



**INSTITUTO POLITECNICO NACIONAL**

---

---

**ESCUELA SUPERIOR DE INGENIERIA  
MECANICA Y ELECTRICA**

**SECCIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**

**DISEÑO E IMPLEMENTACION DE UN  
ALGORITMO PARA RELEVADORES DE  
DISTANCIA EN DSP's**

**T E S I S**

**QUE PARA OBTENER EL GRADO DE  
MAESTRO EN CIENCIAS  
CON ESPECIALIDAD EN INGENIERIA ELECTRICA**

**P R E S E N T A :**

**ADRIAN CABALLERO RODRIGUEZ**



**MEXICO, D. F. MARZO 2005**



**INSTITUTO POLITECNICO NACIONAL**  
**COORDINACION GENERAL DE POSGRADO E INVESTIGACION**

*ACTA DE REVISION DE TESIS*

En la Ciudad de México, D.F. siendo las 17:00 horas del día 13 del mes de Diciembre del 2004 se reunieron los miembros de la Comisión Revisora de Tesis designada Por el Colegio de Profesores de Estudios de Posgrado e Investigación de la E. S. I. M. E. Para examinar la tesis de grado titulada:

**“DISEÑO E IMPLEMENTACION DE UN ALGORITMO PARA RELEVADORES DE DISTANCIA EN DSP´s”**

Presentada por el alumno:

**CABALLERO**  
Apellido paterno

**RODRIGUEZ**  
materno

**ADRIAN**  
nombre(s)

Con registro: 

B	0	2	1	4	4	9
---	---	---	---	---	---	---

aspirante al grado de:

**MAESTRO EN CIENCIAS**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **SU APROBACION DE LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

LA COMISION REVISORA

Director de tesis

DR. DAVID SEBASTIAN BALTAZAR

DR. DANIEL OLGUIN SALINAS

DR. RAUL CORTES MATEOS

M. EN C. GILBERTO ENRIQUEZ HARPER

DR. RICARDO OCTAVIO MOTA PALOMINO

M. EN C. TOMAS I. ASIAIN OLIVARES

EL PRESIDENTE DEL COLEGIO

DR. FLORENCIO SANCHEZ SILVA





INSTITUTO POLITECNICO NACIONAL  
COORDINACION GENERAL DE POSGRADO E INVESTIGACION

**CARTA SESION DE DERECHOS**

En la Ciudad de México, Distrito Federal, el día **15** del mes **DICIEMBRE** del año **2004** el (la) que suscribe **ADRIAN CABALLERO RODRIGUEZ** alumno(a) del Programa de **MAESTRIA EN CIENCIAS EN INGENIERIA ELECTRICA** con número de registro **B021449**, adscrito a la Sección de Estudios de Posgrado e Investigación de la ESIME Unidad Zacatenco, manifiesta que es autor(a) intelectual del presente Trabajo de Tesis bajo la dirección del **DR. DAVID SEBASTIAN BALTAZAR** y cede los derechos del trabajo intitulado: **“DISEÑO E IMPLEMENTACION DE UN ALGORITMO PARA RELEVADORES DE DISTANCIA EN DSP 's”**, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección: **acr\_caballero@yahoo.es, dsebasti@hotmail.com**.

Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

**ADRIAN CABALLERO RODRIGUEZ**

Nombre y firma

***DEDICADA A***

***GLORIA y EDUARDO***

***EUGENIA y ANGEL***

## AGRADECIMIENTOS

*A Dios, por darme la vida, salud, y muchas bendiciones.*

*A mis padres, Gloria y Eduardo, por todo su amor y comprensión, por enseñarme a disfrutar la vida e infundar en mi el deseo de superación.*

*Si se puede, verdad papá?.*

*A mis hermanos Angel y Eugenia, por su cariño y apoyo incondicional en todos los aspectos.*

*Al Dr. David Sebastián Baltazar, por brindarme su amistad, por su contribución en la dirección de este trabajo de Tesis y por motivarme en el desarrollo del mismo.*

*A los Profesores de la SEPI-ESIME-IPN, por ayudarme a adquirir los conocimientos necesarios para la realización de este trabajo.*

*A mis amigos chiapanecos: Eriván, Edgar, José, Elio, Manuel, Angel, Otilia, por darme ánimos y palabras de aliento, especialmente a Brenda y Reynol.*

*A todos mis compañeros y amigos de la SEPI, por ofrecerme su valiosa amistad.*

*Al CONACYT e IPN, por el apoyo económico recibido durante los estudios de maestría y en la elaboración de esta tesis, ya que sin éste hubiera sido muy difícil.*

## RESUMEN:

En este trabajo se realiza el diseño y la implementación de un algoritmo digital de protección para relevadores de distancia tipo mho en un microcontrolador DSP.

Este proyecto consiste en el desarrollo de un algoritmo de protección que incluye varias funciones como son: muestrear las señales del sistema eléctrico, procesar estas señales a través de los coeficientes de un filtro digital, obtenidos con el método de *errores mínimos cuadrados*, para formar los fasores de las señales de voltajes y corrientes trifásicas. Posteriormente, se obtiene la impedancia “vista” o medida, a partir de los fasores de voltaje y corriente. El siguiente paso es determinar si la impedancia “vista” se encuentra dentro de la zona de operación del relevador, si es el caso, entonces se genera la señal de disparo del interruptor. El algoritmo también tiene la función de enviar y recibir datos para actualizar sus ajustes cuando así se requiera, todo esto con la finalidad de cumplir con la función primordial de la protección, que es detectar una condición de falla y desconectar el elemento fallado, en el menor tiempo posible.

Esta implementación se realizó en un microcontrolador DSP de punto fijo, es decir que este proyecto se desarrolló en una pastilla o *chip* que solo hace operaciones de punto fijo. Posee un puerto de comunicación serial por donde se transmite en forma bidireccional la información requerida. Esto se realiza mediante un programa digital desarrollado en lenguaje de LabVIEW, instalado en una computadora (PC) que interactúa con el DSP para modificar los ajustes cuando se considere necesario, según la filosofía de protección adaptable. Adicionalmente, se registran las muestras del primer ciclo de falla en el DSP, estas pueden ser descargadas a la PC cuando se requiera analizar el comportamiento de una contingencia dada, o para conocer los voltajes y corrientes presentes en el sistema.

El desempeño del algoritmo de protección implementado en el DSP, se probó utilizando voltajes y corrientes de falla de sistemas eléctricos que fueron modelados con el software EMTDC/PSCAD. Se comparan los resultados obtenidos con el DSP, y los obtenidos a partir de programas escritos en LabVIEW y MATLAB, como se muestran en el capítulo 5.

Finalmente, se presentan simulaciones de fallas en sistemas eléctricos reales donde se muestra que no en todos los sistemas es indispensable la aplicación de esquemas de protección adaptable.

**ABSTRACT:**

This work presents the design and implementation of a type mho protective relaying algorithm for distance relays, using a microcontroller DSP.

This project consists of the development of a protection algorithm that includes several functions such as: sampling of electrical system signals, processing these signals employing the coefficients obtained from a *minimum square error* digital filter to form phasors from the sampled three-phase voltages and currents. Then, the "seen" or measured impedance is computed from the voltage and current phasors. The following step is to determine whether the "seen" impedance is within the relay operating area; if it is the case, a signal to trip the breaker is sent. The algorithm also have functions to send or receive data, in order to update their settings when required, while satisfying the main relay function, this is, detecting any fault condition and isolating faulted element in the minimum possible time.

This implementation was done using a fixed point microcontroller DSP; it means that this project was developed employing a *chip* which is only capable of performing fixed point operations. It has a serial communication port from where the required information is transmitted in a bidirectional form. This is managed by computer programs developed in LabVIEW, installed in a PC that interacts with the DSP in order to modify the settings when required, according to an adaptable protection philosophy. Additionally, the samples the first cycle samples of the fault are registered in the DSP, and they can be unloaded to the PC whenever it is required so as to be able to analyze the system performance under a given contingency, or to know the actual voltages and currents in the system while the contingency was actually happening.

The performance of the protective relaying algorithm implemented in the DSP, was tested using the short circuit voltages and currents of electric power systems modeled on the EMTDC/PSCAD software. The results obtained using the DSP, and the ones from the computer programs written in LabVIEW and MATLAB are compared, as shown in chapter 5.

Finally, the fault simulation results of real electrical systems show that not all the systems would require adaptable protection schemes.

---

## CONTENIDO

<b>RESUMEN.</b>	<b>i</b>
<b>ABSTRACT.</b>	<b>ii</b>
<b>CONTENIDO.</b>	<b>iii</b>
<b>ÍNDICE DE FIGURAS Y TABLAS.</b>	<b>vi</b>
<b>GLOSARIO DE TÉRMINOS Y VARIABLES UTILIZADAS.</b>	<b>ix</b>
<b>CAPITULO 1: INTRODUCCIÓN.</b>	<b>1</b>
1.1. Introducción.	1
1.2. Descripción del problema.	2
1.3. Objetivo.	2
1.4. Justificación.	3
1.5. Antecedentes.	3
1.6. Alcance.	7
1.7. Aportaciones.	7
1.8. Estructura del trabajo de tesis.	8
<b>CAPITULO 2: SISTEMAS DE PROTECCIÓN CON RELEVADORES DE DISTANCIA Y MODELADO DE LINEAS CON EMTDC/PSCAD.</b>	<b>9</b>
2.1 Introducción.	9
2.2. Protección de líneas de transmisión.	10
2.3. Sistemas de protección adaptable.	12
2.4. La protección de distancia.	13
2.4.1. Principio de operación de las protecciones de distancia.	13
2.4.2. Los relevadores de distancia.	15
2.4.2.1. Relevador tipo impedancia.	15
2.4.2.2. Relevador tipo reactancia.	16
2.4.2.3. Relevador tipo mho.	17
2.4.2.4. Relevador con características poligonales.	18
2.4.2.5. Característica tipo mho usada en el algoritmo.	19
2.5. Criterios de ajuste y coordinación de relevadores de distancia con esquema de protección adaptable.	20
2.5.1. Ajuste de la zona 1.	21
2.5.2. Ajuste de la zona 2.	22





2.5.3. Ajuste de la zona 3.	23
2.6. Esquema de protección adaptable para relevadores de distancia.	23
2.7. Características de simulación del sistema eléctrico de potencia.	24
2.7.1. Modelado de las líneas de transmisión en EMTDC/PSCAD.	24
<b>CAPITULO 3: DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO DE PROTECCIÓN EN UN MICROCONTROLADOR DSP.</b>	<b>27</b>
3.1 Introducción.	27
3.2 Algoritmo del filtro digital para el cálculo de fasores implementado en el microcontrolador DSP.	27
3.3 Identificación de la ubicación de fallas en el plano complejo $R+jX$ .	39
3.4 Implementación del esquema propuesto en el DSP 56807 de Motorola.	41
3.4.1. Características del DSP56807 de Motorola.	42
3.4.2. Operación del programa implementado en el microcontrolador.	43
3.5 Adquisición de datos o muestreo de señales del sistema eléctrico de potencia.	49
3.5.1. Adecuación de las señales de voltaje y corriente para ser leídas por el ADC.	51
3.5.1.1. Manejo de las señales de voltaje.	51
3.5.1.2. Manejo de las señales de corriente.	53
3.5.2. Muestreo de señales por el microcontrolador.	55
3.5.2.1. Determinación de la frecuencia de muestreo.	55
3.5.2.2. Programación de la frecuencia de muestreo para el ADC.	56
3.5.2.3. La conversión analógico–digital.	61
3.5.2.4. Ejemplo de la toma de muestras por el ADC.	63
3.6 Actualización de los ajustes de impedancias en el DSP.	65
3.6.1. Diagrama de flujo del programa en LabVIEW.	65
3.6.2. Presentación del programa en LabVIEW.	67
<b>CAPITULO 4: COMPARACIÓN DE RELEVADORES DE DISTANCIA: CON AJUSTES CONVENCIONALES Y EN ESQUEMAS DE PROTECCIÓN ADAPTABLE.</b>	<b>69</b>
4.1. Introducción.	69
4.2 Cálculo de ajustes de los relevadores de distancia.	69
4.2.1 Ajuste de la zona 1.	69

4.2.2 Ajuste de la zona 2.	71
4.3. Cálculo de ajustes con esquemas de protección adaptable.	73
4.3.1 Ajuste adaptable para zona 1.	73
4.3.2 Ajuste adaptable para zona 2.	73
4.4. Comparación de ambos esquemas de protección ante distintas fallas en el sistema eléctrico.	76
<b>CAPITULO 5: ANÁLISIS DE RESULTADOS.</b>	<b>78</b>
5.1. Introducción.	78
5.2 Representación esquemática del DSP como relevador de distancia.	79
5.3 Sistemas de prueba para el DSP.	80
5.3.1 Resultados de la prueba 1.	81
5.3.2 Resultados de la prueba 2.	84
5.4 Sistemas eléctricos de potencia utilizados.	86
5.4.1. Simulación de fallas en el sistema eléctrico de prueba.	87
5.4.2. Simulación de fallas en el sistema eléctrico real 1.	92
5.4.3. Simulación de fallas en el sistema eléctrico real 2.	93
5.4.4. Simulación de fallas en el sistema eléctrico real 3.	95
<b>CAPITULO 6: CONCLUSIONES Y RECOMENDACIONES PARA TRABAJOS FUTUROS.</b>	<b>97</b>
6.1. Conclusiones.	97
6.2. Recomendaciones para trabajos futuros.	99
6.3. Aportaciones de la tesis.	100
<b>REFERENCIAS.</b>	<b>101</b>
<b>APÉNDICE A: CARACTERÍSTICAS Y OPERACIÓN DE LOS MICROCONTROLADORES DSP's 56800 DE MOTOROLA.</b>	<b>105</b>
<b>APÉNDICE B: METODOLOGIA PARA LA PROGRAMACIÓN DEL DSP 56800 CON EL SOFTWARE "CODE WARRIOR".</b>	<b>114</b>
<b>APÉNDICE C: PROGRAMA DIGITAL PARA RELEVADORES IMPLEMENTADO EN EL DSP.</b>	<b>121</b>
<b>APENDICE D: PROGRAMA DESARROLLADO EN LabVIEW PARA INTERACTUAR CON EL DSP.</b>	<b>132</b>
<b>APENDICE E: PROGRAMA EN MATLAB DEL FILTRO DE MÍNIMOS ERRORES CUADRADOS.</b>	<b>137</b>



## ÍNDICE DE FIGURAS Y TABLAS

<b>Figura</b>		<b>Pg.</b>
Figura 2.1	Impedancia medida por el relevador de distancia ante una Falla en el punto F.	14
Figura 2.2	Característica general de un relevador tipo impedancia.	16
Figura 2.3	Característica general de un relevador tipo reactancia.	17
Figura 2.4a	Característica general de un relevador tipo mho.	18
Figura 2.4b	Relevador tipo mho desplazado.	18
Figura 2.5	Característica general de un relevador tipo poligonal.	19
Figura 2.6	Impedancia de la línea en el plano complejo.	20
Figura 2.7.	Zonas de operación del relevador de distancia.	21
Figura 2.8	Relevador de distancia $R_r$ , protegiendo en zona 1 una línea con tres terminales.	21
Figura 2.9.	Relevador de distancia $R_r$ , protegiendo en zona 2 una línea con tres terminales.	22
Figura 2.10.	Relevador de distancia $R_r$ , protegiendo en zona 3 una línea con tres terminales.	23
Figura 3.1	Característica de distancia tipo mho en zona 1, para el DSP.	40
Figura 3.2	Programa principal del algoritmo.	44
Figura 3.3	Transmisión de datos.	45
Figura 3.4	Operación del ADC.	46
Figura 3.5	Función de recepción de datos.	47
Figura 3.6	Ventana de 24 muestras para un ciclo completo de frecuencia fundamental.	50
Figura 3.7a	onda senoidal de voltaje de 360 V de pico a pico.	52
Figura 3.7b	onda senoidal de voltaje de 3 V pico a pico con offset de CD	52
Figura 3.8	Circuito para adecuación de señales de voltaje del sistema de potencia.	52
Figura 3.9	Circuito para adecuación de señales de corriente del sistema de potencia.	54
Figura 3.10	Diagrama esquemático del circuito del ADC.	62
Figura 3.11	Diagrama de tiempos.	63
Figura 3.12	Muestras en tiempo real de señales de voltaje de las fases A y B.	63
Figura 3.13.	Muestras en tiempo real de señales de voltaje y corriente, sin falla.	64
Figura 3.14.	Muestras en tiempo real de señales de voltaje y corriente, con falla.	65
Figura 3.15	Diagrama del envío de datos para el ajuste del ADC.	66
Figura 3.16	Diagrama para recepción de muestras del DSP.	67
Figura 3.17	Programa en LabVIEW para interactuar con el DSP.	68
Figura 3.18	Pantalla en LabVIEW de indicadores de las señales.	68
Figura 4.1	Ajuste en zona 1 para el relevador $r_1$	70
Figura 4.2	Sistema de subtransmisión radial protegido por relevadores de distancia.	71
Figura 4.3	Porción de un sistema de potencia usado como ejemplo	72
Figura 4.4	Localización de fallas para determinar el ajuste en zona 2.	75



Figura 4.5	Porcentaje de alcance en zona 2 para líneas remotas en condición de máxima generación.	77
Figura 4.6	Porcentaje de alcance en zona 2 para líneas remotas en condición de mínima generación.	77
Figura 5.1	Diagrama esquemático del DSP como protección.	79
Figura 5.2	Programa gráfico de EMTDC/PSCAD.	80
Figura 5.3	Prueba al algoritmo con una falla trifásica.	82
Figura 5.4	Impedancia con todas las muestras de la simulación.	83
Figura 5.5	Falla trifásica con muestras de 2 ciclos, uno antes de la falla y uno después de la misma.	84
Figura 5.6	Prueba al algoritmo con una falla monofásica.	85
Figura 5.7	Comprobación de la falla monofásica en Matlab.	86
Figura 5.8	Sistema eléctrico de prueba usado en las simulaciones.	87
Figura 5.9	Gráficas de la impedancia de falla de la tabla 5.3.	88
Figura 5.10	Gráficas de la impedancia de falla de la tabla 5.4.	89
Figura 5.11	Gráficas de la impedancia de falla de la tabla 5.5.	90
Figura 5.12	Sistema eléctrico real 1.	92
Figura 5.13	Gráfica de la impedancia de falla de la tabla 5.6.	93
Figura 5.14	Sistema eléctrico real 2.	93
Figura 5.15	Gráfica de la impedancia de falla de la tabla 5.7.	94
Figura 5.16	Sistema eléctrico real 3.	95
Figura 5.17	Gráfica de la impedancia de falla de la tabla 5.8.	96
Figura A.1.	Diagrama de bloques del núcleo del DSP56800.	105
Figura A.2.	Arquitectura harvard.	106
Figura A.3.	Ejemplo de la construcción alrededor del núcleo del chip DSP56800	107
Figura A.4.	Algoritmos mapeados dentro del hardware del DSP.	108
Figura A.5.	Complemento del diagrama de bloques del núcleo del DSP56800.	109
Figura A.6.	Unidad aritmética lógica.	109
Figura A.7.	Unidad generadora de direcciones.	110
Figura A.8.	Controlador de programas.	111
Figura A.9.	Unidad de buses y manipulación de bits.	112
Figura A.10.	Unidad once.	113
Figura B.1.	Menú del CD de instalación.	114
Figura B.2.	Bienvenida para la instalación.	114
Figura B.3.	Licencia, términos y condiciones.	115
Figura B.4.	Descripciones.	115
Figura B.5.	Tipo de instalación.	115
Figura B.6.	Elección del folder.	116
Figura B.7.	Archivos asociados.	116
Figura B.8.	Preparado para copiar archivos.	116
Figura B.9.	Progreso de la copia de archivos.	117
Figura B.10.	Pregunta de registro.	117
Figura B.11.	Fin de la instalación.	117
Figura B.12.	Menú para crear un programa.	118
Figura B.13.	Menú para la elección del Microcontrolador DSP.	119
Figura B.14.	Proyecto creado en modo estacionario en lenguaje C.	120
Figura B.15.	Botones de compilación, hacer y debug.	120



Figura D.1. Lectura de datos del DSP.	132	
Figura D.2. Formación de arreglos de muestras correspondientes a voltajes y corrientes.	132	
Figura D.3. Subinstrumento "FasoresV3.vi"	133	
Figura D.4. Cálculo de la impedancia de secuencia positiva con IA de falla compensada.	133	
Figura D.5. Subinstrumento "IFcompensada.vi"	134	
Figura D.6. Uso del subinstrumento para detección de fallas.	134	
Figura D.7. Subinstrumento "Deteccióndefallas.vi"	135	
Figura D.8. Programación para graficar la impedancia en forma polar.	135	
Figura D.9. Algoritmo para el envío de datos (nuevos ajustes).	136	
Tabla 3.1. Coeficientes del filtro digital de mínimos cuadrados.	35	
Tabla 3.2. Elección de las entradas del ADC.	60	
Tabla 3.3. Fasores de Voltaje de las fases A y B.	64	
Tabla 5.1. Resultados de la falla trifásica.	81	
Tabla 5.2. Resultados de la falla monofásica.	84	
Tabla 5.3. Impedancias de falla de la prueba S.E.P. 1.	88	
Tabla 5.4. Impedancias de falla de la prueba S.E.P.2.	89	
Tabla 5.5. Impedancias de falla de la prueba S.E.P.3.	90	
Tabla 5.6. Impedancias de falla de la prueba S.E.R. 1.	93	
Tabla 5.7. Impedancias de falla de la prueba S.E.R.2.	94	
Tabla 5.8. Impedancias de falla de la prueba S.E.R.3.	95	



