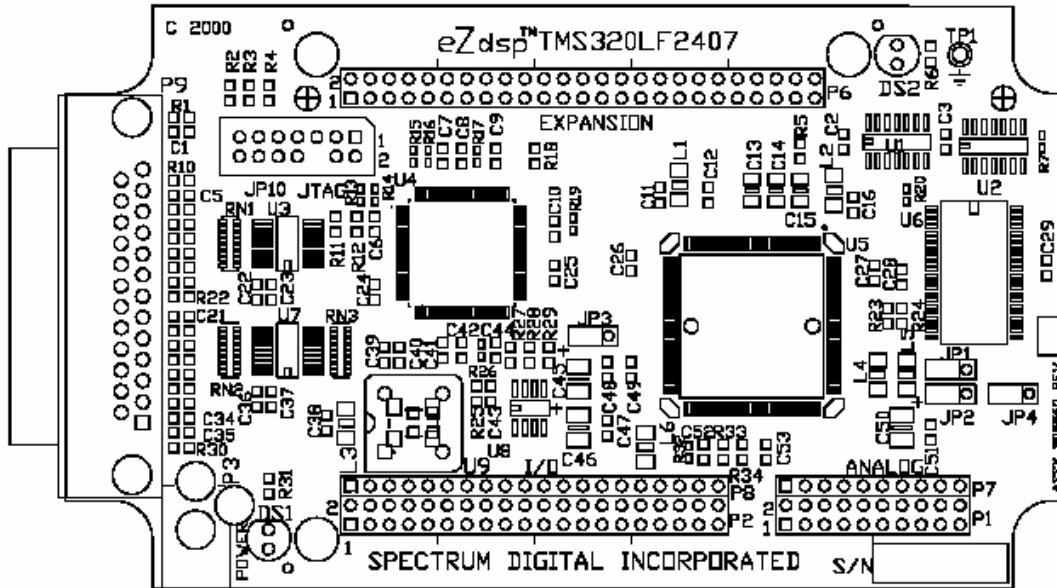


Fig. D.2 Tarjeta eZDSP320LF2407A.

En las siguientes tablas, Tabla D.2 y Tabla D.3, se muestran las señales asignadas a los conectores analógico y digital de los puertos de la tarjeta de evaluación utilizados en este proyecto.

**Tabla D.2**  
Conector de la interfaz analógica P1.

<i>Terminal</i>	<i>Señal</i>	<i>Terminal</i>	<i>Señal</i>
1	GND	2	ADCIN0
3	GND	4	ADCIN1
5	GND	6	ADCIN2
7	GND	8	ADCIN3
9	GND	10	ADCIN4
11	GND	12	ADCIN5
13	GND	14	ADCIN6
15	GND	16	ADCIN7
17	GND	18	VREFLO
19	GND	20	VREFHI

**Tabla D.3**  
Conector de la interfaz digital del puerto P2.

<i>Terminal</i>	<i>Señal</i>	<i>Terminal</i>	<i>Señal</i>
1	+5V	2	+5V
3	SCITXD/IOPA0	4	SCIRXD/IOPA1
5	XINT1-/IOPA2	6	CAP1/QEP1/IOPA3
7	CAP2/QEP2/IOPA4	8	CAP3/IOPA5
9	PWM1/IOPA6	10	PWM2/IOPA7
11	PWM3/IOPB0	12	PWM4/IOPB1
13	PWM5/IOPB2	14	PWM6/IOPB3
15	T1PWM/T1CMP/IOPB4	16	T2PWM/T2CMP/IOPB5
17	TDIRA/IOPB6	18	TCLKINA/IOPB7
19	GND	20	GND
21	WNR/IOPC0	22	BIO-/IOPC1
23	SPISIMO/IOPC2	24	SPISOMI/IOPC3
25	SPICLK/IOPC4	26	SPISTE/IOPC5
27	CANTX/IOPC6	28	CANRX/IOPC7
29	CLKOUT/IOPE0	30	PWM7/IOPE1
31	PWM8/IOPE2	32	PWM9/IOPE3
33	PWM10/IOPE4	34	PWM11/IOPE5
35	PWM12/IOPE6	36	CAP4/QEP3/IOPE7
37	PDPINTA-	38	PDPINTB-
39	GND	40	GND

### D.3. TARJETAS DE INTERFAZ

#### Limitadores de Voltaje.

Dado que la entrada a los puertos ADCs está limitada a un rango de 0V a 3.3V, según especificaciones de la TMS320LF2407A. Se diseñó una tarjeta de interfaz para limitar estos voltajes de entrada y así proteger a la tarjeta de control como se puede observar en la Fig. D.3.