## GLOSARIO DE TERMINOS Y VARIABLES UTILIZADAS.

- R<sub>p</sub> = Relevador primario.  
R<sub>r</sub> = Relevador de respaldo.  
TC's = Transformadores de corriente.  
TP's = Transformadores de potencial.  
K1 = Coeficiente de seguridad para definir el alcance de la zona 1 de los relevadores de distancia.  
K2 = Coeficiente de seguridad para definir el alcance de la zona 2 de los relevadores de distancia.  
Z<sub>r</sub> = impedancia de los relevadores de distancia.  
V<sub>r</sub> = voltaje proporcional al voltaje de falla.  
I<sub>r</sub> = Corriente proporcional a la corriente da falla.  
Z<sub>ar</sub> = mpedancia de arranque del relevador.  
Z<sub>r</sub> = Impedancia "vista" por el relevador.  
X<sub>ar</sub> = Es la reactancia de arranque, parámetro de ajuste.  
X<sub>r</sub> = Es la reactancia "vista" por el relevador ante una falla.  
Z<sub>r</sub> = Impedancia que "ve" el relevador.  
Z<sub>ar max</sub> = Impedancia máxima de alcance.  
θ<sub>r</sub> = Angulo de la impedancia de falla que "ve" el relevador.  
θ<sub>sm</sub> = Angulo de sensibilidad máxima.  
Z<sub>F</sub> = Impedancia de falla  
θ = Angulo de la impedancia de falla.  
[V<sub>modo</sub>] = Modo de voltaje al final de la línea.  
[I<sub>modo</sub>] = Modo de corriente al final de la línea.  
[V<sub>fase</sub>] = Voltaje de fase al final de la línea.  
[I<sub>fase</sub>] = Corriente de fase al final de la línea.  
[T<sub>θ</sub>] = Modo de voltaje para poner en fase la matriz de transformación de voltaje.  
[T<sub>I</sub>] = Modo de corriente para poner en fase la matriz actual de transformación.  
m = Es la muestra o medida tomada.  
t = Es el tiempo.  
v = Es la muestra en cuestión.  
V<sub>p</sub> = Es el voltaje pico de la señal.  
V<sub>P1</sub> = Es el voltaje pico de la fundamental.  
V<sub>P2</sub> = Es el voltaje pico de la 2<sup>a</sup> armónica.  
V<sub>P3</sub> = Es el voltaje pico de la 3<sup>a</sup> armónica.  
V<sub>P4</sub> = Es el voltaje pico de la 4<sup>a</sup> armónica.  
V<sub>0</sub> = Es el voltaje de CD.  
VA<sub>m</sub> = Es la magnitud del fasor de voltaje de la fase A.  
reVA = Es la parte real del fasor de voltaje de la fase A.  
imVA = Es la parte imaginaria del fasor de voltaje de la fase A.  
coefseno[ ] = Vector de coeficientes del filtro digital que forma la parte real del fasor.  
coefcoseno[ ] = Vector de coeficientes del filtro digital que forma la parte imaginaria del fasor.  
muestrasVA[ ] = Son las muestras de un ciclo completo de onda senoidal a



- frecuencia fundamental de voltaje de la fase A.
- $I_{Am}$  = Es la magnitud del fasor de corriente de la fase A.
- $reIA$  = Es la parte real del fasor de corriente de la fase A.
- $ImIA$  = Es la parte imaginaria del fasor de corriente de la fase A.
- $muestrasIA[ j ]$  = Son las muestras de un ciclo completo de la onda senoidal a la frecuencia fundamental de corriente de la fase A.
- $Z_{1falla}$  = Es la impedancia de falla de secuencia positiva medida.
- $VA$  = Es el fasor de voltaje de la fase A.
- $I_{Ax}$  = Es el fasor de corriente compensado de la fase A.
- $IA$  = Es el fasor de corriente de la fase A.
- $I_0$  = Es el fasor de corriente de secuencia cero.
- $m$  = Es una constante compleja, ver ecuación 3.27.
- $IA, IB, IC$  = Son los fasores de corriente de cada una de las fases A, B y C.
- $Z_0$  = Es la impedancia de secuencia cero dada por la longitud de la línea.
- $Z_1$  = Es la impedancia de secuencia positiva dada por la longitud de la línea.
- $angZL$  = Angulo de la impedancia de secuencia positiva de la línea  $ZL$ .
- Zona 1** = Es el porcentaje de la línea a proteger.
- ZE** = Es la impedancia esperada.
- $Z_{0r}$  = Es la parte real de la impedancia de ajuste.
- $Z_{0j}$  = Es la parte imaginaria de la impedancia de ajuste.
- $V_{offset}$  = Es el valor de voltaje offset de CD
- $V_T$  = Es el voltaje total  $V_{cc}$ .
- $fm$  = Es la frecuencia de muestreo.
- $T$  = Es el período de la señal a muestrear (la fundamental).
- $Ftemp$  = Es el periodo del temporizador.
- $FIP$  = Es la frecuencia de operación del IP BUS.
- $Pre$  = Es número del preescalador.
- $Fmuestra$  = Frecuencia de muestreo deseada.
- $inter$  = Intervalo de tiempo entre cada muestra.
- $Z1r1$  = Es el ajuste en Zona 1 para el relevador  $r1$ .
- $Z_{1L}$  = Es la impedancia de secuencia positiva de la línea.
- $K$  = Es el factor de seguridad y se establece entre 0.8 y 0.9.
- $R_{BK}$  = Relevador de respaldo cuyo ajuste en zona 2 debe ser determinado;
- $R_{PRi}$  = Relevador primario protegiendo la línea B-C<sub>i</sub>;
- $Z_{AB}$  = Impedancia de secuencia positiva de la línea A-B;
- $Z_A(R_{BK}, F_i)$  = Impedancia aparente vista por el relevador  $R_{BK}$  para la falla  $F_i$ ;
- $Z_2(R_{BK}, R_{PRi})$  = Posible ajuste de la zona 2 para el relevador  $R_{BK}$
- $Error$  = Es el error relativo.
- $resPSCAD$  = Resultado arrojado por la simulación en EMTDC/PSCAD.
- $resDSP$  = Resultado calculado por el Microcontrolador DSP.
- Nodo remoto.** Está relacionado con la ubicación de un relevador y corresponde al nodo situado en el extremo opuesto de la línea donde se ubica el relevador.
- Líneas remotas.** Corresponde a las líneas adyacentes al nodo remoto.
- Impedancia aparente o medida.** Es el cociente entre el voltaje y corriente que miden los relevadores de distancia.
- DSP.** Procesador de señales digitales o microcontrolador.
- ADC.** Convertidor analógico – digital.

# CAPÍTULO I

## 1. INTRODUCCIÓN

### 1.1 Introducción.

En cualquier aplicación referente a protección de sistemas eléctricos de potencia, el principio básico es que dicha protección opere en el menor tiempo posible para, evitar daños al sistema en cuestión [1,2]. Dichas protecciones son ajustadas para brindar una protección adecuada y proporcionar una coordinación en la operación, de tal manera, que se aisle únicamente al elemento fallado [1,2,3,4,5].

Las líneas de transmisión de los sistemas eléctricos de potencia son de gran importancia debido a que están encargadas de transportar la energía eléctrica, desde el punto de generación, hasta las subestaciones de distribución. Por tal motivo es importante desarrollar sistemas de protección eficientes y confiables para la protección de estos elementos.

En la actualidad algunos sistemas eléctricos de potencia necesitan de una protección que pueda cambiar los parámetros de ajuste a medida que sea requerido, es decir, cuando las condiciones del sistema varíen, por ejemplo, debido a las variaciones de carga en determinadas horas del día, por el mantenimiento de líneas de transmisión o de generadores o después de la liberación de una falla. Todo esto ocasiona un cambio en la topología de la red, por lo tanto, también se provocan cambios en los parámetros del sistema. Por estos motivos es necesario un ajuste actualizado en los relevadores para lograr una protección confiable [5]. Para lograr esto, es necesario disponer de un esquema de protección que pueda adaptarse a las condiciones cambiantes del sistema eléctrico de potencia [6].



## **1.2 Descripción del problema.**

Las protecciones de un sistema de transmisión tienen como función básica detectar fallas en líneas o subestaciones y aislarlas, interrumpiendo la corriente de cortocircuito. La liberación de una falla debe hacerse lo más rápido posible tratando de desconectar únicamente al elemento fallado, para minimizar daños al sistema, a equipos y usuarios [1,2,3,4,5].

En la protección de un sistema de transmisión se tienen varios esquemas de protección, entre los cuales se encuentran los relevadores de distancia, que es el esquema a tratar en este trabajo. En estos relevadores se pueden tener ajustes que en determinado momento pueden no ser los adecuados para las condiciones de operación del sistema, es decir, para condiciones diferentes para las cuales se realizó el estudio de protecciones, estas modificaciones pueden ser: los cambios de carga y configuración del sistema ante liberación de fallas o cualquier otra contingencia, por estas condiciones de operación es necesario disponer de relevadores de distancia que puedan cambiar sus ajustes para dicha condición. Cabe recalcar que no siempre es necesario hacer modificaciones en los ajustes de los relevadores, porque como se muestra en el capítulo 5, en algunos casos no hay necesidad de aplicar técnicas de protección adaptable.

## **1.3 Objetivo.**

La propuesta de este trabajo de tesis es diseñar e implementar un algoritmo digital de protección para relevadores de distancia en un microcontrolador DSP, que pueda modificar sus ajustes en tiempo real para proporcionar el mayor alcance posible en las líneas de transmisión del sistema eléctrico de potencia sin llegar a tener traslapes de zonas. Se pretende mantener una comunicación frecuente (disponible en cualquier momento) entre el DSP y la computadora encargada de interactuar con éste, para poder modificar el ajuste cuando sea necesario.

#### **1.4 Justificación.**

En el área de protección de los sistemas eléctricos de potencia, las herramientas computacionales se han utilizado para realizar simulaciones y analizar el comportamiento de dichos sistemas y sus elementos, durante cortocircuitos y otros regímenes de operación, estas simulaciones proporcionan la información necesaria para determinar los ajustes de las protecciones.

En la actualidad las protecciones en los sistemas eléctricos de potencia se desarrollan para cumplir con los requerimientos de confiabilidad y seguridad necesarios para mantener la mejor protección posible [2]. Por tal motivo es muy importante hacer estudios de simulaciones para instalar esquemas de protección que cumplan con las normas y que hagan mas eficiente, confiable y segura, la operación de las protecciones en los sistemas eléctricos de potencia.

La evolución de la tecnología de protecciones ha introducido en el mercado relevadores digitales para la protección de sistemas de potencia, así como la infraestructura de comunicaciones en subestaciones y centros de control, que sugieren la posibilidad de desarrollar sistemas de protección que puedan adaptarse a las condiciones cambiantes del sistema, aunque su instalación requiere de un alto grado de confiabilidad de las comunicaciones y en su caso, de la inversión económica incurrida para mantener comunicados todos los puntos del sistema de potencia.

#### **1.5 Antecedentes.**

En los sistemas eléctricos de potencia, se han desarrollado diferentes esquemas de protección para cada uno de sus componentes [1,2,4,5]. Por ejemplo, en la protección de sistemas de transmisión de 115 kV o niveles de voltajes superiores donde se usan esquemas de protección primaria, a la par con esquemas de respaldo. La protección de distancia es uno de los esquemas que presenta mayor

dificultad para determinar la impedancia de ajuste en las zonas de respaldo, especialmente en sistemas mallados, donde las aportaciones de corriente de falla a un nodo intermedio entre el relevador y el punto de falla, dependen de las condiciones de operación y estas corrientes, a su vez, ocasionan que la impedancia medida por los relevadores sea distinta a la impedancia equivalente de las líneas involucradas entre el relevador y el punto de falla [3].

La tecnología convencional de protecciones es rígida y generalmente, se ajusta en base a un punto de operación, y ésta se actualiza periódicamente, es decir, se modifica en cada ocasión en que se considere un aumento de carga en el sistema, debido al incremento de usuarios o por la introducción al sistema de potencia de un nuevo generador o por alguna nueva línea de transmisión. Por otro lado, cuando los ajustes de los relevadores de protección se establecen considerando únicamente un punto de operación del sistema, la operación del relevador durante una falla puede tener como consecuencia un subalcance o sobrealcance ante cualquier contingencia, sobre todo si han ocurrido modificaciones de la configuración de la red o cambios de condiciones de carga/generación [1,2,3].

El concepto de protección adaptable aparece a finales de los años ochentas y se define como una filosofía de protección que permite y busca hacer cambios de diversas funciones de protección, todo esto con el objetivo de actualizar el ajuste de algunos elementos de protección para que respondan a las nuevas condiciones presentes en el sistema, después de determinada contingencia o condición de operación.

El objetivo fundamental de un sistema de protección adaptable es mejorar el funcionamiento de los esquemas de protección, en términos de la confiabilidad de operación; la confiabilidad de un sistemas de refiere al grado de certeza que ofrece un elemento o sistema de operar correctamente, o cumplir satisfactoriamente la función para la cual fue diseñado. La confiabilidad de la protección se divide en dos aspectos: la dependabilidad y la seguridad. La dependabilidad se define como “el

grado de certeza de que un relevador o el sistemas de protección operará correctamente para todas las fallas para las cuales fue diseñado”. La seguridad se define como “el grado de certeza de que un relevador o el sistema de protección no operará para una condición anormal para la cual no fue diseñado” [5].

En la simulación de protecciones se han desarrollado herramientas computacionales para estudiar el comportamiento de los relevadores con el uso del software MATLAB® y SIMULINK® [7], un ejemplo de esto, es el desarrollo de un simulador digital para diseñar y probar los algoritmos que pueden ser implementados en los nuevos relevadores de protección [8,9].

Hoy en día se tienen en el mercado relevadores de protección microprocesados, es decir, que cuentan con un microprocesador digital, y que tiene el potencial de poder modificar remotamente sus ajustes en tiempo real, vía enlaces de comunicación digitales. Sin embargo, se requiere de un centro de control donde puedan realizarse estudios para determinar los ajustes que respondan adecuadamente ante condiciones de fallas, la aplicación de estos relevadores se presentan para la protección de líneas de transmisión y distribución [10,11], transformadores de potencia y generadores, etc. En México se están sustituyendo relevadores electromecánicos por digitales, debido a que éstos presentan muchas ventajas por el uso de la tecnología digital, principalmente porque se reducen los costos y se mejora la confiabilidad de los esquemas de protección al usar este tipo de dispositivos [12].

En 1995, CIGRE reporta varios trabajos revisados en el “Comité de Estudio del Grupo 34”. Uno de estos trabajos está dirigido al diseño y evaluación de un sistema de protección adaptable para una red de distribución y probado en el laboratorio [13]. En la aplicación de los microcontroladores y microprocesadores se tienen varios desarrollos, en los que se encuentran trabajos de sistemas de protección, control y medición (PCYM), por ejemplo, en el VII simposio iberoamericano sobre protección de sistemas eléctricos de potencia, se reporta un trabajo donde se presenta una implementación de un sistema integrado PCYM, para el monitoreo de las variables

eléctricas incluyendo la comunicación entre relevadores y el centro de control, vía red de área local (LAN) o intranet (WAN), para registrar las condiciones del sistema eléctrico, así poder controlar su operación [14].

Una compañía estadounidense también reporta trabajos de sistemas SCADA con la aplicación de relevadores microprocesados y procesadores de comunicación, para el monitoreo y automatización de una Subestación Integrada (SI), con esto se logra un gran avance en la interfaz hombre-máquina (MMI) que permite tener un control integral de toda la subestación, desde los relevadores de protección, hasta los parámetros del transformador de potencia [15].

En trabajos de protecciones en la SEPI ESIME IPN, se presentan varias tesis. En [3] se presenta una metodología para ajustar y coordinar las protecciones de sobrecorriente de un sistema eléctrico considerando la filosofía de protección adaptable. También en [16] se presenta una alternativa para detectar y clasificar las fallas en un sistema de transmisión utilizando redes neuronales artificiales. En [17] se realiza el monitoreo y protección digital de un generador experimental, implementando las seis protecciones básicas del generador.

También se han presentado trabajos con aplicaciones de microcontroladores y/o microprocesadores en el departamento de Ingeniería Eléctrica de la SEPI-ESIME, Zacatenco, en donde se encuentra el desarrollo de un sistema de control con técnicas difusas para un generador síncrono [18]. Dentro de estos desarrollos también se ha realizado el control de una planta de emergencia basado en microcontrolador [19]. Otro trabajo destacado se refiere a la aplicación de un microcontrolador en la realización de un prototipo para la medición de potencia eléctrica [20]. También se han hecho aplicaciones en protecciones como en [21], donde se realiza un algoritmo para la protección de un transformador de potencia y en [22] se desarrolla un relevador de multifunciones para la protección de un generador.

Las aplicaciones desarrolladas con el software comercial LabVIEW® [23] destaca el trabajo donde se monitorean los parámetros y se calcula el ángulo de carga de un generador síncrono de laboratorio por medio de instrumentación virtual [24] y en [25] se presenta un estudio dirigido al diagnóstico oportuno de los generadores eléctricos para evitar daños que puedan interrumpir su funcionamiento.

### **1.6 Alcance.**

Diseñar e implementar un algoritmo de protección digital en un DSP. Esta implementación cuenta con funciones que permiten descargar a una computadora, cuando así se requiera, las muestras de voltajes y corrientes de falla de un ciclo de frecuencia fundamental, que pueden ser útiles para analizar el comportamiento de una contingencia, o para conocer dichos parámetros presentes en el sistema eléctrico de potencia.

### **1.7 Aportaciones.**

Diseño e implementación de un algoritmo digital de protección para relevadores de distancia en el microcontrolador DSP56807.

Desarrollo de programas computacionales en MATLAB y LabVIEW para simular la operación de relevadores de distancia.

Coordinación entre el microcontrolador y la PC mediante un programa en LabVIEW para el intercambio de información por medio del puerto de comunicación serial.

Desarrollo de un programa en MATLAB para el cálculo de los coeficientes del filtro digital de errores mínimos cuadrados para 24 muestras, incluyendo en el modelo, señales de frecuencia fundamental, 2ª, 3ª y 4ª armónica, CD y CD decreciente.

### **1.8 Estructura del trabajo de tesis.**

La estructura de este trabajo de tesis se presenta de la siguiente manera: en el Capítulo 1 se describe la justificación del trabajo y el problema a resolver, el objetivo, los trabajos que se han desarrollado en este campo. Así como también el alcance. En el Capítulo 2 se describen las protecciones eléctricas, las características de los relevadores de distancia y sus criterios de ajuste, así como también el modelo matemático de simulación para las líneas de transmisión con EMTDC/PSCAD. En el Capítulo 3 se presenta el algoritmo desarrollado en un microcontrolador DSP que hará la función de protección, con la característica de poder modificar sus ajustes cuando se requiera, y también se presenta el programa escrito con el software de LabVIEW para interactuar con DSP. En el Capítulo 4 se realiza la comparación del comportamiento de sistemas de protección cuando los ajustes se calculan con métodos convencionales y cuando se obtienen con la filosofía de protección adaptable. En el Capítulo 5 se presentan los sistemas de prueba, los tipos de pruebas y el análisis de resultados. Finalmente, en el Capítulo 6 se presentan las conclusiones del trabajo, las aportaciones de la tesis y recomendaciones para trabajos futuros se puedan realizarse en este campo.

## CAPÍTULO II

### **2. SISTEMAS DE PROTECCIÓN CON RELEVADORES DE DISTANCIA Y MODELADO DE LINEAS CON EMTDC/PSCAD.**

#### **2.1. Introducción:**

La protección del sistema de transmisión tiene como función básica detectar las fallas que se presentan en las líneas o en las subestaciones y aislarlas rápidamente, abriendo todos los caminos de la corriente de cortocircuito. La liberación de la falla se debe hacer lo más rápido posible para minimizar los daños al sistema y a los usuarios. Sin embargo, la protección debe ser selectiva, esto es, tratar de aislar solo el elemento fallado [4], para evitar la salida innecesaria de elementos, por lo tanto, minimizar la sección desconectada del sistema.

La protección de los sistemas de transmisión está formada por una protección primaria y una protección de respaldo. En la protección primaria la operación es de alta velocidad y es la que trata de aislar la mínima sección de la red ante la falla; la protección de respaldo es de acción retardada (con un tiempo suficiente para permitir que opere primero la primaria), es decir, opera solo si falla la protección primaria. Además, la operación de esta protección seguramente dejará fuera de servicio un área mayor del sistema.

La coordinación de los relevadores de distancia, se lleva a cabo mediante la definición del conjunto de relevadores primarios que debe respaldar cada relevador, definiendo además el ajuste de todos los relevadores, de tal manera que, ante una falla, los relevadores que actúan sobre los interruptores que conectan o desconectan el elemento fallado sean los que deben liberarla y solo en caso de que falle la protección primaria, deberá operar la protección de respaldo.

Las ideas acerca de esquemas de protección adaptable se han desarrollado desde hace más de 25 años [6,26]. Aunque se puede decir que la adaptabilidad resurge en



los años ochenta, como resultado del desarrollo en la tecnología de los sistemas y las computadoras digitales para realizar funciones de protección y control de subestaciones.

## **2.2. Protección de líneas de transmisión.**

De acuerdo con estadísticas de la operación de los sistemas eléctricos de potencia, el mayor porcentaje de las fallas se presentan en las líneas de transmisión, independientemente del origen de las mismas, las más comunes son las de fase a tierra y las de fase a fase [1].

En consecuencia, se puede decir que los dispositivos que más operan son los que están protegiendo a las líneas de transmisión.

El estudio de protecciones se agrupa para su aplicación general, tomando en consideración los aspectos particulares de las redes eléctricas en la transmisión, como son:

- Nivel de voltaje.
- Longitud de las líneas de transmisión.
- Ubicación dentro de la red.
- Grado de interconexión.

Para establecer una diferencia en los conceptos de aplicación por niveles de voltaje, se pueden agrupar en la forma siguiente:

- Líneas de transmisión.

De 69 kV y tensiones mayores, por lo general manejan cantidades importantes de potencia y constituyen la estructura principal de la red eléctrica. Pueden ser radiales, longitudinales o formar mallas de interconexión [27].

- Líneas de subtransmisión.  
Son las correspondientes a tensiones inferiores a 69 kV y que no caen en las tensiones de distribución, transportan potencias menores que las de transmisión y generalmente constituyen la etapa intermedia de una red eléctrica, constituyendo la fuente de alimentación de las redes de distribución [27].
- Redes de distribución.  
Forman las partes de menor nivel de voltaje en la transmisión (13.8 y 34.5 kV), también representan la parte más interconectada y que manejan por circuito individual menor cantidad de potencia [27].

La protección de las líneas de transmisión está considerada como la más compleja de las aplicaciones que tiene la protección eléctrica. Esto se debe al gran volumen de información y factores que influyen en la definición de ajuste de los relevadores. La variedad de configuraciones que pueden existir en la topología de la red y los niveles de voltaje de los sistemas influyen en la determinación del esquema de protección.

Los esquemas de protección que se emplean en las líneas de transmisión pueden ser: la protección de sobrecorriente, la protección de distancia, la protección hilo piloto y la protección híbrida. Es común, que la protección de sobrecorriente se use para proteger las líneas contra fallas a tierra y la protección de distancia para fallas entre fases y fallas a tierra.

La protección piloto se emplea como una protección primaria, es decir, está diseñada para proveer en forma instantánea el disparo simultáneo de las terminales de la línea fallada. Este esquema de protección es usado en las situaciones siguientes: cuando se requiere que un interruptor opere en un tiempo más corto que el tiempo que operan normalmente la protección de distancia o sobrecorriente, cuando se usa el recierre de alta velocidad o cuando no es posible coordinar adecuadamente la protección primaria y respaldo de los relevadores de distancia o sobrecorriente.

El esquema de protección que se emplea en muchos sistemas para proteger las líneas de transmisión de alto voltaje, es la protección de distancia, porque es el tipo de protección que mejor detecta las fallas que se presentan dentro de su zona de alcance.

### **2.3 Sistemas de protección adaptable.**

Se entiende por un esquema de protección adaptable como aquel que se **adapta** o ajusta a las condiciones de operación del sistema en cada instante de tiempo, proporcionando, en todo momento, la función para el cual es diseñado [6,28,29,30], es decir, que independientemente del punto de operación del sistema o aparato protegido, el esquema de protección seguirá operando correctamente, aún si el sistema eléctrico de potencia varíe en sus características de operación, ya sea por la liberación de una falla en otra sección del sistema o por la salida o entrada de generadores, también por la variación de la propia carga, en determinadas horas del día o por alguna otra razón. Sin embargo, la adaptabilidad se puede lograr para algunos esquemas, aún sin modificar el ajuste de los relevadores, debido a que su principio de operación responde adecuadamente en distintas condiciones de operación del sistema, es decir, que con las nuevas condiciones del sistema no es necesario hacer una modificación de los ajustes en los relevadores, para seguir proporcionando una protección adecuada, como es el caso, por ejemplo, de los relevadores de sobrecorriente de tiempo inverso [5,3].

Hoy en día, el desarrollo de las comunicaciones permite la aplicación de sistemas de protección adaptable. Sin embargo el gasto económico para su instalación es elevado, además, como se demuestra en el capítulo V de este trabajo, no siempre es aplicable el uso de estos esquemas, debido a que su aplicación depende en gran medida, de la configuración del sistema eléctrico bajo estudio.

En la actualidad, la evolución tecnológica apunta hacia la tendencia de proteger los elementos de un sistema de potencia con relevadores que sean adaptables; el término adaptable se refiere a que el relevador puede modificar sus ajustes en base

a las condiciones actuales de operación del sistema eléctrico de potencia, siempre y cuando el estudio de las simulaciones así lo demuestren.

#### **2.4 La protección de distancia.**

La protección de distancia en líneas de transmisión es aplicable normalmente a líneas de alto voltaje, por ejemplo 230 y 400 KV. y en otros casos cuando la característica de la red lo justifique en tensiones menores[1,2].

El principio de operación de esta protección se basa en el uso de los relevadores de distancia que en realidad miden la “distancia” entre el punto de su localización y el de la falla en forma indirecta, por medio de la impedancia de la línea o su admitancia.

Las características de operación de los relevadores de distancia se basan en el cociente de los fasores voltaje/corriente, y operan cuando la relación voltaje/corriente medida es menor que el valor de ajuste, definido en el relevador.

En condiciones de falla en un sistema radial, la impedancia medida por el relevador es proporcional a la longitud de la sección de línea comprendida entre el relevador y el punto de cortocircuito, debido a que la impedancia medida es igual a la impedancia de secuencia positiva de esa porción de línea. Esta característica de operación permite que los relevadores de distancia tengan distintos alcances, conocidas como zonas de operación.

##### **2.4.1. Principio de operación de las protecciones de distancia**

El principio de operación de los relevadores de distancia se muestra en la Figura 2.1. Se considera que el relevador está ubicado en el punto K y que sus bobinas reciben las señales de  $V_r$ , que es el voltaje proporcional al producido por la falla  $V_F$  y de la corriente de falla  $I_r$ , que es proporcional a la corriente  $I_F$ . La ecuación (2.1) representa la impedancia medida por los relevadores de distancia ( $Z_r$ ). Durante los cortocircuitos

trifásicos la impedancia medida  $Z_r$  coincide con la impedancia de la sección de la línea comprendida entre el punto de la ubicación del relevador y del punto del cortocircuito, esto se cumple si no existen contribuciones de corriente en la sección protegida por el relevador.

$$Z_r = V_r / I_r \quad (2.1)$$

En un sistema radial, la relación de voltaje – corriente que mide un relevador de distancia ante una falla, es proporcional a la distancia física que existe entre la localización del relevador y el punto de falla.

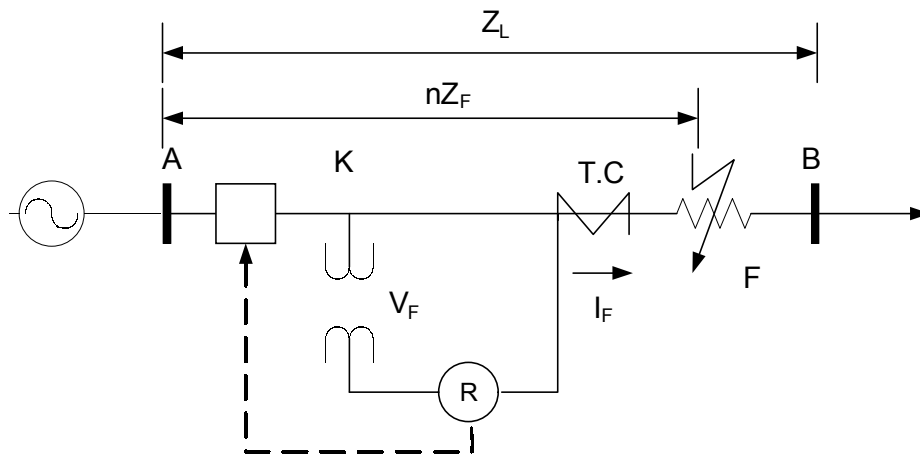


Figura 2.1. impedancia medida por el relevador de distancia ante una falla en el punto F.

En un sistema mallado, no siempre se cumple que la relación de voltaje – corriente proporcione la distancia física aproximada al punto de falla. Esta relación únicamente es válida cuando las fallas se presentan en la línea donde se localiza el relevador o a un nodo de distancia en la dirección de operación del relevador.

Para todas aquellas fallas que se presentan en cualquier otro punto del sistema, y que hagan operar al relevador, la relación voltaje – corriente medida proporciona una distancia eléctrica que difiere de la distancia física que existe entre el punto donde se localiza el relevador y el punto de falla. Esto se explica debido a la contribución de corriente de otras líneas que inciden en nodos ubicados entre el relevador y la falla;

en estos casos la referencia del punto donde ocurrió la falla la proporciona la información de la zona donde operó el relevador.

#### **2.4.2. Los relevadores de distancia.**

Los relevadores de distancia son dispositivos de protección que funcionan en base a una impedancia medida que es comparada con una impedancia de ajuste, de esta manera pueden detectar cuando ocurre una falla en su zona de alcance [1]; la impedancia medida se obtiene como la relación voltaje/corriente existente en la línea protegida. En algunos casos se presenta el efecto *infeed* o de fuente intermedia, que se define como las aportaciones de corriente de falla a puntos intermedios entre la ubicación del relevador y el punto de falla.

En la actualidad se tienen en el mercado relevadores de distancia microprocesados, es decir, que tienen como “cerebro” un microprocesador que realiza la formación de los fasores de voltaje y corriente que son utilizados para obtener la impedancia medida, desde la ubicación del relevador, hasta el lugar donde ocurre la falla.

Estos dispositivos tienen normalmente 3 zonas de operación y existen diversas características de operación de los relevadores de distancia cuando se representan en el plano complejo. A continuación, se presentan algunos tipos comunes de relevadores de distancia dependiendo de la forma de su característica en el plano complejo.

##### **2.4.2.1 Relevador tipo Impedancia.**

La característica de un relevador tipo impedancia es una circunferencia con centro en el origen de coordenadas [31], como se muestra en la Figura 2.2; el valor de la impedancia de arranque ( $Z_r$ ) es independiente al ángulo de la impedancia de falla, es decir, del ángulo  $\theta_r$  que “ve” el relevador y por lo tanto carece de direccionalidad. La ecuación 2.2 representa la condición de operación de este tipo de relevadores.

$$Z_r \leq Z_{ar} \quad (2.2)$$

donde:

$Z_{ar}$  = Impedancia de arranque del relevador.

$Z_r$  = Impedancia "vista" por el relevador.

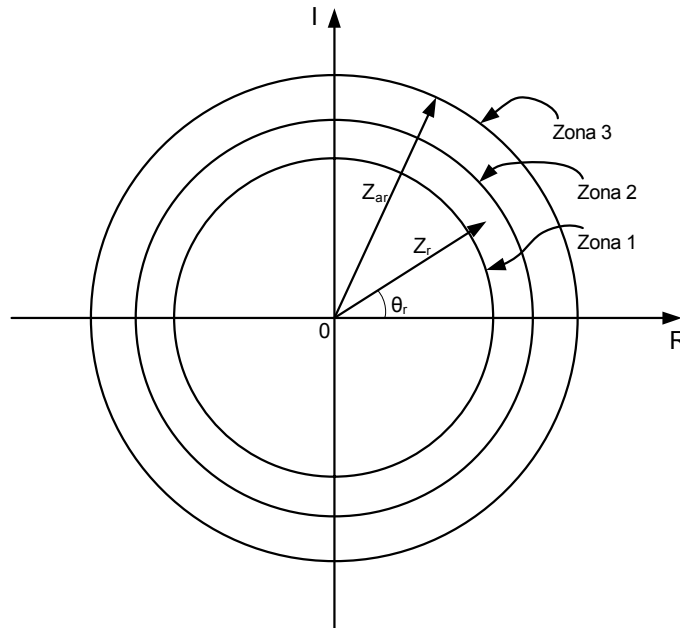


Figura 2.2. Característica general de un relevador tipo impedancia.

#### 2.4.2.2 Relevador tipo Reactancia.

La característica de un relevador tipo reactancia es una línea recta paralela el eje real del plano complejo, tal como se muestra en la Figura 2.3. su condición de operación está dada por la ecuación 2.3.

$$X_r \leq X_{ar} \quad (2.3)$$

donde:

$X_{ar}$  = Es la reactancia de arranque, parámetro de ajuste.

$X_r$  = Es la reactancia "vista" por el relevador ante una falla.

Estos relevadores solo verifican la componente reactiva de la impedancia de falla, y no son afectados por la resistencia de falla, lo que los hace recomendables para la protección de líneas cortas y para la protección contra cortocircuitos a tierra, en los que la resistencia de falla puede tener valores muy elevados.

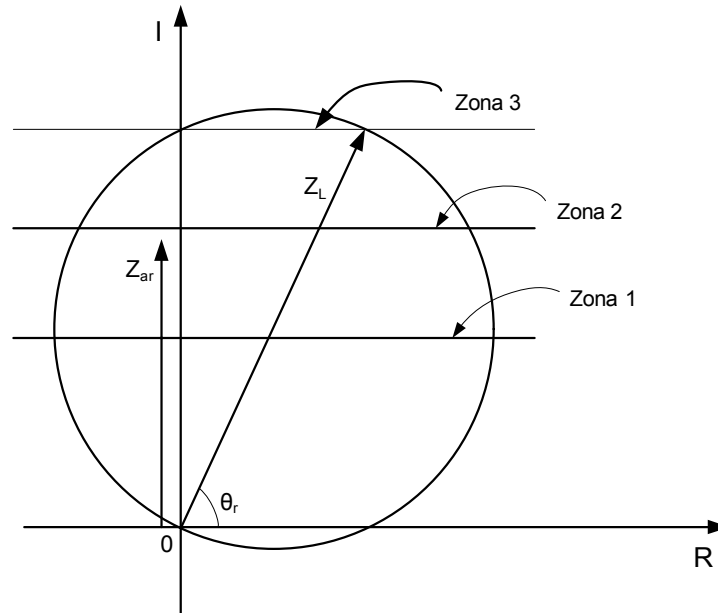


Figura 2.3. Característica general de un relevador tipo reactancia.

### 2.4.2.3 Relevador tipo Mho.

La característica de operación de un relevador tipo mho es una circunferencia que cruza por el origen de coordenadas, tal como se presenta en la Figura 2.4a. Este tipo de relevadores no responden a fallas que se presentan en las líneas situadas detrás del relevador, éstos son considerados direccionales, porque solo operan ante fallas que se presentan en la dirección de disparo del relevador. La condición de operación está dada por la ecuación 2.4.

$$Z_r \leq Z_{ar \max} \cos(\theta_r - \theta_{sm}) \quad (2.4)$$

donde:

$Z_r$  = Impedancia que "ve" el relevador.



$Z_{ar \max}$  = Impedancia máxima de alcance.

$\theta_r$  = Angulo de la impedancia de falla que “ve” el relevador.

$\theta_{sm}$  = Angulo de sensibilidad máxima.

La característica de operación de este tipo de relevador puede desplazarse con respecto al origen de coordenadas, como se muestra en la Figura 2.4b.

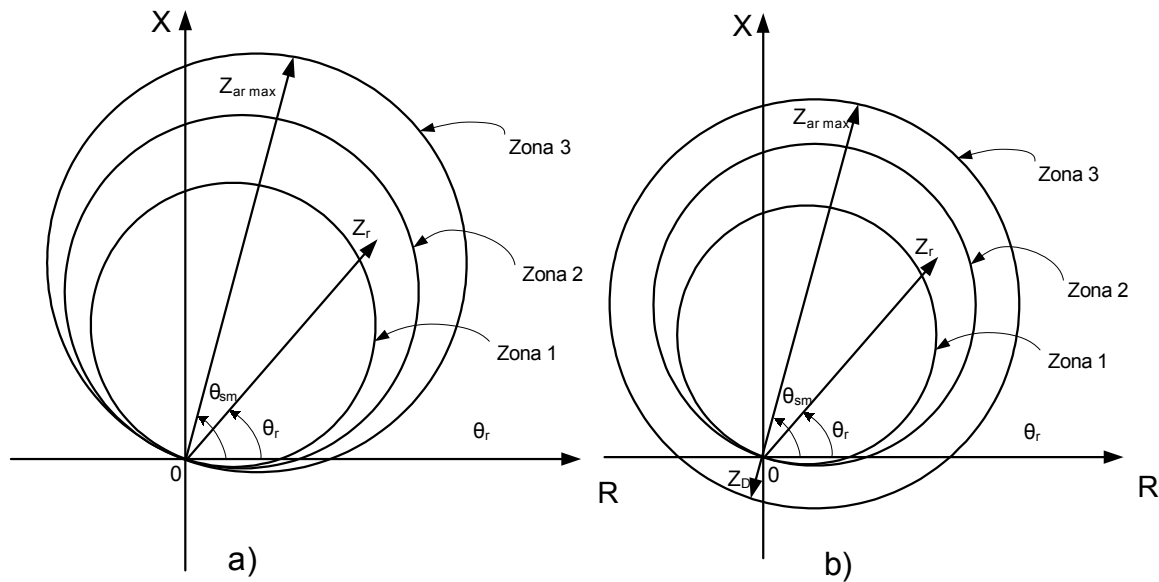


Figura 2.4. a) Característica general de un relevador tipo Mho, b) relevador tipo Mho desplazado.

#### 2.4.2.4 Relevador con características Poligonales.

La zona de operación de este tipo de relevadores está definido en el plano complejo a través de líneas rectas que cortan al eje X y al eje R, con un alcance máximo de  $Z_{m\acute{a}x}$ . La Figura 2.5. muestra la característica de un relevador poligonal; el ajuste de este tipo de relevadores lo define la reactancia de ajuste  $\{X_{ajus}\}$  (distancia del origen al punto donde la recta pendiente cero corta al eje X), y la relación R/X, que define el alcance sobre el eje R como un múltiplo de  $X_{ajus}$  [32]; en la Figura 2.5. se muestra que el alcance en el eje reactivo y resistivo tienen el mismo alcance, es decir, existe

una relación  $R/X = 1$ , pero el ajuste se puede hacer independientemente uno del otro. La condición de operación está dada por las ecuaciones 2.5a y 2.5b.

$$ZF \sin(\theta) \leq X_{ajus} \quad (2.5a)$$

$$ZF \cos(\theta) \leq (R/X) X_{ajus} \quad (2.5b)$$

donde:

ZF = Impedancia de falla

$\theta$  = Angulo de la impedancia de falla.

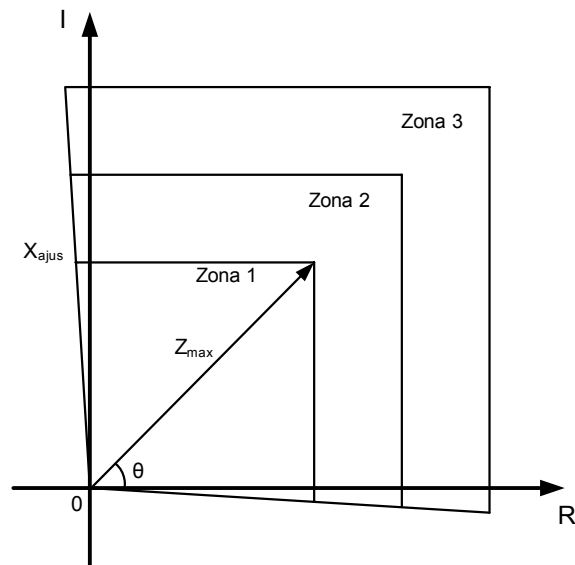


Figura 2.5. Característica general de un relevador tipo poligonal.

#### 2.4.2.5 Característica tipo mho usada en el algoritmo.

La característica de operación del relevador de distancia que se usa en el algoritmo desarrollado, es el tipo Mho, únicamente se ajusta la primera zona, para comprobar la operación correcta de dicho algoritmo. Esta impedancia se puede representar en el plano complejo, tal como se presenta en la Figura. 2.6.

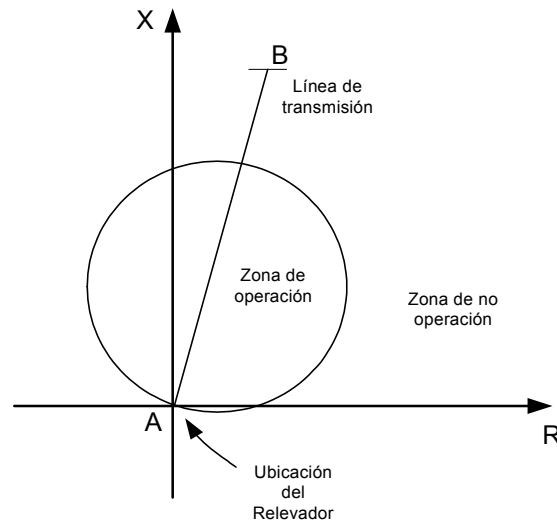


Figura. 2.6. Impedancia de la línea en el plano complejo.

### 2.5. Criterios de ajuste y coordinación de los relevadores de distancia con esquema de protección adaptable.

En un sistema de protección adaptable se requiere que los ajustes de los relevadores se actualicen de acuerdo con las condiciones actuales de operación del sistema, a fin de mantener la coordinación y operación adecuada de los relevadores.

Un esquema de protección adaptable requiere software con capacidad de realizar cálculos de parámetros de ajuste en forma automática y de manera continua, además, el hardware debe ser de alto rendimiento, para obtener información del sistema de potencia y del sistema de protección en el momento requerido [30,33,34].

La característica de operación de los relevadores de distancia se basan en el cociente voltaje/corriente y del ángulo resultante, es decir, la impedancia. Operan cuando esta impedancia medida es menor que la impedancia de ajuste y cae dentro de la zona de operación del relevador.

En condiciones de falla, la impedancia medida por el relevador es proporcional a la longitud de la sección de línea comprendida entre el relevador y el punto de cortocircuito. Esta característica de operación permite que los relevadores de distancia tengan distintos alcances, conocidas como zonas de operación que se

pueden observar en la Figura 2.7., la zona 1 es para proteger a un elemento con protección primaria, y la zona 2 para cubrir la sección de la línea no cubierta por la zona 1, y proporcionar una protección de respaldo, y la zona 3 proporciona una protección de respaldo a la línea donde se ubica el relevador y a las líneas remotas.

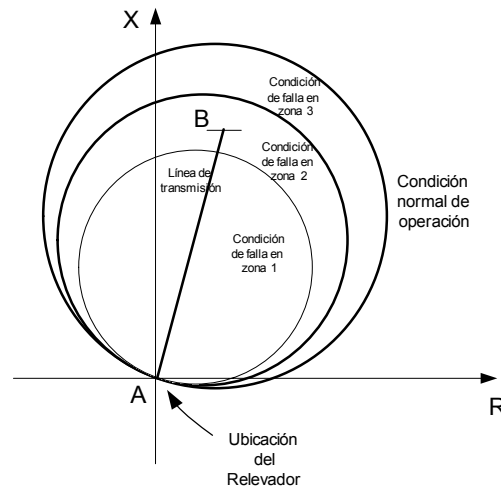


Figura 2.7. Zonas de operación del relevador de distancia.

A continuación se describen en resumen, los ajustes para cada zona de operación en los relevadores de distancia, con características de protección adaptable.

### 2.5.1. Ajuste de la zona 1

En esquemas de protección adaptable, es posible definir el alcance de esta zona hasta en un 95% la impedancia de la línea protegida, como se propone en [28]. En este trabajo el ajuste para dicha zona se observa en la Figura 2.8 y comprenderá un 90% de longitud de la línea A-C donde se ubica el relevador  $R_r$ .

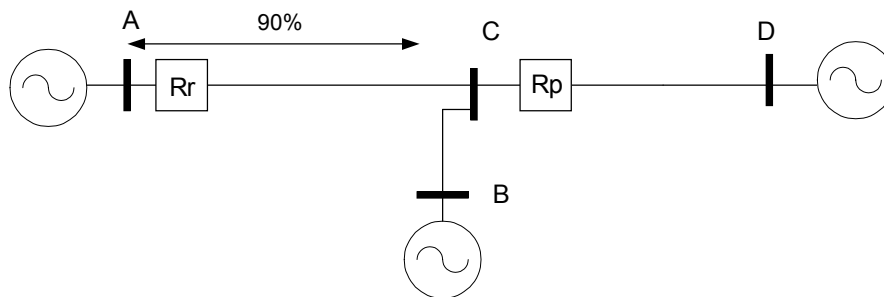


Figura 2.8. Relevador de distancia  $R_r$ , protegiendo en zona 1 a una línea con tres terminales.