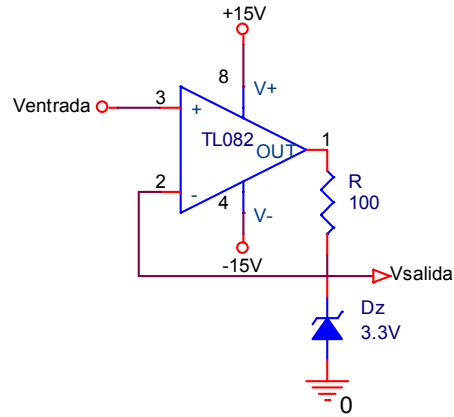


Fig D.3. Limitador de voltaje con rango de salida de 0V a 3.3V para enviar señales de la tarjeta dSPACE al DSP.

### Tarjeta de interfaz del los puertos digitales a salidas analógicas.

Las señales de control  $u$ , parámetro  $\rho$  y voltaje de capacitor  $V_C$ , provenientes de la tarjeta de evaluación son convertidas a señales analógicas por medio de DAC's implementados en esta tarjeta de interfase como se muestra en la Fig. D.4. Esto con el fin de hacer accesible el control  $u$  a la tarjeta dSPACE, mientras que las otras son simplemente desplegadas en el osciloscopio. En esta misma tarjeta se ha incluido un botón de pánico con el fin de parar el proceso manualmente en caso que el operador lo requiera.



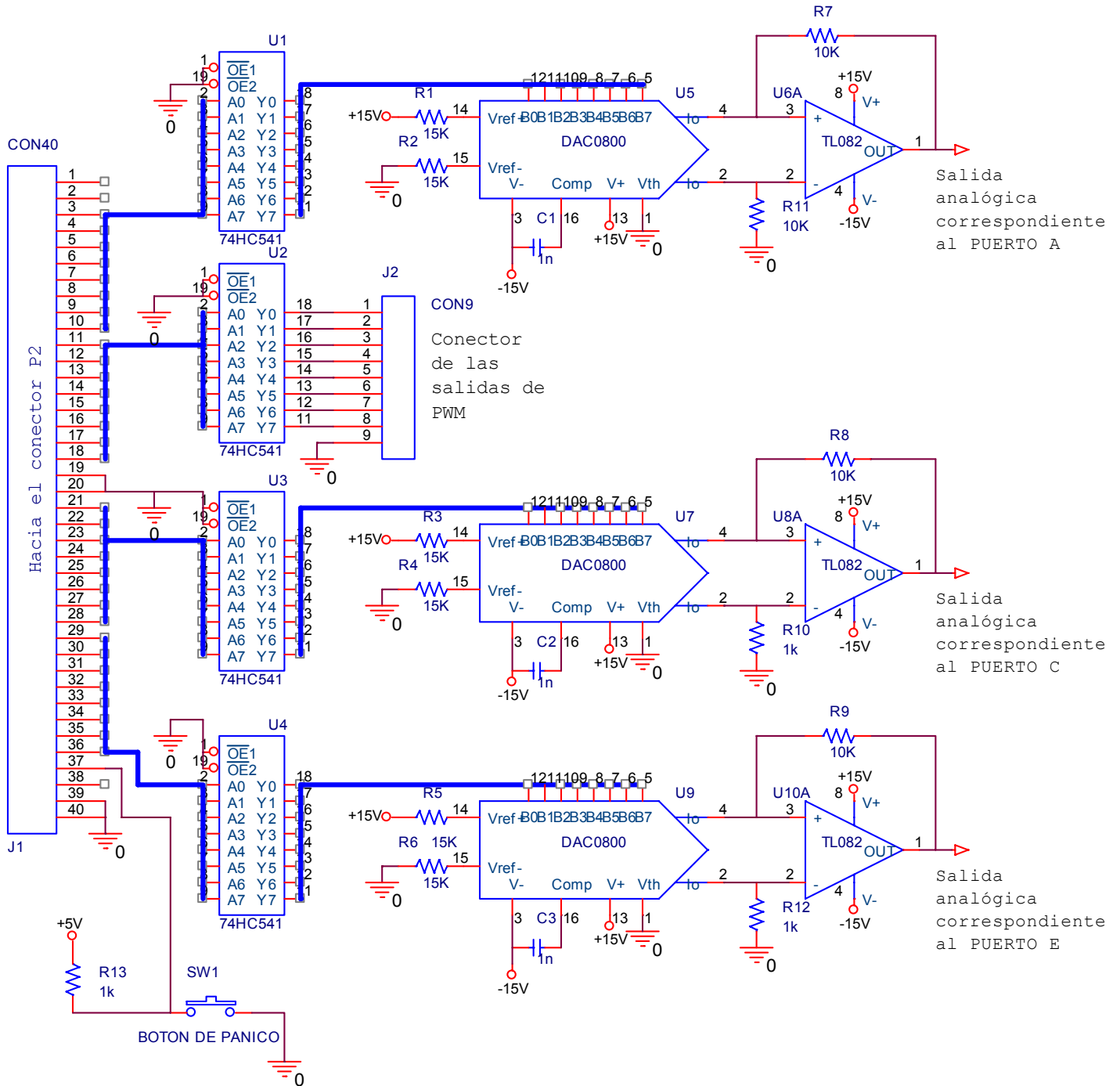


Fig D.4. Tarjeta de interfaz analógica del DSP hacia la tarjeta dSPACE.