### 2.5.2. Ajuste de la Zona 2

La metodología que se utiliza para determinar el alcance de esta zona permite establecer la máxima protección de respaldo, pero sin que se presenten problemas de traslapes de alcance de esta zona con sus relevadores primarios, tomando en cuenta el efecto de fuente intermedia para el cálculo del ajuste. Se realizan distintas evaluaciones para verificar que se cumplan los aspectos de seguridad y dependabilidad que debe proporcionar esta zona de protección.

En un sistema de protección adaptable, la evaluación de la seguridad y dependabilidad debe realizarse continuamente para la condición de operación presente en el sistema, de esta manera es posible identificar los relevadores que requieran modificar su ajuste [35].

El alcance de esta zona pretende cubrir hasta el límite del alcance de la zona 1 de sus relevadores primarios, pero sin llegar a traslapes. Finalmente, se verifica que este ajuste sea cuando menos 1.2 veces la impedancia de línea protegida, como se observa en la Figura 2.9, para garantizar que una falla en su nodo remoto sea “vista” por el relevador en zona 2.

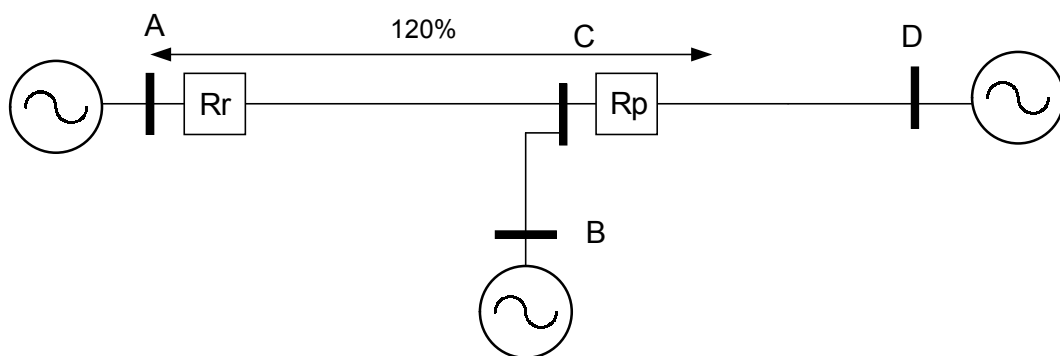


Figura 2.9. Relevador de distancia  $R_r$ , protegiendo en zona 2 a una línea con tres terminales.

El algoritmo de ajuste y coordinación siguiente se describe en [3].

### 2.5.3. Ajuste de la Zona 3

El alcance de esta zona tiene como función brindar una protección de respaldo a líneas remotas en un 100%, es decir, que el alcance debe cubrir hasta dos nodos de distancia de donde se ubica el relevador, como se muestra en la Figura 2.10. el análisis completo de estos ajustes se encuentran en [3].

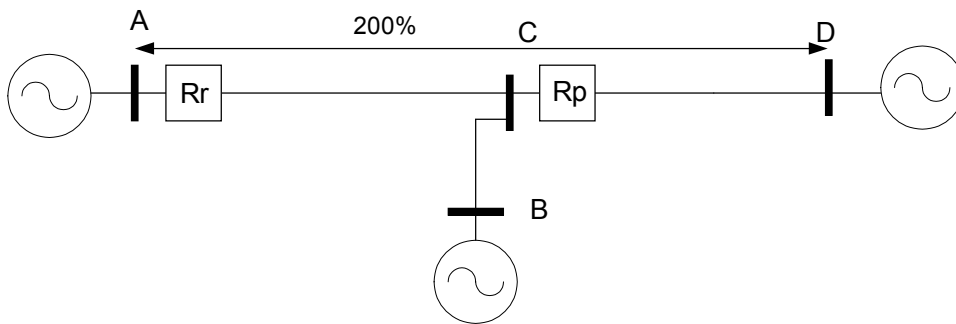


Figura 2.10. Relevador de distancia  $R_r$ , protegiendo en zona 3 a una línea con tres terminales.

### 2.6. Esquema de protección adaptable para relevadores de distancia.

El criterio para desarrollar un esquema de protección adaptable es el descrito en los puntos anteriores como criterios de ajuste, ya que son éstos los parámetros usados para ajustar a un relevador, (cada vez que el sistema así lo requiera) y éste opere para las características que presente el sistema en determinado momento.

De manera general, un sistema de protección adaptable es desarrollado y ajustado para operar con una primera condición; que es la obtenida cuando el sistema está operando con todos sus elementos, si la topología de la red cambia por alguna razón, los equipos de medición y control detectan el cambio, mandan dicha información a una computadora encargada de interactuar con ellos, y si se considera que el cambio en la topología de la red es suficiente como para cambiar los ajustes en los relevadores, entonces es necesario procesar la información para determinar el nuevo ajuste para el o los relevadores que así lo requieran y se envía la información

a los relevadores con los nuevos parámetros de ajuste, para que éstos estén actualizados y no se presenten operaciones indebidas de la protección.

## **2.7. Características de simulación del sistema eléctrico de potencia.**

El sistema de potencia usado para la simulación de fallas, es desarrollado en el software comercial EMTDC/PSCAD [36], el cual tiene la capacidad de simular sistemas eléctricos de potencia en estado transitorio, esta característica lo hace un software excelente para la simulación de fallas, que pueden aplicarse para probar el algoritmo desarrollado con el objetivo de estudiar su comportamiento, ya que cuando ocurre un transitorio, algunos filtros digitales pueden arrojar resultados incorrectos.

En el tema siguiente se presentan algunas ecuaciones que este software utiliza para el modelado de las líneas de transmisión, para mostrar como ejemplo, la forma en que se simulan estos elementos del sistema de potencia.

### **2.7.1 Modelado de las líneas de transmisión en EMTDC/PSCAD.**

El mecanismo de la línea de transmisión distribuida de EMTDC usa un circuito equivalente PI para representar el modelo de Bergeron: Generalmente, las líneas de transmisión consisten en varias fases o conductores mutuamente acoplados. Éstos necesitan estar separados en modos [37]. Hay un modo para cada fase mutuamente acoplada. Cada modo se puede tratar como línea de transmisión monofásica independiente. Este método permite que las líneas mutuamente acopladas desbalanceadas puedan ser modeladas. Una transformación modal convierte la respuesta modal nuevamente dentro de las cantidades de la red de EMTDC. La transformación modal (y las cantidades modales) es obtenida automáticamente por las constantes de la línea de transmisión y del cable con el análisis de eigenvalores.

Para las líneas de transmisión desbalanceadas, cada modo tendrá diversas impedancias y tiempos. Generalmente hay dos tipos de modos principales. En primer

lugar, el modo de tierra o modo común o modo de secuencia cero. Este modo está activo siempre que fluyan las corrientes a tierra. En segundo lugar, los modos restantes se conocen como modos metálicos, modos diferenciados o modos positivos y negativos de secuencia. El modo de tierra tiene una impedancia y una resistencia más alta de la línea que los modos metálicos. El aspecto importante del análisis modal es la matriz modal de transformación. Para resumir este procedimiento, una técnica de diagonalización de la matriz se realiza de tal forma que resulta la ecuación 2.6.

$$\begin{aligned} [V_{fase}] &= [T_{\theta}] [V_{modo}] \\ [I_{fase}] &= [T_I] [I_{modo}] \\ [T_{\theta}]^T &= [T_I]^{-1} \end{aligned} \quad (2.6)$$

donde:

$[V_{modo}]$  = Modo de voltaje al final de la línea.

$[I_{modo}]$  = Modo de corriente al final de la línea.

$[V_{fase}]$  = Voltaje de fase al final de la línea.

$[I_{fase}]$  = Corriente de fase al final de la línea.

$[T_{\theta}]$  = Modo de voltaje para poner en fase la matriz de transformación de voltaje.

$[T_I]$  = Modo de corriente para poner en fase la matriz actual de transformación.

NOTA: El análisis de eigenvalores es utilizado por la rutina de las constantes de la línea de transmisión y del cable para calcular las matrices de transformación.

Para el modo de tierra, la corriente fluye en ambos conductores y a través de la tierra. La corriente fluye abajo de un conductor y detrás del otro para el modo metálico. Cada uno de estos modos puede ser distinguido definiendo la dirección en la cual los flujos actuales como +1 o -1. La matriz modal de transformación se presenta en la ecuación 2.7 y cuando es normalizada resulta la ecuación 2.8.

$$[T_{\theta}] = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad (2.7)$$





$$[T_{\theta}] = [T_I] = \begin{bmatrix} 0.707 & 0.707 \\ 0.707 & -0.707 \end{bmatrix} \quad (2.8)$$

Es interesante observar que la matriz de la transformación para 1 y 2 líneas planas del conductor no cambia con la frecuencia. Esto es porque los conductores son siempre equilibrados. Si una línea trifásica es equilibrada, después la transformación es también constante con la frecuencia. Bajo esas condiciones, la transformación se convierte en componentes de Clark como en la ecuación 2.9 y cuando es normalizado se tiene la ecuación 2.10.

$$[T_{\theta}] = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 2 \\ 1 & -1 & -1 \end{bmatrix} \quad (2.9)$$

$$[T_{\theta}] = [T_I] = \begin{bmatrix} 0.5774 & 0.7071 & -0.4082 \\ 0.5774 & 0 & 0.8165 \\ 0.5774 & -0.7071 & -0.4082 \end{bmatrix} \quad (2.10)$$

## CAPÍTULO III

### **3. DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO DE PROTECCIÓN EN UN MICROCONTROLADOR DSP.**

#### **3.1 Introducción**

La implementación de este algoritmo se realiza en el microcontrolador DSP56807 de Motorola [38], que posee características de ejecutar una instrucción por cada ciclo de máquina. Tomando en cuenta que la velocidad del CPU es de 40 MHz, se puede decir que realiza 40 millones de operaciones por segundo [38,39].

El algoritmo desarrollado requiere de seis señales de entrada a frecuencia fundamental de los voltajes y las corrientes trifásicas. Estas señales son atenuadas por los TC's, TP's y acondicionadores de señales analógicas, antes de ser muestreadas por el convertidor analógico digital (ADC) del microcontrolador, a una frecuencia de 1.44 KHz., esto permite tener una ventana de 24 muestras por cada ciclo de frecuencia fundamental. También es necesario contar con una computadora y licencia para usar el software LabVIEW [23], así como también un programa desarrollado en este lenguaje que se encarga de interactuar con el DSP.

#### **3.2. Algoritmo del filtro digital para el cálculo de fasores implementado en un Microcontrolador DSP.**

Para obtener la impedancia “vista” por los relevadores de distancia, se requiere estimar los fasores de voltaje y corriente de cada una de las fases, estos fasores son usados para calcular la impedancia “vista” la cual es equivalente a la impedancia de secuencia positiva de la línea desde la ubicación del relevador al punto de falla, es decir, la distancia a la cual ocurrió la falla, por lo tanto es posible determinar si la contingencia se encuentra en su zona de operación.

Para obtener estos fasores de voltajes y corrientes, se utiliza un filtro digital; esta técnica de procesamiento de señales digitales se describe a continuación:

El algoritmo de mínimos errores cuadrados (“LES” por sus siglas en ingles) se clasifica como una técnica no recursiva, ya que la salida solo depende de las muestras que representan la ventana actual y no de la salida anterior. Este algoritmo puede funcionar como ventana corta o larga de muestras [40]. Esta técnica de filtro digital es la siguiente:

Primeramente, se presenta la ecuación 3.1. en la que se considera que una muestra o medida satisface dicha ecuación.

$$a + bt = m \tag{3.1}$$

Donde:

$m$  = muestra o medida tomada.

$t$  = es el tiempo.

$a$  y  $b$  = son no conocidas. Son las variables a calcular.

Ahora considerando que se toman “n” muestras se presenta la ecuación 3.2.

$$\begin{aligned} a + bt_1 &= m_1 \\ a + bt_2 &= m_2 \\ &\dots\dots\dots \\ a + bt_n &= m_n \end{aligned} \tag{3.2}$$

Como  $a$  y  $b$  son los valores a estimar, en la ecuación 3.3 se usa la técnica de errores.

$$\begin{aligned} \hat{a} + \hat{b}t_1 - m_1 &= \varepsilon_1 \\ \hat{a} + \hat{b}t_2 - m_2 &= \varepsilon_2 \\ &\dots\dots\dots \\ \hat{a} + \hat{b}t_n - m_n &= \varepsilon_n \end{aligned} \tag{3.3}$$

Y en forma matricial se puede expresar como en la ecuación 3.4.

$$\begin{array}{c}
 \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \cdot & \cdot \\ 1 & t_n \end{bmatrix} \begin{bmatrix} \hat{a} \\ \hat{b} \end{bmatrix} - \begin{bmatrix} m_1 \\ m_2 \\ \cdot \\ m_n \end{bmatrix} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \cdot \\ \varepsilon_n \end{bmatrix} \\
 n \times 2 \quad 2 \times 1 \quad n \times 1 \quad n \times 1 \\
 [A][x] - [m] = [e]
 \end{array} \tag{3.4}$$

Entonces se tiene  $[A][x] - [m] = [e]$ , y elevando el error al cuadrado, resulta :

$$[e]^T [e] = ([A][x] - [m])^T ([A][x] - [m])$$

Desarrollando el lado derecho se tiene que :

$$\begin{aligned}
 [e]^T [e] &= ([A][x])^T ([A][x]) - ([A][x])^T [m] - [m]^T ([A][x]) + [m]^T [m] \\
 &= [x]^T [A]^T [A][x] - 2[x]^T [A]^T [m] + [m]^T [m]
 \end{aligned}$$

Para que la suma de los errores sea mínima, se debe derivar a esta ecuación con respecto a "x" y se iguala a cero, como en la ecuación 3.5.

$$\begin{aligned}
 \frac{d}{d[x]} \{ [x]^T [A]^T [A][x] - 2[x]^T [A]^T [m] + [m]^T [m] \} &= 0 \\
 2[A]^T [A][x] - 2[A]^T [m] &= 0 \\
 [A]^T [A][x] &= [A]^T [m] \\
 [x] &= [A]^T [A]^{-1} [A]^T [m]
 \end{aligned} \tag{3.5}$$

Reescribiendo resulta la ecuación 3.6.

$$\begin{aligned}
 [x] &= [A]^T [A]^{-1} [A]^T [m] \\
 &= [ [A]^T [A]^{-1} [A]^T ] [m] \\
 &= [A]^{1L} [m]
 \end{aligned} \tag{3.6}$$

donde:

$$[A]^{1L} = \text{es la matriz pseudoinversa izquierda de } [A] \text{ dada por } [A]^T [A]^{-1} [A]^T .$$

Si el tiempo de muestreo es conocido, se puede calcular la matriz pseudoinversa izquierda de [A] antes de tomar las muestras. Además este método puede ser usado si las ecuaciones que describen las formas de onda de voltajes y corrientes se pueden expresar en forma lineal.

Analizando una forma de onda senoidal y de frecuencia fundamental se puede expresar como en la ecuación 3.7.

$$v = V_p \sin(\omega t + \theta) \quad (3.7)$$

Donde:

$$\begin{aligned} v &= \text{es la muestra en cuestión.} \\ V_p &= \text{es el voltaje pico de la señal.} \end{aligned}$$

Ahora tomando 3 muestras y discretizando la ecuación 3.7. se presenta la ecuación 3.8.

$$\begin{aligned} v_{-1} &= V_p \cos(\theta) \sin(-\omega\Delta T) + V_p \sin(\theta) \cos(-\omega\Delta T) \\ v_0 &= V_p \cos(\theta) \sin(0) + V_p \sin(\theta) \cos(0) \\ v_{+1} &= V_p \cos(\theta) \sin(\omega\Delta T) + V_p \sin(\theta) \cos(\omega\Delta T) \end{aligned} \quad (3.8)$$

Se asume que la frecuencia es 60 Hz. y que la frecuencia de muestreo es 720 Hz. entonces resulta la ecuación 3.9.

$$\begin{aligned} \omega\Delta T &= \frac{120\pi}{720} = \frac{\pi}{6} \\ v_{-1} &= V_p \cos(\theta) \sin(-\frac{\pi}{6}) + V_p \sin(\theta) \cos(-\frac{\pi}{6}) \\ v_0 &= V_p \cos(\theta) \sin(0) + V_p \sin(\theta) \cos(0) \\ v_{+1} &= V_p \cos(\theta) \sin(\frac{\pi}{6}) + V_p \sin(\theta) \cos(\frac{\pi}{6}) \end{aligned} \quad (3.9)$$

Se expresa la ecuación 3.9. en forma matricial en la ecuación 3.10.

$$\begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ 0 & 1 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_p \cos(\theta) \\ V_p \sin(\theta) \end{bmatrix} = \begin{bmatrix} v_{-1} \\ v_0 \\ v_{+1} \end{bmatrix} \quad (3.10)$$

donde la matriz [A] esta es:

$$[A] = \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ 0 & 1 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

Se calcula la matriz pseudoinversa de [A] ( $[[A]^T [A]]^{-1} [A]^T$ ) en las ecuaciones 3.11 y 3.12.

$$\begin{aligned}
 [A]^T [A] &= \begin{bmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{\sqrt{3}}{2} & 1 & \frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ 0 & 1 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \\
 &= \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 2\frac{1}{2} \end{bmatrix}
 \end{aligned} \tag{3.11}$$

$$\begin{aligned}
 [A]^T &= \begin{bmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{\sqrt{3}}{2} & 1 & \frac{\sqrt{3}}{2} \end{bmatrix}; [A] [A] = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 2\frac{1}{2} \end{bmatrix} \\
 [[A]^T [A]]^{-1} &= \begin{bmatrix} 2 & 0 \\ 0 & \frac{2}{5} \end{bmatrix} \\
 [A]^{-1L} &= \begin{bmatrix} 2 & 0 \\ 0 & \frac{2}{5} \end{bmatrix} \begin{bmatrix} -\frac{1}{2} & 0 & \frac{1}{2} \\ \frac{\sqrt{3}}{2} & 1 & \frac{\sqrt{3}}{2} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ \frac{\sqrt{3}}{5} & \frac{2}{5} & \frac{\sqrt{3}}{5} \end{bmatrix}
 \end{aligned} \tag{3.12}$$

En la ecuación 3.13. se representa la solución para encontrar la parte real e imaginaria de este ejemplo.

$$\begin{bmatrix} V_p \cos(\theta) \\ V_p \sin(\theta) \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 \\ \frac{\sqrt{3}}{5} & \frac{2}{5} & \frac{\sqrt{3}}{5} \end{bmatrix} \begin{bmatrix} v_{-1} \\ v_0 \\ v_{+1} \end{bmatrix} \tag{3.13}$$

Las señales de entrada deben ser senoidales, pero no necesariamente a frecuencia de 60 Hz., ya que en el modelo de este algoritmo puede incluirse cualquier otra frecuencia que sea de interés o que se desee eliminar, por ejemplo: señales de CD o CD decreciente y cualquier señal de frecuencias armónicas.

En este trabajo, el filtro “LES” se realiza considerando una ventana de 24 muestras por ciclo, y tomando las señales de frecuencia fundamental (60 Hz), 2ª, 3ª y 4ª armónica, además de CD, y CD decreciente. Estas señales están modeladas en la ecuación 3.14.

$$\begin{aligned}
 v &= V_{p1} \sin(\omega t + \theta_1) + V_{p2} \sin(2\omega t + \theta_2) + V_{p3} \sin(3\omega t + \theta_3) + \\
 &V_{p4} \sin(4\omega t + \theta_4) + V_0 e^{-\frac{t}{\tau}}
 \end{aligned} \tag{3.14}$$



donde:

$V_{P1}$  = Es el voltaje pico de la fundamental.

$V_{P2}$  = Es el voltaje pico de la 2<sup>a</sup> armónica.

$V_{P3}$  = Es el voltaje pico de la 3<sup>a</sup> armónica.

$V_{P4}$  = Es el voltaje pico de la 4<sup>a</sup> armónica.

$V_0$  = Es el voltaje de CD.

Ahora se considera que la frecuencia de muestreo es 1440 Hz. por lo que al resolver la ecuación 3.15 se presenta el paso entre muestras.

$$\omega\Delta T = \frac{120\pi}{1440} = \frac{\pi}{12} \quad (3.15)$$

Expresando en forma discreta la ecuación 3.14, se tiene, en la ecuación 3.16, el sistema a resolver por el método de errores mínimos cuadrados.

$$\begin{aligned} v_{-12} &= V_{P1} \cos(\theta_1) \sin(-\pi) + V_{P1} \sin(\theta_1) \cos(-\pi) + V_{P2} \cos(\theta_2) \sin(-\pi) + V_{P2} \sin(\theta_2) \cos(-\pi) + \\ &V_{P3} \cos(\theta_3) \sin(-\pi) + V_{P3} \sin(\theta_3) \cos(-\pi) + V_{P4} \cos(\theta_4) \sin(-\pi) + V_{P4} \sin(\theta_4) \cos(-\pi) + I_0 - \frac{I_0}{1440\tau} \\ v_{-11} &= V_{P1} \cos(\theta_1) \sin\left(-\frac{11\pi}{12}\right) + V_{P1} \sin(\theta_1) \cos\left(-\frac{11\pi}{12}\right) + V_{P2} \cos(\theta_2) \sin\left(-\frac{11\pi}{12}\right) + V_{P2} \sin(\theta_2) \cos\left(-\frac{11\pi}{12}\right) + \\ &V_{P3} \cos(\theta_3) \sin\left(-\frac{11\pi}{12}\right) + V_{P3} \sin(\theta_3) \cos\left(-\frac{11\pi}{12}\right) + V_{P4} \cos(\theta_4) \sin\left(-\frac{11\pi}{12}\right) + V_{P4} \sin(\theta_4) \cos\left(-\frac{11\pi}{12}\right) + \\ &I_0 - \frac{I_0}{1440\tau} \\ &\dots\dots \\ v_{-1} &= V_{P1} \cos(\theta_1) \sin\left(-\frac{1\pi}{12}\right) + V_{P1} \sin(\theta_1) \cos\left(-\frac{1\pi}{12}\right) + V_{P2} \cos(\theta_2) \sin\left(-\frac{1\pi}{12}\right) + V_{P2} \sin(\theta_2) \cos\left(-\frac{1\pi}{12}\right) + \\ &V_{P3} \cos(\theta_3) \sin\left(-\frac{1\pi}{12}\right) + V_{P3} \sin(\theta_3) \cos\left(-\frac{1\pi}{12}\right) + V_{P4} \cos(\theta_4) \sin\left(-\frac{1\pi}{12}\right) + V_{P4} \sin(\theta_4) \cos\left(-\frac{1\pi}{12}\right) + \\ &I_0 - \frac{I_0}{1440\tau} \\ v_0 &= V_{P1} \cos(\theta_1) \sin(0) + V_{P1} \sin(\theta_1) \cos(0) + V_{P2} \cos(\theta_2) \sin(0) + V_{P2} \sin(\theta_2) \cos(0) + \\ &V_{P3} \cos(\theta_3) \sin(0) + V_{P3} \sin(\theta_3) \cos(0) + V_{P4} \cos(\theta_4) \sin(0) + V_{P4} \sin(\theta_4) \cos(0) + I_0 - \frac{I_0}{1440\tau} \\ &\dots(3.16) \end{aligned}$$



$$v_{+1} = V_{P1} \cos(\theta_1) \sin\left(+\frac{1\pi}{12}\right) + V_{P1} \sin(\theta_1) \cos\left(+\frac{1\pi}{12}\right) + V_{P2} \cos(\theta_2) \sin\left(+\frac{1\pi}{12}\right) + V_{P2} \sin(\theta_2) \cos\left(+\frac{1\pi}{12}\right) +$$

$$V_{P3} \cos(\theta_3) \sin\left(+\frac{1\pi}{12}\right) + V_{P3} \sin(\theta_3) \cos\left(+\frac{1\pi}{12}\right) + V_{P4} \cos(\theta_4) \sin\left(+\frac{1\pi}{12}\right) + V_{P4} \sin(\theta_4) \cos\left(+\frac{1\pi}{12}\right) +$$

$$I_0 - \frac{I_0}{1440\tau}$$

.....

$$v_{+10} = V_{P1} \cos(\theta_1) \sin\left(+\frac{5\pi}{6}\right) + V_{P1} \sin(\theta_1) \cos\left(+\frac{5\pi}{6}\right) + V_{P2} \cos(\theta_2) \sin\left(+\frac{5\pi}{6}\right) + V_{P2} \sin(\theta_2) \cos\left(+\frac{5\pi}{6}\right) +$$

$$V_{P3} \cos(\theta_3) \sin\left(+\frac{5\pi}{6}\right) + V_{P3} \sin(\theta_3) \cos\left(+\frac{5\pi}{6}\right) + V_{P4} \cos(\theta_4) \sin\left(+\frac{5\pi}{6}\right) + V_{P4} \sin(\theta_4) \cos\left(+\frac{5\pi}{6}\right) +$$

$$I_0 - \frac{I_0}{1440\tau}$$

$$v_{+11} = V_{P1} \cos(\theta_1) \sin\left(+\frac{11\pi}{12}\right) + V_{P1} \sin(\theta_1) \cos\left(+\frac{11\pi}{12}\right) + V_{P2} \cos(\theta_2) \sin\left(+\frac{11\pi}{12}\right) + V_{P2} \sin(\theta_2) \cos\left(+\frac{11\pi}{12}\right) +$$

$$V_{P3} \cos(\theta_3) \sin\left(+\frac{11\pi}{12}\right) + V_{P3} \sin(\theta_3) \cos\left(+\frac{11\pi}{12}\right) + V_{P4} \cos(\theta_4) \sin\left(+\frac{11\pi}{12}\right) + V_{P4} \sin(\theta_4) \cos\left(+\frac{11\pi}{12}\right) +$$

$$I_0 - \frac{I_0}{1440\tau}$$

....(3.16 cont.)

Este sistema es presentado en su forma matricial en la ecuación 3.17, para identificar la matriz A, encontrar la matriz pseudoinversa izquierda de ésta y resolver el sistema. La solución se realiza por medio de un programa de cómputo escrito en lenguaje de MATLAB, ver apéndice E.





$$\begin{bmatrix}
 -1.0000 & -0.0000 & 1.0000 & 0.0000 & -1.0000 & -0.0000 & 1.0000 & 0.0000 & 1.0000 & -12.0000 \\
 -0.9659 & -0.2588 & 0.8660 & 0.5000 & -0.7071 & -0.7071 & 0.5000 & 0.8660 & 1.0000 & -11.0000 \\
 -0.8660 & -0.5000 & 0.5000 & 0.8660 & 0.0000 & -1.0000 & -0.5000 & 0.8660 & 1.0000 & -10.0000 \\
 -0.7071 & -0.7071 & -0.0000 & 1.0000 & 0.7071 & -0.7071 & -1.0000 & -0.0000 & 1.0000 & -9.0000 \\
 -0.5000 & -0.8660 & -0.5000 & 0.8660 & 1.0000 & 0.0000 & -0.5000 & -0.8660 & 1.0000 & -8.0000 \\
 -0.2588 & -0.9659 & -0.8660 & 0.5000 & 0.7071 & 0.7071 & 0.5000 & -0.8660 & 1.0000 & -7.0000 \\
 0.0000 & -1.0000 & -1.0000 & -0.0000 & -0.0000 & 1.0000 & 1.0000 & 0.0000 & 1.0000 & -6.0000 \\
 0.2588 & -0.9659 & -0.8660 & -0.5000 & -0.7071 & 0.7071 & 0.5000 & 0.8660 & 1.0000 & -5.0000 \\
 0.5000 & -0.8660 & -0.5000 & -0.8660 & -1.0000 & -0.0000 & -0.5000 & 0.8660 & 1.0000 & -4.0000 \\
 0.7071 & -0.7071 & 0.0000 & -1.0000 & -0.7071 & -0.7071 & -1.0000 & -0.0000 & 1.0000 & -3.0000 \\
 0.8660 & -0.5000 & 0.5000 & -0.8660 & 0.0000 & -1.0000 & -0.5000 & -0.8660 & 1.0000 & -2.0000 \\
 0.9659 & -0.2588 & 0.8660 & -0.5000 & 0.7071 & -0.7071 & 0.5000 & -0.8660 & 1.0000 & -1.0000 \\
 1.0000 & 0 & 1.0000 & 0 & 1.0000 & 0 & 1.0000 & 0 & 1.0000 & 0 \\
 0.9659 & 0.2588 & 0.8660 & 0.5000 & 0.7071 & 0.7071 & 0.5000 & 0.8660 & 1.0000 & 1.0000 \\
 0.8660 & 0.5000 & 0.5000 & 0.8660 & 0.0000 & 1.0000 & -0.5000 & 0.8660 & 1.0000 & 2.0000 \\
 0.7071 & 0.7071 & 0.0000 & 1.0000 & -0.7071 & 0.7071 & -1.0000 & 0.0000 & 1.0000 & 3.0000 \\
 0.5000 & 0.8660 & -0.5000 & 0.8660 & -1.0000 & 0.0000 & -0.5000 & -0.8660 & 1.0000 & 4.0000 \\
 0.2588 & 0.9659 & -0.8660 & 0.5000 & -0.7071 & -0.7071 & 0.5000 & -0.8660 & 1.0000 & 5.0000 \\
 0.0000 & 1.0000 & -1.0000 & 0.0000 & -0.0000 & -1.0000 & 1.0000 & -0.0000 & 1.0000 & 6.0000 \\
 -0.2588 & 0.9659 & -0.8660 & -0.5000 & 0.7071 & -0.7071 & 0.5000 & 0.8660 & 1.0000 & 7.0000 \\
 -0.5000 & 0.8660 & -0.5000 & -0.8660 & 1.0000 & -0.0000 & -0.5000 & 0.8660 & 1.0000 & 8.0000 \\
 -0.7071 & 0.7071 & -0.0000 & -1.0000 & 0.7071 & 0.7071 & -1.0000 & 0.0000 & 1.0000 & 9.0000 \\
 -0.8660 & 0.5000 & 0.5000 & -0.8660 & 0.0000 & 1.0000 & -0.5000 & -0.8660 & 1.0000 & 10.0000 \\
 -0.9659 & 0.2588 & 0.8660 & -0.5000 & -0.7071 & 0.7071 & 0.5000 & -0.8660 & 1.0000 & 11.0000
 \end{bmatrix}
 \begin{bmatrix}
 V_{P1} \cos(\theta_1) \\
 V_{P1} \sin(\theta_1) \\
 V_{P2} \cos(\theta_2) \\
 V_{P2} \sin(\theta_2) \\
 V_{P3} \cos(\theta_1) \\
 V_{P3} \cos(\theta_1) \\
 V_{P4} \cos(\theta_1) \\
 V_{P4} \cos(\theta_1) \\
 \frac{I_0}{1440 \tau}
 \end{bmatrix}
 =
 \begin{bmatrix}
 v_{-12} \\
 v_{-11} \\
 v_{-10} \\
 v_{-9} \\
 v_{-8} \\
 v_{-7} \\
 v_{-6} \\
 v_{-5} \\
 v_{-4} \\
 v_{-3} \\
 v_{-2} \\
 v_{-1} \\
 v_0 \\
 v_{+1} \\
 v_{+2} \\
 v_{+3} \\
 v_{+4} \\
 v_{+5} \\
 v_{+6} \\
 v_{+7} \\
 v_{+8} \\
 v_{+9} \\
 v_{+10} \\
 v_{+11}
 \end{bmatrix}$$

...(3.17)

Este algoritmo presenta las ventajas y desventajas siguientes.

#### Ventajas.

- Al aumentar los componentes en el modelo del filtro digital se obtienen los coeficientes para estimar la magnitud de una frecuencia en particular, que haya sido incluida en dicho modelo (regularmente armónicos y señal de CD decreciente).
- Elimina los armónicos que no sean de interés, pero incluidos en el modelo del filtro digital.
- Respuesta aceptable ante la componente de CD decreciente, cuando está incluida en el modelo.

### Desventajas.

- Mayor número de muestras para el cálculo de los fasores, comparado con los algoritmos trigonométricos, que solo requieren de 3 muestras.
- Por consiguiente, mayor número de cálculos.

Cabe mencionar que para formar la parte real e imaginaria de los fasores de voltaje y corriente a frecuencia fundamental, solo se toman en cuenta los coeficientes que corresponden a dicha frecuencia, por lo que en la tabla 3.1 solo se muestran dos columnas de coeficientes, correspondientes a la frecuencia fundamental.

*Tabla 3.1. Coeficientes del filtro digital de mínimos errores cuadrados.*

Coeficientes para la parte real	Coeficientes para la parte imaginaria
0.4171	-0.0284
0.0019	-0.0774
-0.1701	-0.0891
-0.1318	-0.0685
-0.0338	-0.0366
-0.0072	-0.0119
-0.0656	0.0023
-0.1275	0.0154
-0.1192	0.0355
-0.0504	0.0601
0.0076	0.0787
0.0062	0.0842
-0.0278	0.0797
-0.0277	0.0740
0.0331	0.0710
0.1060	0.0651
0.1192	0.0479
0.0628	0.0192
0.0100	-0.0097
0.0421	-0.0266
0.1450	-0.0321
0.1874	-0.0420
0.0182	-0.0753
-0.3956	-0.1354

Éstos coeficientes se utilizan para determinar los fasores de voltaje y corriente durante un ciclo completo de la señal a la frecuencia fundamental, posteriormente, el fasor voltaje se divide entre el fasor corriente, ambos de la misma fase; de esta manera se obtiene la impedancia de falla en ese ciclo [41]. Este concepto es válido cuando se presenta el caso de una falla trifásica sólida, porque la impedancia de falla medida resultante, corresponde a la impedancia de secuencia positiva al punto de falla.

Para el caso de una falla monofásica, se tiene que realizar la transformación de las componentes de fase de la corriente a las componentes de secuencia, esto es necesario para compensar el fasor de corriente de la fase fallada y así, obtener el valor correcto de la impedancia de falla, la cual corresponde a la impedancia de secuencia positiva de la línea, medida desde la ubicación del relevador, hasta el punto de falla, si la resistencia de falla es de cero Ohms.

El término de fasores, se refiere a vectores en el plano complejo que tienen una magnitud y un ángulo (forma polar) y se mueven en el tiempo, los cuales también pueden representarse como dos elementos, parte real y parte imaginaria (forma rectangular). Usando estos conceptos se presenta el cálculo de los fasores de voltaje y corriente de la fase A.

La magnitud del fasor voltaje de la fase A se obtiene a partir de la ecuación 3.18.

$$VA_m = \sqrt{reVA^2 + imVA^2} \quad (3.18)$$

donde:

$VA_m$  =es la magnitud del fasor de voltaje de la fase A.

$reVA$  =es la parte real del fasor de voltaje de la fase A.

$imVA$ =es la parte imaginaria del fasor de voltaje de la fase A.

Los componentes de la ecuación 3.18, están dados por las ecuaciones 3.19 y 3.20

$$reVA = coefseno[ ] * muestrasVA[ ] \quad (3.19)$$

$$imVA = coefcoseno[ ] * muestrasVA[ ] \quad (3.20)$$

donde:

$coefseno[ ]$  = es el vector de coeficientes del filtro digital que forma la parte real del fasor.

$coefcoseno[ ]$  = es el vector de coeficientes del filtro digital que forma la parte imaginaria del fasor.

$muestrasVA[ ]$  = son las muestras de un ciclo completo de la onda senoidal a la frecuencia fundamental de voltaje de la fase A.

De igual manera se obtiene la magnitud del fasor corriente evaluando la ecuación 3.21.

$$IA_m = \sqrt{reIA^2 + imIA^2} \quad (3.21)$$

$$reIA = coefseno[ ] * muestrasIA[ ] \quad (3.22)$$

$$imIA = coefcoseno[ ] * muestrasIA[ ] \quad (3.23)$$

donde:

$IA_m$  = es la magnitud del fasor de corriente de la fase A.

$reIA$  = es la parte real del fasor de corriente de la fase A.

$imIA$  = es la parte imaginaria del fasor de corriente de la fase A.

$muestrasIA[ ]$  = son las muestras de un ciclo completo de la onda senoidal a la frecuencia fundamental de corriente de la fase A.

De esta misma forma, se obtienen los fasores de voltaje y corriente para cada una de las otras fases (B y C).

Para obtener la impedancia de falla de secuencia positiva ( $Z_{1falla}$ ) medida a partir de las señales de voltaje y corriente en una condición de falla monofásica, por ejemplo una falla en la fase A, es necesario evaluar la ecuación 3.24, en la que el fasor de voltaje de la fase A, se divide entre el fasor de corriente de la misma fase, pero compensado, para obtener la medición correcta de impedancia de secuencia positiva desde la ubicación del relevador hasta el punto de falla.

$$Z_{1falla} = \frac{VA}{IAx} \quad (3.24)$$

donde:

$Z_{1falla}$  = es la impedancia de falla de secuencia positiva medida.

$VA$  = es el fasor de voltaje de la fase A.

$IAx$  = es el fasor de corriente compensado de la fase A.

El fasor de corriente compensado de la fase A ( $IAx$ ), se calcula con la ecuación 3.25.

$$IAx = IA + I_0 * m \quad (3.25)$$

donde:

$IA$  = es el fasor de corriente de la fase A.

$I_0$  = es el fasor de corriente de secuencia cero.

$m$  = es una constante compleja, ver ecuación 3.27.

La transformación de componentes de fase a componentes de secuencia, se utiliza para obtener la corriente de secuencia cero ( $I_0$ ) como se presenta en la ecuación 3.26.

$$I_0 = \frac{IA + IB + IC}{3} \quad (3.26)$$

donde:

$IA, IB, IC$  = son los fasores de corriente de cada una de las fases A, B y C.

La constante compleja  $m$  se obtiene evaluando la ecuación 3.27.

$$m = \frac{Z_0 - Z_1}{Z_1} \quad (3.27)$$

donde:

$Z_0$  = es la impedancia de secuencia cero dada por la longitud de la línea.

$Z_1$  = impedancia de secuencia positiva dada por la longitud de la línea.

En la ecuación 3.27,  $Z_1$  es la impedancia de secuencia positiva de toda la línea, dada por su longitud, y en la ecuación 3.24,  $Z_{1falla}$  es la impedancia de falla de secuencia positiva desde la ubicación del relevador hasta el punto de falla.

### 3.3. Identificación de la ubicación de fallas en el plano complejo $R+jX$ .

Para identificar la ubicación de las fallas se puede observar la Figura 3.1, en la que se puede apreciar como el algoritmo identifica si existe una falla en su zona de operación. Donde el vector  $ZD$  representa un medio del ajuste dado al DSP para la zona 1 y cuya magnitud es obtenida por el microcontrolador,  $ZF$  es el vector de la impedancia de secuencia positiva medida y  $ZE$  es el vector resultante de la diferencia  $ZF-ZD$ , desplazado al centro del origen.

Cuando una impedancia medida por el DSP se encuentra dentro del círculo de operación de éste, se genera una señal para el disparo del interruptor de la línea, encargado de liberar la falla del sistema. El DSP mide lo siguiente:

Si se considera que la impedancia de la línea a proteger es  $ZL$  y se desea cubrir el 90% de la misma (Zona 1), entonces el radio del círculo ( $ZD$ ) se obtiene con la ecuación 3.28. Para identificar el centro del círculo se evalúan las ecuaciones 3.29a y 3.29b.

$$ZD = \frac{1}{2} Zona 1 \quad (3.28)$$

$$Z0r = ZD * \text{coseno}(\text{ang}ZL) \quad (3.29a)$$

$$Z0j = ZD * \text{seno}(\text{ang}ZL) \quad (3.29b)$$

donde:

$Z0r$  = es la parte real del vector  $ZD$ .

$Z0j$  = es la parte imaginaria del vector  $ZD$ .

$\text{ang}ZL$  = ángulo de la impedancia de secuencia positiva de la línea  $ZL$ .

**Zona 1** = es el porcentaje de la línea a proteger.

Entonces la magnitud del vector  $ZD$  se obtiene en la ecuación 3.30:

$$ZD = \sqrt{(Z0r^2 + Z0j^2)} \quad (3.30)$$

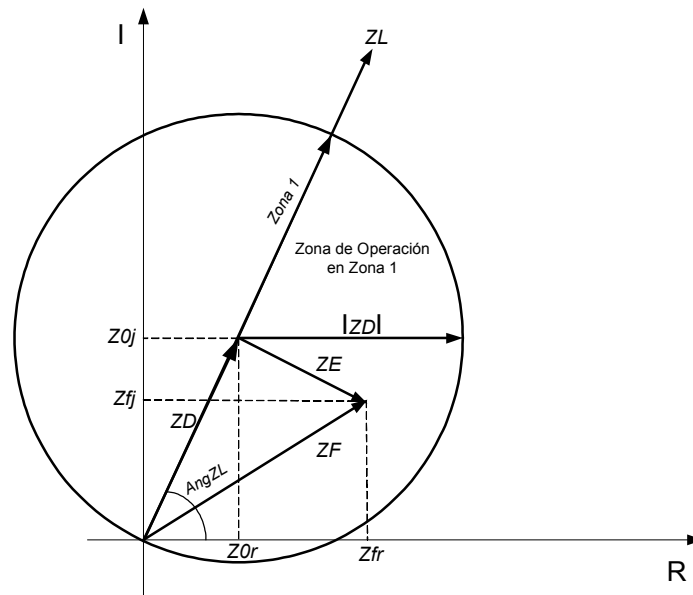


Figura 3.1. Característica de distancia tipo mho en zona 1, para el DSP

Una vez que **ZF** es calculada por el microcontrolador, como quedó demostrado en la ecuación 3.24, posteriormente se obtiene un nuevo vector de impedancia (**ZE**), que parte del centro de la característica de operación y que es la diferencia entre **ZF** y **ZD**, como se muestra en la ecuación 3.31.

$$ZE = ZF - ZD \quad (3.31)$$

El microcontrolador realiza esta operación en forma rectangular, como se presenta la ecuación 3.32.

$$ZE = (Zfr - Z0r) + j(Zfj - Z0j) \quad (3.32)$$

donde:

**ZE** = es la impedancia esperada.

**Z0r** = es la parte real de la impedancia de ajuste, dada desde la PC como ajuste.

**Z0j** = es la parte imaginaria de la impedancia de ajuste, dada desde la PC como ajuste.

En la ecuación 3.33 se muestra el resultado sustituyendo coeficientes.

$$ZE = ZEr + ZEj \quad (3.33)$$

donde:

$$ZEr = Zfr - Z0r$$

$$ZEj = Zfj - Z0j$$

y por último, en la ecuación 3.34 se obtiene la magnitud de la impedancia esperada ( $ZE$ ).

$$ZE = \sqrt{ZEr^2 + ZEj^2} \quad (3.34)$$

Para saber si la impedancia medida está dentro del área del círculo de operación del algoritmo, se compara con la magnitud del radio ( $ZD$ ) de la zona de operación, propuesta en la ecuación 3.30.

Así:

si  $ZE < ZD$  entonces es una falla dentro de la zona de operación.

Si una condición de falla es evaluada dentro de la zona de operación por el algoritmo implementado en el  $\mu c$  DSP, éste guarda las muestras de voltajes y corrientes correspondientes a un ciclo completo de posfalla de cada fase. Estas cantidades pueden ser descargadas en la PC para su análisis posterior.

### **3.4. Implementación del esquema propuesto en el DSP56807 de Motorola.**

La implementación del algoritmo para obtener los fasores de voltajes y corrientes, así como el algoritmo de protección, se ejecuta al descargar un programa en el Microcontrolador DSP56807 de la familia DSP56800 de motorola [38,39], por medio de una PC usando el software Code Warrior de la compañía Metrowerks [42] (ver apéndice B). El ambiente del programa computacional es una combinación de lenguaje C y ensamblador (ver apéndice C), con algunas restricciones tales como: no



puede imprimir en pantalla, necesita de subrutinas para aplicar funciones seno, coseno, tangente, y algunas otras.

### **3.4.1. Características del DSP56807 de Motorola.**

El DSP56807 es un Procesador de Señales Digitales que tiene la característica de realizar sumas de productos en un ciclo de máquina. El núcleo del DSP56800 tiene una Unidad Central de Procesamiento (CPU) como propósito general, diseñada para un eficiente procesamiento de señales digitales y una variedad de operaciones de control [38,39].

El DSP56807 tiene un Procesador de Señales Digitales programable de 16 BITS, este consiste de una Unidad de Aritmética y Lógica de 16 Bits (DATA ALU), una Unidad Generadora de Direcciones de 16 Bits (AGU), un Decodificador de Programas, *On-Chip Emulation* (ONCE), Buses y grupo de Instrucciones.