## REFERENCIAS

---

- [1] Grady W. Mack, Gilleskie J. Robert. Harmonics and how they relate to power factor. *Proceedings EPRI Power Quality Issues & Opportunities Conference (PQA'93)*, San Diego, CA, November 1993.
- [2] Emanuel A. E. Pileggi D.J. Gentile T.J. Distribution feeders with nonlinear loads in the northeast U.S.A. parte I – voltaje distortion forecast. *IEEE Trans. On Power Delivery*, Vol. 10, No. 1, January 1995, pp. 340-347.
- [3] R. Ortega, A. Loria, P. J. Nicklasson y H. Sira-Ramirez. *Passivity-based control of Euler-Lagrange systems*. Springer-Verlag, 1998.
- [4] L.T. Morán and G. Joos. Principles of active filters. Tutorial course notes. In *Proc. 34th. Annual Meeting IEEE Ind. Appl. Conf. (IAS'99)*, Vol. 4, pp. 2201-2207, 1999.
- [5] IEEE PES Working Group, *FACTS Applications*, IEEE Press, Publ. No. 96-TP-116, 1996.
- [6] J.G. Nielsen, F. Blaabjerg, and N. Mohan. Control strategies for dynamic voltage restorer compensating voltage sags with phase jump. In *Proc. 16th IEEE Applied Power Electr. Conf. and Exposition (APEC'01)*, Vol. 2, pp. 1267-1273, 2001.
- [7] G. Joos. Three-phase static series voltage regulator control algorithms for dynamic sag compensation. In *Proc. 25th Int. Conf. Ind. Elec., Contr., and Instr.: IECON '99*, Vol. 2, pp. 515-520, 1999.
- [8] L. T. Morán, P. Werlinger, J. Dixon, and R. Wallace. A series active power filter which compensates current harmonics and voltage unbalance simultaneously. In *Proc. 26th IEEE Pow. Elec. Spec. Conf. (PESC95)*, Vol. 1, pp. 222-227, 1995.
- [9] J.G. Nielsen, F. Blaabjerg y N. Mohan. Control strategies for dynamic voltage restorer compensating voltage sags with phase jump. In *Proc. 16th IEEE Applied Power Electr. Conf. and Exposition (APEC'01)*, Vol. 2, pp. 1267-1273, 2001.

- [10] H. Akagi and A. Nabae. Control strategy of active power filters using multiple voltage source pwm converters. *IEEE Trans. on Ind. Appl.*, IA-22(3), pp. 460-465, May/June 1986.
- [11] G. Escobar, A.M. Stankovic, V. Cárdenas And P. Mattavelli. An adaptive controller for a series active filter to compensate unbalance, harmonic distortion and flicker. In *Proc. IEEE CIEP'02*, Guadalajara, México, October 20-24, 2002.
- [12] G. Escobar, A.M. Stankovic, V. Cárdenas And P. Mattavelli. A controller to compensate unbalance, harmonic distortion and flicker using a series active filter. In *Proc. IEEE Conference on Decision and Control (CDC'2002)*, Las Vegas, Nevada, December 10-13, 2002.
- [13] G. Escobar, A.M. Stankovic, V. Cárdenas And P. Mattavelli. A dissipativity-based adaptive controller for a series active filter used to compensate current harmonics and voltage unbalance. In *Proc. 2002 IEEE CCA/CACSD02*, Glasgow, Scotland, September 18-20, 2002.
- [14] H.K. Khalil. *Nonlinear systems*. Prentice Hall, 2nd edition, 1996.
- [15] *Control Desk Experiment Guide Dspace for Release 4.0*, August 2003. Dspace Inc.
- [16] *eZDP320LF2407A Technical Reference*. Spectrum Digital Inc.
- [17] *TMS320F/C24x DSP Controllers Reference Guide CPU and Instruction Set*. Texas Instruments.
- [18] *TMS320C2xx/C24x Code Composer User's Guide*. Texas Instruments.
- [19] *TMS320C1x/C2x/C2xx/C5x Assembly Language Tools User's Guide*. Texas Instruments.
- [20] *TMS320LF/LC240xA DSP Controllers Reference Guide*. Texas Instrument.
- [21] J. Leyva-Ramos, G. Escobar, P.R. Martínez and P. Mattavelli. Analog circuits to implement a repetitive controller for harmonic compensation. In *Proc. IEEE CIEP04*, Celaya, México, October 2004.
- [22] Chi-Tsong Chen. *Linear system theory and design*. Oxford University Press, 3ra Ed. 1999.