La arquitectura del DSP56807 es de tipo Harvard, esto significa que tiene de manera separada la memoria de datos con la memoria de programas, esta característica le permite realizar al mismo tiempo la ejecución de una instrucción y guardar o leer información de la memoria de datos, obteniendo con esto, un mejor desempeño en cuanto a velocidad del procesamiento.

Las principales características del DSP56807 son las siguientes:

- Capacidad de procesamiento de 40 millones de instrucciones por segundo (MIPS) a 80 MHz.
- Requiere solo 2.7-3.6 Volts.
- Paralelamente multiplica y acumula en un simple ciclo de instrucción 16 Bits x 16 Bits.
- Registros de corrimientos de simple instrucción de 16 Bits.
- Tiene dos acumuladores de 36 Bits que incluyen extensión de Bits.
- Grupo de instrucciones paralelas con único modo de direccionamiento del DSP.
- Modo de direccionamiento paralelo.
- Instrucciones de bucles mediante Hardware DO y REP.
- Requiere dos pines externos de interrupción.
- Tres buses de datos de 16 Bits.

- Tres buses de direccionamiento de 16 Bits.
- Opera con frecuencias bajas cercanas a CD.
- Eficiente compilador en Lenguaje C y soporte de variables locales.
- Subrutinas de software e interruptor de apilamiento.

Algunas otras características del microcontrolador DSP56807 de Motorola son presentadas a continuación:

Cuenta con dos Convertidores Analógico Digital (ADC), ambos con ocho canales de entrada cada uno, es decir, que puede muestrear hasta 16 señales diferentes, con la capacidad de tomar muestras a frecuencias hasta de 2.5 MHz. (con algunas restricciones en cuanto a la cantidad de operaciones a las que estas señales estén sometidas), dos canales de PWM con ocho salidas cada uno, comunicación serial con RS232, entre otras (apéndice A). El puerto de comunicación serial se emplea para recibir la información de los nuevos ajustes obtenidos para satisfacer la filosofía de esquemas de protección adaptable. Esta información es transmitida a una PC encargada de interactuar con el DSP.

#### **3.4.2. Operación del programa implementado en el microcontrolador.**

La operación del algoritmo desarrollado e implementado en el microcontrolador, se puede explicar con la ayuda del diagrama de flujo del programa, como se puede ver en las Figuras 3.2, 3.3, 3.4 y 3.5. La razón por la que se encuentra en 4 secciones es debido a que la operación del ADC, la transmisión y recepción de datos se realiza por medio de interrupciones o banderas, es decir, que cuando una de estas funciones ha terminado de realizar sus operaciones, activa una bandera que alerta al programa principal o a las otras funciones, para que éstas realicen una o varias operaciones más. El empleo de interrupciones permite al microcontrolador hacer operaciones de manera separada, por ejemplo, si en el programa principal se están calculando los fasores, el ADC puede estar capturando las siguientes muestras sin que esto afecte al primero.

En el diagrama de la Figura 3.2. muestra la operación del programa principal, donde se realizan la formación de fasores, el cálculo de la impedancia de secuencia positiva

que se usa para la detección de fallas y la generación de la señal de operación del interruptor.

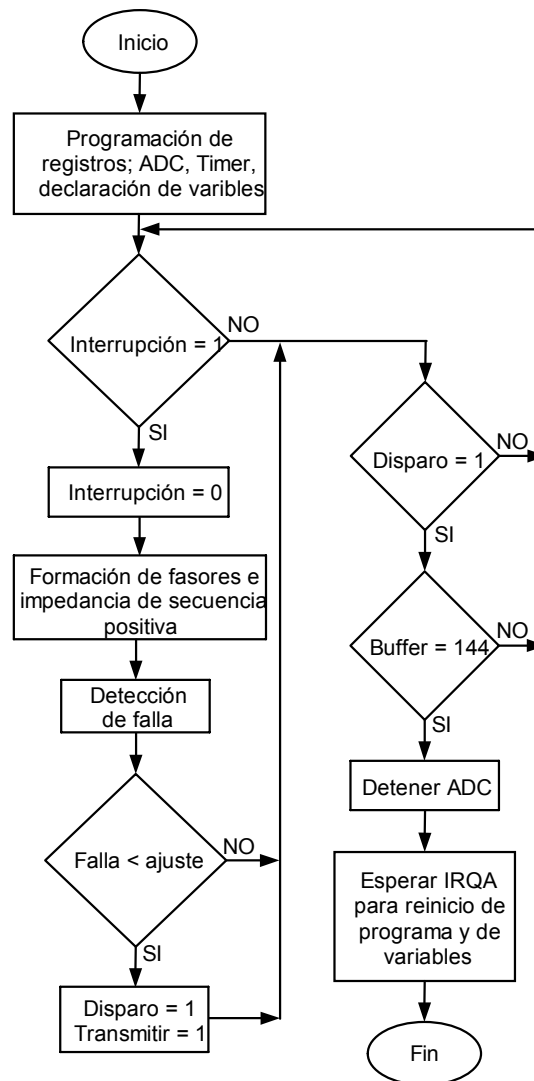


Figura 3.2. Programa principal del algoritmo.

La función de transmisión de datos se puede ver la Figura 3.3. En este diagrama se observa que se requiere que la variable *Envío* sea igual a 1 (activada en la función del ADC desde la función de recepción de datos, a través del comando “enter”), para que se realice la transmisión de datos. Una vez realizada, se detiene la transmisión y si la variable *Disparo* es diferente de 1, se activa el ADC, puesto que éste se detiene cuando se transmite, por lo tanto, se requiere después de esto, su reactivación. En

caso que  $Disparo = 1$ , entonces no se reactiva al ADC, porque no es necesario la captura de muestras, ya que existe la señal de activación del interruptor, además deben guardarse las muestras capturadas antes del disparo del interruptor.

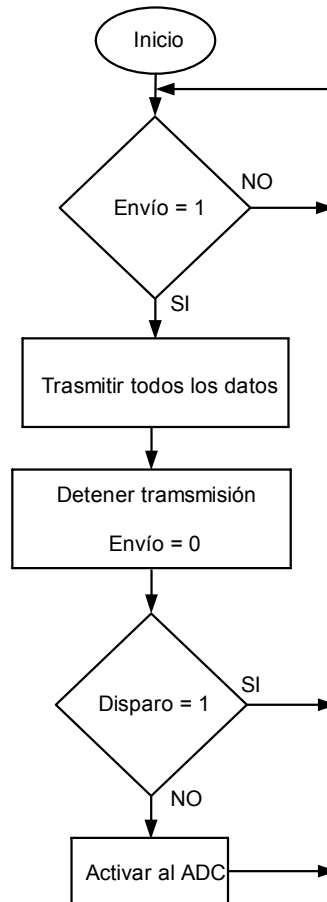


Figura 3.3. Transmisión de datos.

Para explicar el funcionamiento del ADC se presenta el diagrama de flujo de la Figura 3.4., donde puede observarse que se requiere que la variable *Timer* sea igual a 1 (esta variable se activa automáticamente al programar la operación del ADC con pulso de sincronía del temporizador) que depende de la frecuencia de muestreo programada. Cada vez que se activa esta bandera, el ADC hace la captura de 6 muestras (una de cada fase de voltaje y corriente) y se incrementan los contadores *Muestra* y *Buffer* que corresponden a dos arreglos, el primer arreglo se completa con 24 muestras para indicarle al programa principal que se tiene un ciclo completo a la frecuencia fundamental y pueda así realizar el cálculo de la impedancia con estas 24

muestras, el segundo arreglo se completa con 6 ciclos correspondientes a un ciclo de cada fase de voltaje y corriente (144 muestras), los que son necesarios para el envío de datos.

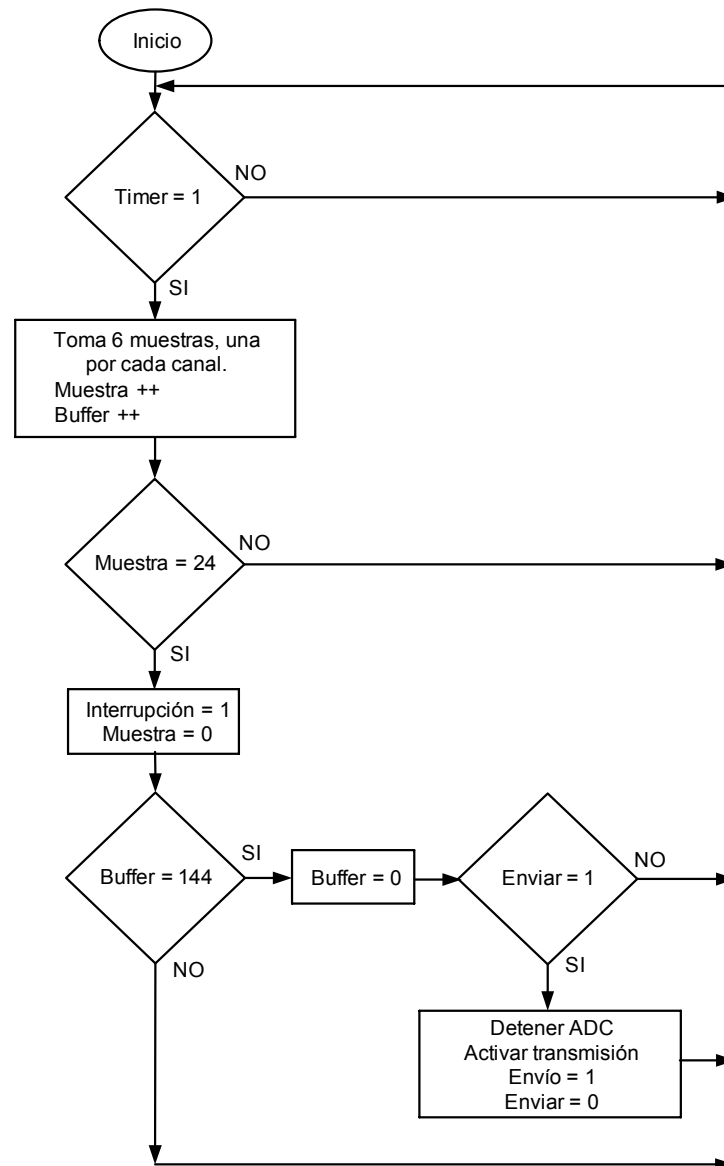


Figura 3.4. Operación del ADC.

Una vez que el arreglo de 6 ciclos de muestras a la frecuencia fundamental es completado, y si la condición siguiente se cumple ( $Envío = 1$ ), entonces se detiene al ADC para activar la transmisión de datos. Al completarse esta transmisión

inmediatamente después, se vuelve a activar el ADC para continuar con la captura de muestras.

La función para la recepción de datos se puede observar en el diagrama de flujo de la Figura 3.5. Se observa que es necesaria la existencia de un caracter en el puerto de comunicación serial del microcontrolador para iniciar la recepción.

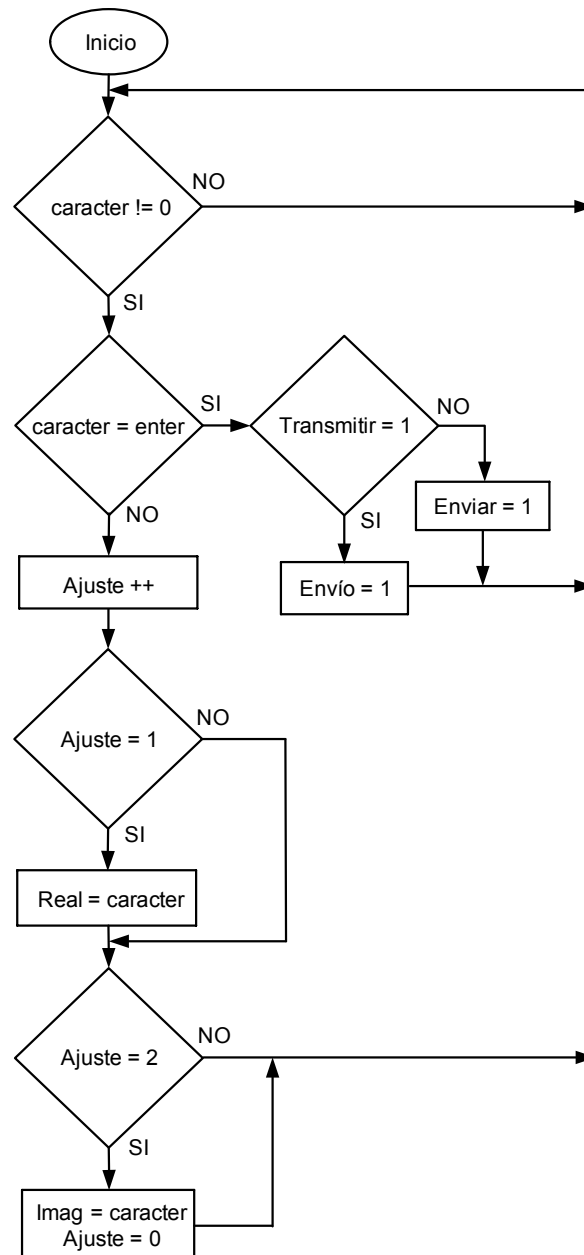


Figura 3.5. Función de recepción de datos

Este caracter es enviado a través del programa desarrollado en LabVIEW, en ASCII y una vez recibido por el DSP, éste lo convierte al correspondiente valor numérico. Explicando este proceso de forma más detallada, sucede lo siguiente: Primero, se dan algunas restricciones: para representar la impedancia de ajuste, esto es, se proporciona la parte real y la imaginaria, los valores de la parte real van de 0 a 10 con variación de 0.1, y los valores de la parte imaginaria van de 0 a 100 con variación de 1 (esto porque regularmente la parte real de la impedancia de una línea de transmisión difícilmente pasará el valor de 10 Ohms y de la misma forma con la parte imaginaria, regularmente no pasa de 100 Ohms).

Las acciones realizadas con el programa de LabVIEW son la conversión del valor introducido como ajuste (dado por el usuario), a su correspondiente ASCII y después enviarlo al DSP, este último, recibe el caracter ASCII y lo convierte al valor decimal de ajuste. El primer valor enviado es la parte real y el segundo es la parte imaginaria. Por ejemplo, se desea programar un nuevo ajuste de impedancia en el DSP con valor de  $1.5 + j 37$  Ohms, primero se escribe el valor de 1.5 en el programa de LabVIEW (este valor es multiplicado por 10 y convertido a ASCII, esto porque solo pueden ser enviados valores entre 0 y 100) y se envía al DSP, éste lo recibe y actualiza la parte real (el valor ASCII se pasa a decimal y se divide entre 10, teniendo así el mismo valor de ajuste de 1.5). Después se hace el mismo procedimiento y se actualiza la parte imaginaria (aquí el valor no es multiplicado por ningún factor).

Pero si el caracter es el comando “enter”, esto indica que no se desea cambiar el ajuste sino transmitir los datos que se tengan cuando se completa el arreglo de 144 muestras, esto es la condición *Transmitir* = 1, si esta no se cumple, entonces el ADC no se detiene porque no existe una condición de falla, de esta forma se hace el envío de datos sin detener al ADC.

### **3.5. Adquisición de datos o muestreo de señales del sistema eléctrico de potencia.**

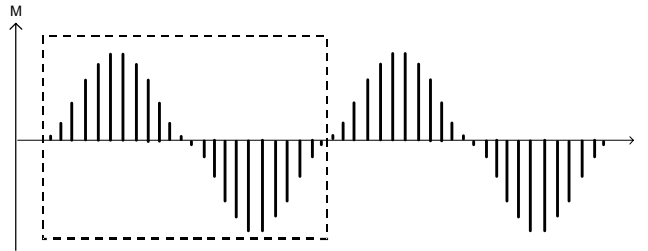
Un aspecto importante para el desarrollo de este trabajo es la adquisición de datos por el microcontrolador, esto es la toma de muestras de las señales analógicas en tiempo continuo que se desean medir, para representarlas en forma de señales en tiempo discreto, posteriormente son procesadas con filtros digitales para poder formar los fasores de dicha señal. Estos fasores serán utilizados para calcular la impedancia de secuencia positiva resultante del cociente voltaje/corriente.

En este trabajo, se desea estudiar el comportamiento del algoritmo implementado ante diferentes tipos de fallas ocurridas en la línea a proteger, de esta manera se puede evaluar la funcionalidad del algoritmo de protección y el tiempo de respuesta del microcontrolador después de realizar todas las operaciones, desde la captura de muestras, hasta el momento en que se genera la señal de disparo del interruptor, cuando una falla dada cae en la zona de operación programada en el microcontrolador.

En particular, las muestras son tomadas de señales senoidales de frecuencia fundamental (60 Hz) de voltaje y corriente, adecuadas para ser leídas por el microcontrolador, estas muestras son obtenidas mediante la simulación de fallas en estado transitorio, para que las pruebas realizadas sean las requeridas y así, verificar que el diseño e implementación del algoritmo de protección digital, opera correctamente como un relevador de distancia.

La adquisición de datos por el DSP se realiza a través del muestreo de las señales de voltaje y corriente procedente de la línea de transmisión a proteger, a través de los transformadores de corriente (TC's) y transformadores de potencial (TP's). Las señales de los TC's y TP's pasadas por elementos que acondicionan estas señales para ser leídas por el convertidor analógico digital (ADC) del microcontrolador, este último, programado para muestrear a una frecuencia de 1.44 KHz., es decir, tomando 24 muestras por ciclo de frecuencia fundamental, como se observa en la Figura 3.6.





*Figura. 3.6 Ventana de 24 muestras para un ciclo completo de frecuencia fundamental.*

Es importante aclarar que debido a las restricciones de poder realizar fallas y tomar las señales de un sistema de potencia real, se tienen que realizar algunos pasos para hacer funcionar al microcontrolador con simulaciones del sistema, para garantizar una medición segura y correcta. Los pasos son los siguientes:

1. Hacer simulaciones de fallas en un sistema de potencia modelado con el software EMTDC/PSCAD. Y se genera un archivo de datos de salida con las condiciones de prefalla, falla y posfalla del sistema.
2. Leer el archivo de datos de salida de voltajes y corrientes, adecuarlas para simular la relación de transformación de TC's y TP's ideales, en un programa escrito en MATLAB.
3. Se introducen las muestras de seis ciclos de las señales discretizadas de voltajes y corrientes (dos de prefalla y cuatro de posfalla), a la memoria del DSP por medio de un programa para descargarlo al microcontrolador.
4. Se programa al microcontrolador para tomar cada una de estas muestras simulando así, la operación del ADC.

La ejecución de estos pasos no genera problemas, porque la programación de captura muestras por el ADC del microcontrolador, se realiza mediante interrupciones producidas por uno de los temporizadores que el microcontrolador posee. Por lo tanto, en lugar de tomar la muestra del ADC, se obliga al "micro" a tomar la muestra que se encuentra en su memoria, las cuales corresponden a la falla simulada en EMTDC/PSCAD, sin que esto influya en el tiempo de muestreo. De esta manera se logra que el micro procese las muestras de voltaje y corriente a través de

los coeficientes de los filtros digitales para lograr que forme los fasores y posteriormente, se calcule la impedancia de acuerdo con las muestras que se tienen de la falla simulada. La programación del microcontrolador para tomar las muestras, se explica de manera detallada en la sección 3.5.2.

### **3.5.1. Adecuación de las señales de voltajes y corrientes para ser leídas por el ADC.**

Para que el microcontrolador pueda tomar las muestras de voltajes y corrientes sin que le provoquen daño, es necesario realizar una adecuación de dichas señales, es decir, reducir la magnitud de las señales que alimentan las terminales del ADC. Primeramente, se reducen estas cantidades por medio de los TC's y TP's, que son los encargados de tomar las señales directamente del sistema de potencia, como estas variables aún son de magnitud elevada para conectarlas a las terminales de ADC, es necesario implementar un circuito electrónico que permita, aún más, la reducción de la amplitud de la onda eléctrica.

#### **3.5.1.1 Manejo de las señales de voltaje.**

Regularmente los TP's entregan en su salida una señal senoidal de voltaje de 360V. de pico a pico (180V. pico y 127V. rms), pero el microcontrolador solo puede aceptar voltajes por debajo de 3V. de pico a pico, pero sin cambio de signo, por ejemplo; para una señal senoidal de 60 Hz. con amplitud de 1.5 V. pico, es necesario un nivel de offset de CD de 1.5 Volts, es decir, solo acepta señales que vayan de 0 a 3 Volts, esto es sin cambios de polaridad. (Ver la Figura 3.7). Por esta razón es necesaria la implementación de un circuito eléctrico para atenuar las señales dentro de los niveles y con las características adecuadas para medir correctamente y no dañar el funcionamiento del microcontrolador.

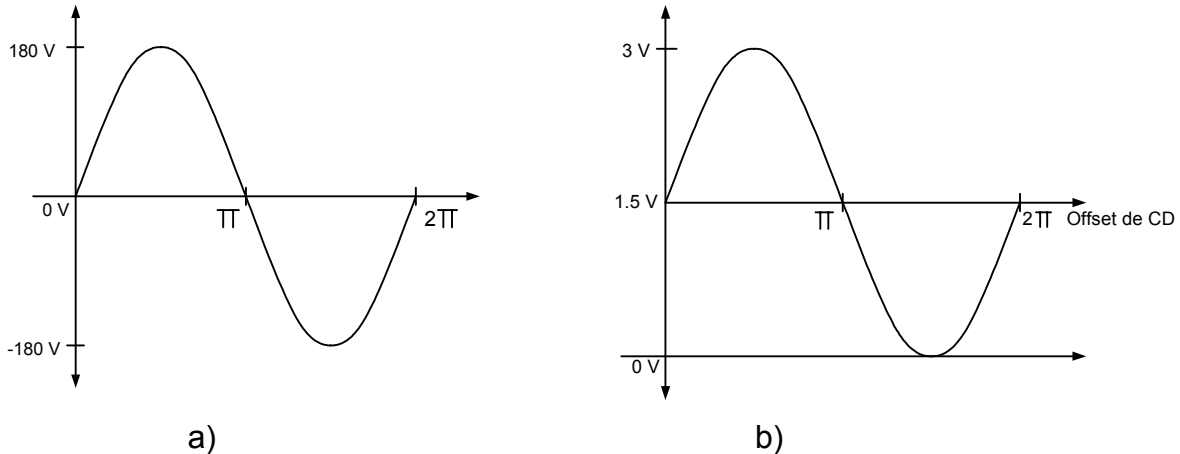


Figura 3.7 a) onda senoidal de voltaje de 360 V. pico a pico. b) onda senoidal de voltaje de 3 V. pico a pico, con un offset de 1.5 V. de CD.

El circuito eléctrico usado para adecuar la señal requiere una fuente de CD para subir la señal (offset), ésta, está disponible en la tarjeta electrónica de desarrollo del microcontrolador. La señal de offset puede utilizarse para ambos casos, para señales de voltaje y señales de corriente.

Para realizar dicha adecuación se usa un transformador de voltaje con relación de 120/6 para reducir el nivel de voltaje de 120 Volts a 6 V. y un arreglo de resistencias eléctricas para subir la señal con un offset de CD de 1.5 V. y así tener la configuración que nos proporcione el voltaje requerido de 0 a 3 V. El diagrama del circuito se muestra en la Figura 3.8.

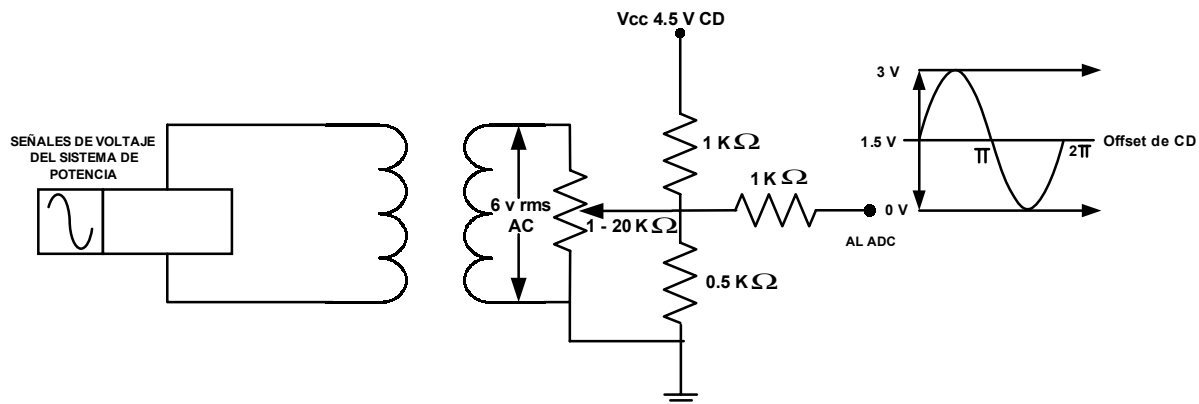


Figura 3.8 Circuito para adecuación de señales de voltaje del sistema de potencia.

### **3.5.1.2 Manejo de las señales de corriente.**

Para la captura de las señales de corriente, se realiza algo similar al del voltaje. Debido a que el ADC del microcontrolador no puede tomar muestras de señales de corriente. Por otro lado, en condiciones normales de operación del sistema, el TC principal en su salida entrega una corriente entre 0 a 5 amperes, entonces se utiliza un TC auxiliar y un arreglo de resistencias para convertir las señales de corriente a señales de voltaje, nuevamente, éste debe estar entre 0 a 3 V. pico a pico, como se mostró en la Figura 3.7. b).

La corriente secundaria de los TC's principales está estandarizada a 1 o 5 Amp., por ejemplo, un TC que está conectado al sistema de potencia y que entrega en su devanado secundario una corriente de 5 amperes, en el caso más severo, la corriente de corto circuito puede incrementarse hasta 30 veces su valor de corriente secundaria nominal [1,3,4]. Esto implica que en el secundario del TC durante la falla, pueden presentarse corrientes de hasta 150 amperes. Ahora bien, para convertir este nivel de corriente a un nivel de voltaje que el microcontrolador pueda muestrear, es necesario utilizar otro transformador de corriente. Para este caso, se toma un transformador auxiliar que tenga una relación de transformación de 0.1 Volt en secundario, por cada 1 ampere en el primario (a veces depende de la resistencia de carga), así cuando una falla presente valores de 150 amperes en el secundario del TC principal, entonces en el secundario del TC auxiliar se tendrá un voltaje máximo de 15 Volts, y con el arreglo serie-paralelo de resistencias se logra reducir este voltaje a un máximo valor pico a pico, de 3 Volts, esto para tener un buen desempeño del microcontrolador al realizar el muestreo de estas señales.

El circuito para convertir estas señales de corriente a señales de voltaje en forma equivalente se presenta en la Figura 3.9. y aplicando análisis de circuitos se puede realizar lo siguiente:

Analizando el circuito de resistencias en serie compuesto por R3 y R4 se puede calcular el valor del "voltaje offset" necesario de 1.5 VCD al evaluar la ecuación 3.35.

$$V_{offset} = \frac{R4}{R3 + R4} (V_T) = \frac{500}{1500} (4.5) = 1.5 \text{ Volts} \quad (3.35)$$

donde:

$V_{offset}$  = Es el valor de voltaje offset de CD

$V_T$  = Es el voltaje total Vcc.

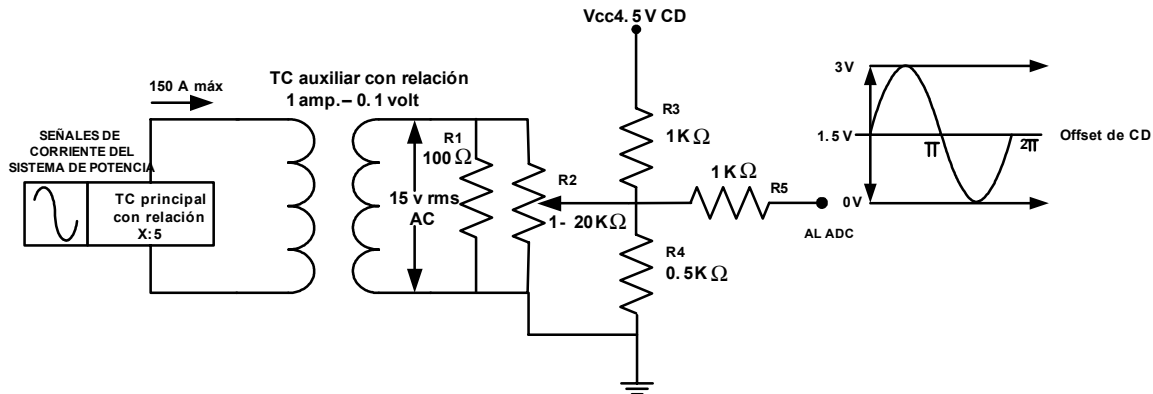


Figura 3.9 Circuito para adecuación de señales de corriente del sistema de potencia.

Después de implementar este circuito físicamente, se conecta una fuente de 4.5 Volts de CD. Como en la Figura 3.9, y se obtiene el voltaje o caída de tensión en la resistencia R4 de 1.5 Volts, después, se debe aplicar una tensión senoidal de 15 Volts rms en R1 para simular la operación del TC auxiliar, entonces se regula la resistencia R2 hasta obtener un voltaje de pico a pico de 3 VCA. Por estar conectado con el voltaje de 1.5 VCD, se realiza una suma de voltajes, teniendo una forma de onda de salida como se muestra en la Figura 3.9. Es decir, una forma de onda tipo senoidal, pero que oscila entre 0 y 3 Volts.

La función de la resistencia R5 de la Figura 3.9. es únicamente de protección para el microcontrolador, y no presenta ningún efecto que perturbe o provoque una incorrecta medición, esto debido a que el ADC presenta una alta resistencia de entrada del orden de Megaohms, por lo que la corriente que fluye a través de éste circuito, se puede despreciar.

Como se mencionó anteriormente, en el caso de fallas reales en un sistema de potencia, se pueden generar corrientes en el secundario del TC principal hasta 150 amperes, pero en este proyecto fue prácticamente imposible generar una corriente de esta magnitud, por lo que fue necesario reducir 10 veces este valor, esto es corrientes máximas de 15 amperes. Sin embargo, por software se puede modificar esta cantidad, para que el algoritmo funcione con valores cercanos a las corrientes de falla reales.

De esta forma se puede medir la corriente, ya que no existe en la actualidad, ningún dispositivo ADC que pueda convertir directamente una señal de corriente a un valor digital, por lo que es necesario convertir la corriente a un voltaje equivalente. El éxito de este arreglo o circuito eléctrico depende directamente de la relación y capacidad del TC secundario.

### ***3.5.2. Muestreo de señales por el microcontrolador***

El algoritmo desarrollado requiere de seis señales de entrada a la frecuencia fundamental, tres de voltaje y tres de corriente. Una vez que estas señales son atenuadas por los TC's, TP's principales, además de los acondicionadores de señales analógicas, como se mostró en el tema anterior, entonces se puede realizar el muestreo por el convertidor analógico digital (ADC) del microcontrolador, a una frecuencia tal que proporcione una ventana de 24 muestras por cada ciclo de frecuencia fundamental. Se eligió esta ventana de 24 muestras porque no presenta ningún problema en la simulación del filtro digital en MATLAB, además, esta frecuencia de muestreo representa el doble del reportado en [40], entonces se asegura que la respuesta sea igual o mejor en cuanto a la estimación de fasores.

#### ***3.5.2.1. Determinación de la frecuencia de muestreo.***

Como se mencionó anteriormente, el número de muestras que el algoritmo necesita para la formación de los fasores de voltaje y corriente es de 24 por cada ciclo de

frecuencia fundamental. Entonces, es necesario calcular la frecuencia de muestreo que será programada en el temporizador del microcontrolador que se encarga enviar la interrupción al ADC para la captura de cada muestra, dicho cálculo se presenta en la ecuación 3.36.

$$f_m = \frac{1}{T/24} \quad (3.36)$$

donde:

$f_m$  = es la frecuencia de muestreo.

$T$  = es el período de la señal a muestrear (la fundamental). Dada por la ecuación 3.37.

$$T = \frac{1}{60} \quad (3.37)$$

En la ecuación 3.38 se presenta el resultado de sustituir a  $T$  en la ecuación 3.36.

$$f_m = \frac{1}{\frac{1/60}{24}} = 1440 \quad (3.38)$$

De lo anterior se tiene que la frecuencia de muestreo para una ventana de 24 muestras por ciclo de frecuencia fundamental es de 1.44 Khz., como se mostró en la Figura 3.6.

### **3.5.2.2. Programación de la frecuencia de muestreo para el ADC.**

Para que el ADC del microcontrolador pueda tomar las muestras a la frecuencia calculada de 1.44 Khz., es necesario programar algunos elementos de control como es el preescalador, el propio ADC, con sus respectivos registros de control, además de los registros del temporizador interno (el contador, su valor de inicialización y su comparador), que se encarga de enviar la señal de sincronía para que el ADC capture una muestra cada vez que reciba esta señal.

Primeramente, se realizan algunos cálculos, relacionados a la operación y funcionamiento del mismo microcontrolador; como la frecuencia de reloj de operación y el empleo del escalador o preescalador de frecuencia, utilizado en el control de los distintos temporizadores.

El reloj del microcontrolador opera a 80 Mhz., pero dadas las características de operación de éste, se considera una frecuencia de trabajo del chip de 40 Mhz., entonces se pueden dar las siguientes conjeturas.

- El IP BUS del microcontrolador opera a 40 Mhz.
- El preescalador divide esta frecuencia en 2, 4, 8 y 16, para este caso se eligió el valor de 16.

Con estas características se puede calcular el periodo de trabajo del temporizador, como se muestra en la ecuación 3.39.

$$F_{temp} = \frac{1}{\frac{FIP}{pre}} \quad (3.39)$$

Donde:

- $F_{temp}$  = es el periodo del temporizador.  
 $FIP$  = es la frecuencia de operación del IP BUS.  
 $Pre$  = es número del preescalador.

Sustituyendo los valores conocidos en la ecuación 3.39, se tiene en la ecuación 3.40, el periodo de cada pulso del temporizador.

$$F_{temp} = \frac{1}{\frac{40000000}{16}} = \frac{1}{2500000} = 400\eta \text{ segundos} \quad (3.40)$$

Como cada pulso se genera a 400  $\eta$  segundos y como es necesario tener un número de pulsos que proporcione una frecuencia de muestreo de 1.44 Khz., entonces debe



programarse al comparador del temporizador con el valor adecuado, esto puede hacerse al evaluar la ecuación 3.41.

$$F_{muestra} = \frac{1}{2 * inter} \quad (3.41)$$

Donde:

$F_{muestra}$  = Frecuencia de muestreo deseada.  
 $inter$  = Intervalo de tiempo entre cada muestra.

Despejando la variable  $inter$  se obtiene la ecuación 3.42.

$$inter = \frac{1}{2 * F_{muestra}} = \frac{1}{2 * 1440} = 0.00034722 \text{ segundos} \quad (3.42)$$

Después de sustituir la frecuencia de muestreo de 1.44 Khz en la ecuación 3.42, Se obtiene como resultado un intervalo de tiempo entre cada muestra de **347.22 microsegundos**. De esto se deduce que el número que debe programarse al comparador ( $comp$ ) es de 868, como puede observarse en la ecuación 3.43.

$$comp = \frac{inter}{F_{temp}} = \frac{0.00034722}{0.000000400} = 868.05 \quad (3.43)$$

Un ejemplo de la programación para este caso se puede observar en los siguientes registros de operación para el temporizador llamado C2 [39].

```
*TmrC2_CMP1 = 868;  
*TmrC2_LOAD = 0x0000;  
*TmrC2_CNTR = 0x0000;
```

donde:

\*TmrC2\_CMP1 = es el registro del comparador.  
\*TmrC2\_LOAD = es el registro de inicialización del contador.  
\*TmrC2\_CNTR = es el registro del contador con el valor al reiniciar el conteo.

El funcionamiento del temporizador es el siguiente: primeramente el contador toma el valor del registro de inicialización. Después, se incrementa al valor que tenga el registro del contador y finalmente, se inicia el conteo hasta el valor deseado que se encuentra preestablecido en el registro del comparador. Una vez alcanzado este valor, el contador se inicializa con los mismos valores que al inicio del ciclo.

La configuración del ADC es más sencilla, puesto que solo se programan dos registros. El primero es para dar al ADC la indicación de hacer la conversión (toma de muestra) cuando reciba la señal de sincronización proveniente del temporizador, también para que genere una interrupción cada vez que se tenga el valor digital de la conversión y para elegir el modo de la toma de muestras de cada uno de los canales habilitados. Y el segundo es para habilitar los canales deseados.

Un ejemplo de programación de estos registros es el siguiente:

```
*ADCA_ADCR1 = 0x5805;
```

Este valor se encuentra en forma hexadecimal y se puede pasar a código binario de 16 bits quedando de la siguiente manera.

```
*ADCA_ADCR1 = 0101100000000101
```

Enumerando de izquierda a derecha cada uno de los bits (del 0 al 15) se explica la operación de cada uno de ellos.

- [15] = 0 no usado
- [14] = 1 ADC detenido.
- [13] = 0 inicio, este bit indica el inicio de la conversión.
- [12] = 1 sincronía, la conversión inicia con sincronía del temporizador.
- [11] = 1 *eosie*, al final de la captura se genera una interrupción.
- [10] = 0 *zcie*, la interrupción de cruce por cero está deshabilitada.
- [9] = 0 LLMTIE, interrupción de límite bajo está desactivada.
- [8] = 0 HLMTIE, interrupción de límite alto está desactivada.
- [7:4]= 000 CHNCHG, seleccionar la entrada finalizada.
- [3] = 0 este bit no se utiliza.
- [2:0]= 101 SMODE, captura simultánea de las entradas.



Ahora, con el siguiente registro se seleccionan las entradas que serán necesarias para la captura de señales.

```
*ADCA_ADSDIS = 0x0088;
```

Se presentan en código binario y se enumeran de la misma forma que el ejemplo anterior y se explica la operación de cada uno.

```
0000000010001000
```

```
[15:14] = 00 TEST, modo normal de operación.
[13:8]   = 000000 reservados.
[7:0]   = 10001000 entradas deshabilitadas 7,3
```

Para habilitar las entradas se coloca 0 en el bit correspondiente y para desactivarlas se coloca 1. por lo que con esta configuración se tienen las entradas 0, 1, 2, 4, 5, 6 activadas y 7, 3 desactivadas.

En este caso en particular se seleccionan 6 entradas, las cuales capturan las señales de corriente y voltaje trifásicos de manera simultánea par por par, es decir, que se capturan 2 canales al mismo tiempo, por ciclo del ADC, la primer captura es para las señales de voltaje y corriente de la fase A (por decir una fase), la segunda es para las correspondientes de la fase B y por último, la tercera para las que corresponden a la fase C. Las parejas se presentan en la Tabla 3.2.

Tabla 3.2. Elección de las entradas del ADC.

Fases	Entrada	Tipo de señal
Pareja de entradas para la fase A	0	Señal de voltaje
	4	Señal de corriente
Pareja de entradas para la fase B	1	Señal de voltaje
	5	Señal de corriente
Pareja de entradas para la fase C	2	Señal de voltaje
	6	Señal de corriente

Esta selección de captura instantánea por parejas de voltaje y corriente de una misma fase, se realiza para evitar un desfasamiento de tiempo en la toma de muestras que provoque un resultado con mayor error en el cálculo de la impedancia medida por el DSP. Se recomienda ver la Figura 3.6 para mayor referencia en la configuración física del ADC.

### **3.5.2.3. La conversión analógico-digital.**

Se llama conversión analógico – digital a la acción de tomar el valor de una señal analógica y pasarla a un valor equivalente en forma digital, por ejemplo; una magnitud de 3 Volts es capturada por el ADC y en su salida se tiene una cantidad digital de 30000. Este valor representa en forma equivalente al voltaje analógico para que pueda ser usado en el programa digital del microcontrolador.

El convertidor analógico digital con que cuenta el microcontrolador DSP56807 tiene varias características, a continuación se mencionan las más importantes.

- Resolución de 12 bits.
- La frecuencia máxima de reloj del ADC es 5 Mhz., con periodo de 200ns.
- 8 conversiones en 26.5 ciclos de reloj del ADC ( $26.5 \times 200 \text{ ns} = 5.3\mu\text{s}$ ) usando el modo simultáneo.
- Multiplexor interno que puede seleccionar 2 de las 8 entradas al mismo tiempo.
- Muestreo secuencial o simultáneo.
- 2 circuitos para muestreo y retención (sample and hold) que permiten capturar 2 muestras de canales diferentes al mismo tiempo como se puede ver en la Figura 3.10.

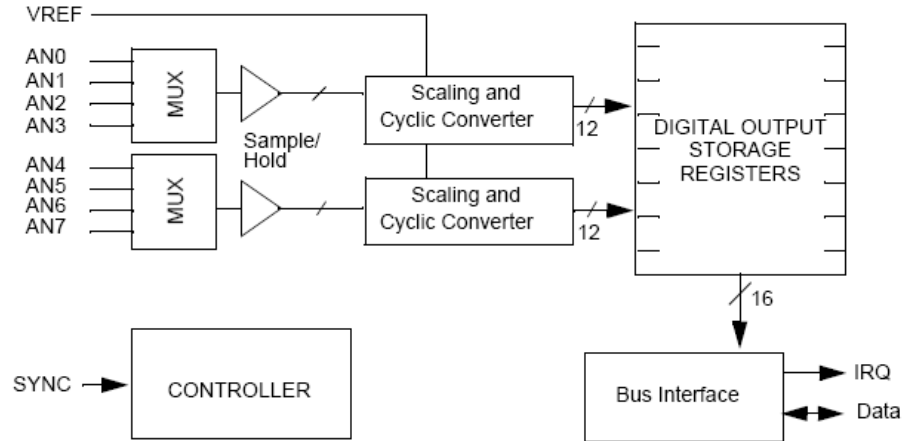


Figura 3.10. Diagrama esquemático del circuito ADC.

Con estas características se puede deducir que la frecuencia de muestreo máxima para tomar las 6 señales que el algoritmo requiere para la formación de la impedancia medida, es de 188.679 KHz [38].

Esta frecuencia es relativa, es decir, que también depende en gran medida de las operaciones que se realicen con las muestras dentro del programa. Esto se puede explicar con la aplicación que se realiza en este trabajo en donde se requiere tomar 6 muestras de voltajes y corrientes trifásicas, aplicar el filtro digital de mínimos errores cuadrados que involucra una serie de sumas, multiplicaciones, aplicación de funciones trigonométricas, etc., después formar la impedancia de secuencia positiva y, por último, determinar si esta impedancia está dentro de la zona de operación. Esto hace que el tiempo transcurrido después de realizar todas estas operaciones, sea aproximadamente 16 veces mayor al intervalo de tiempo entre muestras que fue calculado en  $347.22 \mu$  segundos (ecuación 3.42). Este párrafo se puede explicar de manera más detallada al observar la Figura 3.11., donde se muestra el diagrama de tiempos desde la toma de la primer, hasta la última muestra, de un ciclo completo de frecuencia fundamental.

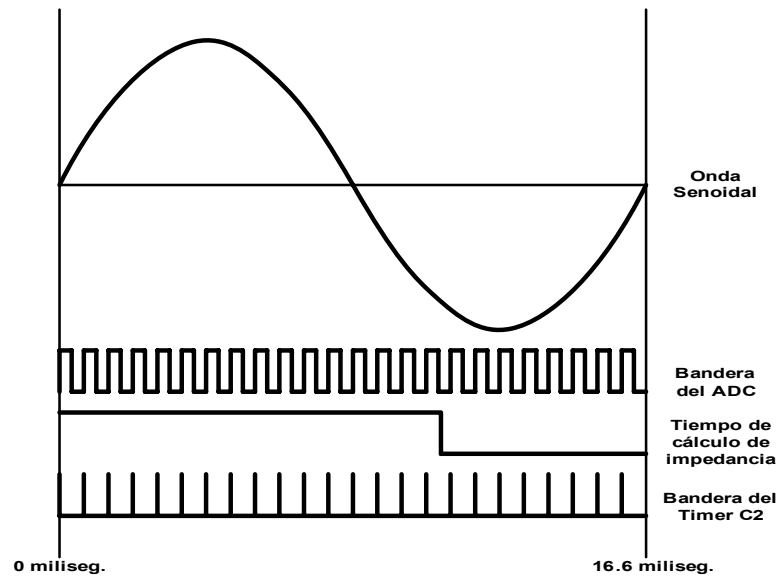


Figura 3.11. Diagrama de tiempos.

#### 3.5.2.4. Ejemplo de la toma de muestras por el ADC

Para ver la operación de la toma de muestras por el ADC, se puede observar la Figura 3.12, donde se presentan las muestras que son adquiridas en tiempo real por el convertidor analógico digital, de las señales de voltaje de dos fases (A y B) y enviadas a la PC para verlas gráficamente en LabVIEW.

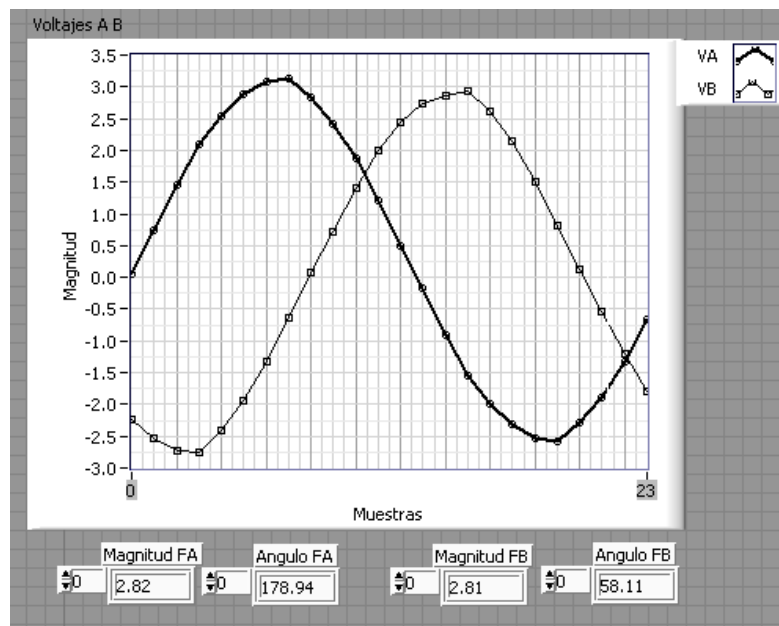


Figura 3.12. Muestras en tiempo real de señales de voltaje de las fases A y B.

La magnitud y ángulo de estas señales se presentan en la Tabla 3.3. donde se observa la diferencia del ángulo entre fases de 120.8 grados, con la magnitud del mismo valor. Estos resultados son los arrojados por el DSP y los obtenidos con el programa en LabVIEW, y dado que la diferencia es mínima, se demuestra que son correctos.

Tabla 3.3. Fasores de voltajes de las fases A y B.

Fase	Magnitud en el DSP	Magnitud en LabVIEW	Angulo en el DSP	Angulo en LabVIEW
A	2.81	2.82	179.2	178.94
B	2.80	2.81	58.4	58.11
Diferencia	0.01	0.01	120.8	120.83

A continuación, se presenta otro ejemplo de la operación del algoritmo, pero ahora con señales de voltaje y corriente provenientes de la fuente de DOUBLE, para simular las señales del sistema de potencia, pero tomando en cuenta una sola fase.

En la Figura 3.13. se puede observar las muestras tomadas por el DSP, enviadas a la PC, donde se calcula la impedancia de falla, en este caso la corriente es mínima, simulando la operación normal del sistema eléctrico, por lo que no hay falla.

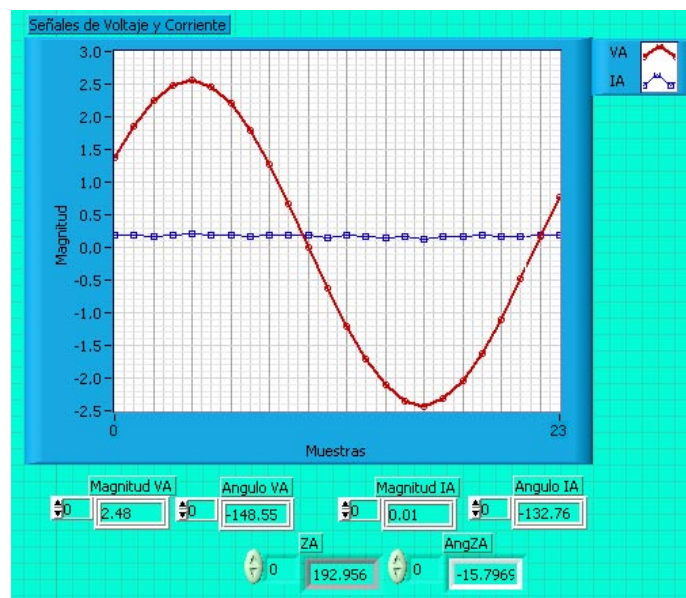


Figura 3.13. Muestras en tiempo real de señales de voltaje y corriente, sin falla.

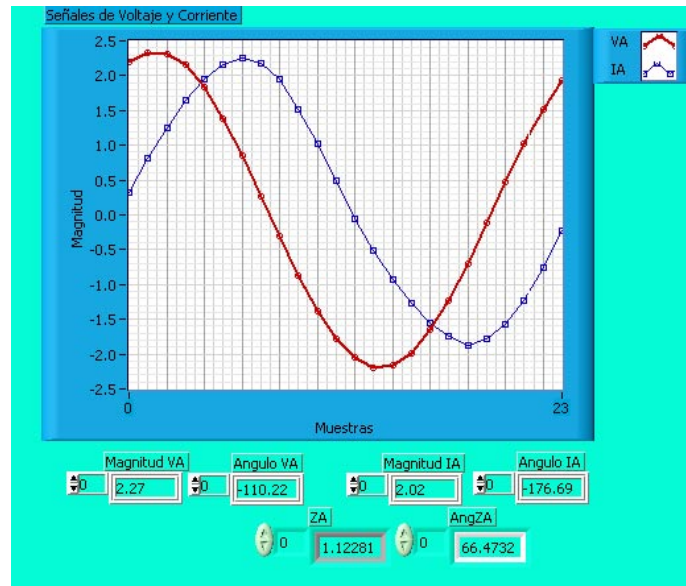


Figura 3.14. Muestras en tiempo real de señales de voltaje y corriente, con falla.

En la Figura 3.14. se observa que la corriente se incrementó considerablemente, por lo que se aprecia una disminución en la impedancia de falla, y de acuerdo con la característica de operación del algoritmo, detecta una falla y dispara la protección.

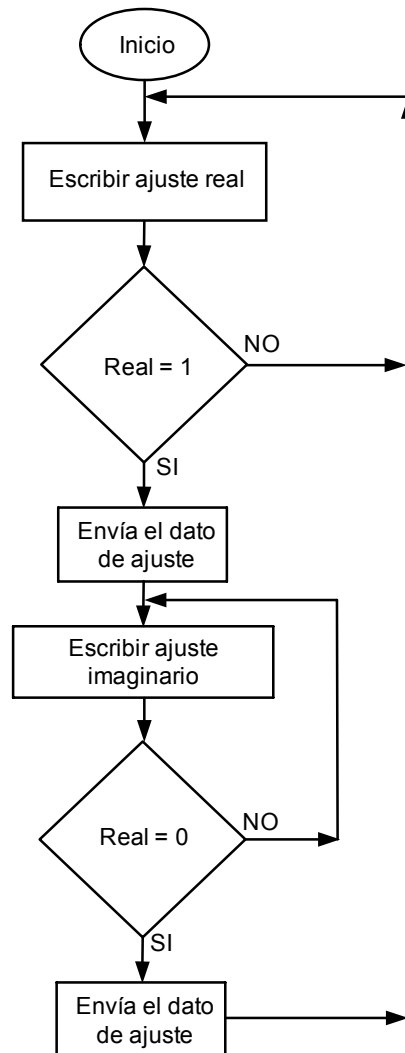
### 3.6. Actualización de los ajustes de impedancia en el DSP.

La acción de modificar los ajustes en el DSP, se realiza a través de un programa escrito en LabVIEW, encargado de interactuar con el DSP y la PC, ver apéndice D.

#### 3.6.1. Diagrama de flujo del programa en LabVIEW.

En la Figura 3.15 se observa el diagrama de flujo correspondiente al envío de datos para modificar la impedancia de ajuste. Para enviar los datos es necesario poner el menú “escritura/lectura” en escritura. El primer paso es escribir la parte real de la impedancia de ajuste dada en valores de 0 a 10 con incrementos de 0.1, en la variable “ajuste”, después, se presiona el botón de control con etiqueta “hacer” para enviar el dato (ver Figura 3.17). El segundo paso es escribir la parte imaginaria de la impedancia de ajuste dado en valores de 0 a 100 con incrementos de 1, también en la variable “ajuste”, y activar el mismo botón anterior. El cambio automático que se realiza en el nombre en la etiqueta del botón “real/imag” es para identificar si es la parte real o imaginaria del ajuste que se envía al DSP.





*Figura 3.15 Diagrama del envío de datos para el ajuste del DSP*

El diagrama de flujo para recibir las muestras provenientes del microcontrolador se presenta en la Figura 3.16. donde se puede observar que es necesario activar la variable “Hacer” para recibir las muestras. Esta variable está dada por un control en el panel frontal del programa en LabVIEW, una vez activado, se envía un caracter identificado en ASCII como “enter” (este valor es 10). El microcontrolador identifica este caracter como la señal para descargar la información del buffer que tiene las muestras. Una vez leídas, se calcula la impedancia y se grafican los resultados, como se verá en la Figura 3.18.

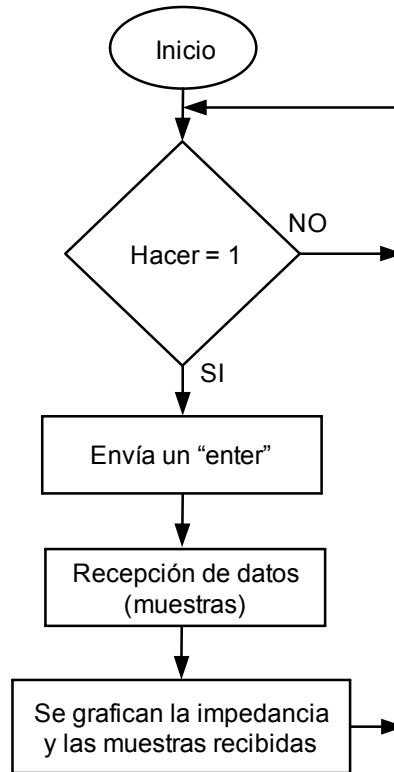


Figura 3.16. Diagrama para recepción de muestras del DSP.

### 3.6.2. Presentación del programa en LabVIEW (panel frontal).

Este programa fue desarrollado para obtener del DSP las mediciones de voltaje y corriente del sistema eléctrico de potencia en el momento en que sean requeridas, además de realizar las modificaciones en los ajustes para la segunda zona, cuando sea necesario. La pantalla o panel frontal de interacción con el DSP se presenta en la Figura 3.17. donde se tienen tres botones de control. El botón “STOP” sirve para detener al programa de manera total, el botón “real” tiene dos posiciones, una es real y la otra es imaginario, es para el envío de la parte real (primera acción) y para la parte imaginaria (segunda acción) y el botón “hacer” es usado para indicar al DSP que se requiere un envío de datos (DSP – PC). En el arreglo “ajuste” se coloca el valor de ajuste deseado, y los de parte real y parte imaginaria son para ver el ajuste actual que se envió al DSP. Los otros indicadores son para ver la cadena de datos enviados por el DSP. En la Figura 3.18 se presentan las gráficas de las muestras del ADC.

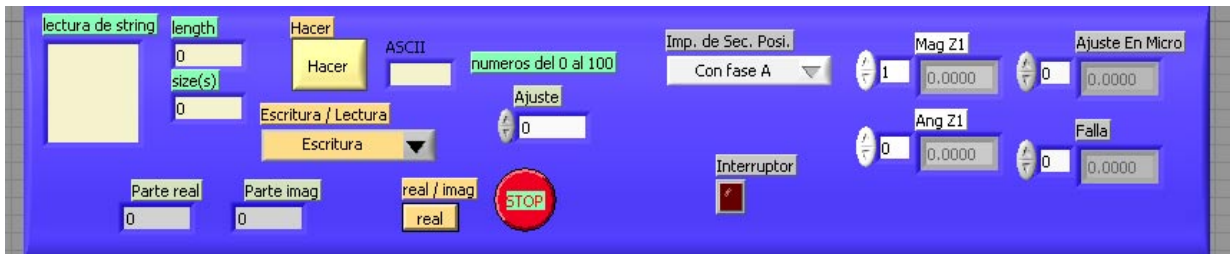


Figura 3.17. Programa en LabVIEW para interactuar con el DSP.

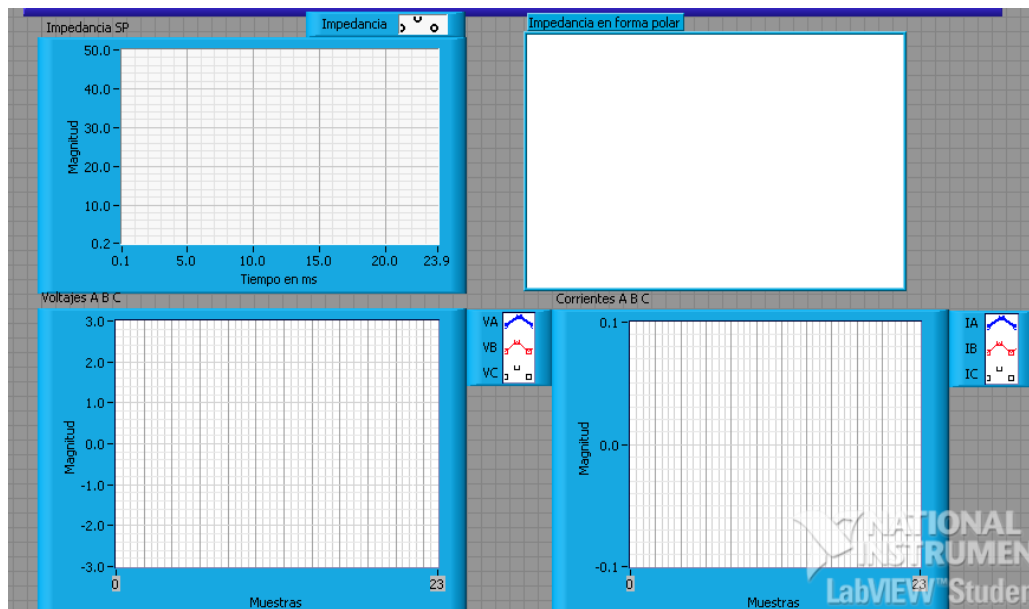


Figura 3.18. Pantalla en LabVIEW de indicadores de las señales muestreadas.

Un ejemplo de la decisión para realizar los ajustes en la segunda zona se estudian en el capítulo cuatro. Estos ajustes se basan en las técnicas de protección adaptable, las cuales son tomadas de [5,35,43], únicamente para apreciar los alcances que se pueden obtener por el uso de esquemas de protección adaptable.

## CAPÍTULO IV

### **4. COMPARACIÓN DE RELEVADORES DE DISTANCIA: CON AJUSTES CONVENCIONALES Y EN ESQUEMAS DE PROTECCIÓN ADAPTABLE.**

#### **4.1. Introducción.**

La práctica común de los ingenieros encargados de las tareas de protección de SEP's, es proponer ajustes de los relevadores que respondan a las condiciones críticas de operación, sin embargo en esquemas de protección adaptable estos ajustes se proponen para las condiciones de operación actual del sistema. En este capítulo se estudian distintas condiciones de operación del SEP para evaluar la operación o no operación del relevador ante distintas fallas, ante cambios de configuración del sistema y en condiciones de carga máxima o carga mínima en algún momento determinado, para observar la conveniencia de proteger un sistema de transmisión con esquemas de protección adaptable.

#### **4.2. Cálculo de ajustes con esquemas de protección convencionales.**

Los ajustes aplicados a esquemas de protección convencionales son aquellos calculados con los métodos en los que se toman en cuenta varios escenarios de operación del sistema de potencia en general, y se considera la condición más crítica para el sistema de protecciones.

##### **4.2.1. Ajuste de la zona 1**

El ajuste de esta zona debe proteger a la línea de transmisión entre un 80 % a 90 % de la impedancia de secuencia positiva de la línea [5,35,43], esta impedancia es proporcional a la longitud de la línea (ver Figura 4.1). El ajuste para esta zona se puede representar con la ecuación (4.1).

$$Z_{1r1} = K * Z_{1L} \quad (4.1)$$

donde:

$Z1_{r1}$  = es el ajuste en Zona 1 para el relevador  $r1$ .

$Z_{1L}$  = es la impedancia de secuencia positiva de la línea.

$K$  = es el factor de seguridad y se establece entre 0.8 y 0.9.

Así con esta ecuación se obtiene el alcance de la zona 1, por ejemplo, en la Figura 4.1, se muestra una línea de transmisión  $Z_{1L}$  con una longitud de 100 km., y considérese el factor  $K$  igual a 0.85 entonces, se tiene:

Longitud de  $Z_{1L}$  = 100 km.

Impedancia de secuencia positiva de  $Z_{1L}$  =  $37.9 \angle 87.5$

Factor  $K$  utilizado = 0.85

Entonces, al evaluar la ecuación (4.1) resulta:

$$Z1_{r1} = 0.85 * 37.9 \angle 87.5 = 32.21 \angle 87.5$$

Y expresada en forma rectangular  $Z1_{r1} = 1.405 + j 32.179$

Este resultado es del diámetro del círculo que representa al área de operación en zona 1 del relevador, y para la identificación de fallas se usa el radio del círculo, entonces el ajuste dado al DSP, que debe ser el radio del origen al centro del círculo, es:

$$\text{Ajuste en zona 1} = \frac{1}{2} Z1_{r1} = 0.7025 + j16.0895$$

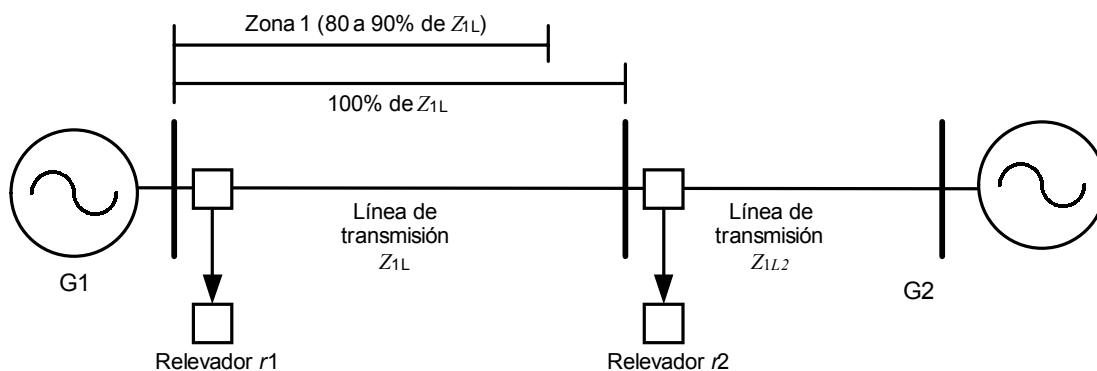


Figura 4.1 Ajuste en zona 1 para el relevador  $r1$ .

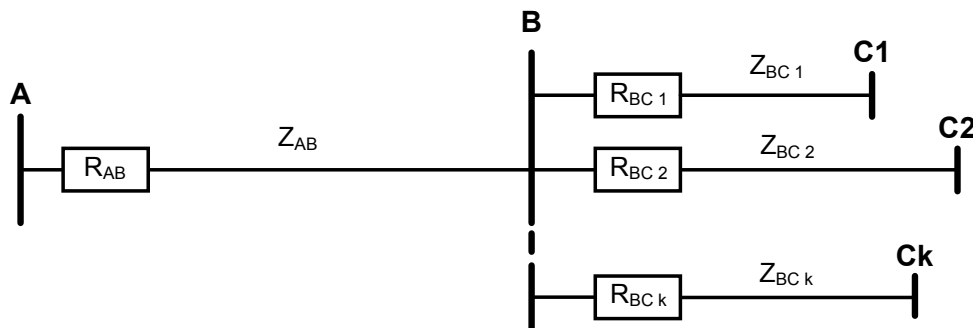
### 4.2.2. Ajuste de la zona 2.

Una metodología comúnmente aplicada para calcular la impedancia de ajuste de la zona 2, se puede resumir de la manera siguiente.

1. Se simulan fallas en el extremo de la línea (line-end fault) en cada una de las líneas remotas del relevador de interés.
2. Se calcula el factor de *infeed* correspondiente a cada falla simulada.
3. Se identifica la línea remota para el cual se obtuvo el menor factor de *infeed*.
4. Se simulan estas fallas solamente en la línea identificada en el punto anterior, pero ahora considerando contingencias en los elementos conectados a la barra intermedia entre el relevador y la falla.
5. Se calcula el factor de *infeed* correspondiente a cada contingencia.
6. Se selecciona el menor factor de *infeed*.
7. Se calcula la impedancia de ajuste de la zona 2 para cubrir hasta un 50% de la línea identificada en el punto 3, y se utiliza el factor de *infeed* seleccionado en el punto 6.

Otra metodología convencional dice que este ajuste de la zona 2 debe estar entre el 40% y 50% de la línea adyacente siguiente que tenga la menor impedancia de secuencia positiva [2].

Un ejemplo se presenta en el sistema de la *Figura 4.2*.



*Figura 4.2 Sistema de subtransmisión radial protegido por relevadores de distancia*

La impedancia de ajuste de la zona 2 para el relevador que protege la línea AB de la Figura 4.2 es:

$$Z2(R_{AB}) = Z_{AB} + 0.5 * \text{Minimo}\{Z_{BCi}\} \quad (4.2)$$

Donde:

- $Z2(R_{AB})$  = es la impedancia de ajuste para segunda zona del relevador  $R_{AB}$   
 $Z_{AB}$  = es la impedancia de secuencia positiva de la línea protegida A-B.  
 $Z_{BCi}$  = es la impedancia de secuencia positiva de la línea remota más corta.  
 $i$  = es igual a 1,2,...,k; donde k es el número total de relevadores primarios.

Este criterio es usado para definir el ajuste en zona 2 del relevador  $R_{26}$  mostrado en la Figura 4.3. La impedancia de secuencia positiva de las líneas, está dada en valores por unidad. La línea con el menor valor de impedancia en el nodo remoto es la línea entre el bus 12 y el bus 11. Sustituyendo estos valores en la ecuación (4.2) se tienen los siguientes ajustes para la zona 2 del relevador  $R_{26}$ .

$$Z2(R_{26}) = 0.03299 + j0.13586 + 0.50 * (0.00446 + j0.01803)$$

$$Z2(R_{26}) = 0.14909 \angle 76.33 \text{ p.u.}$$

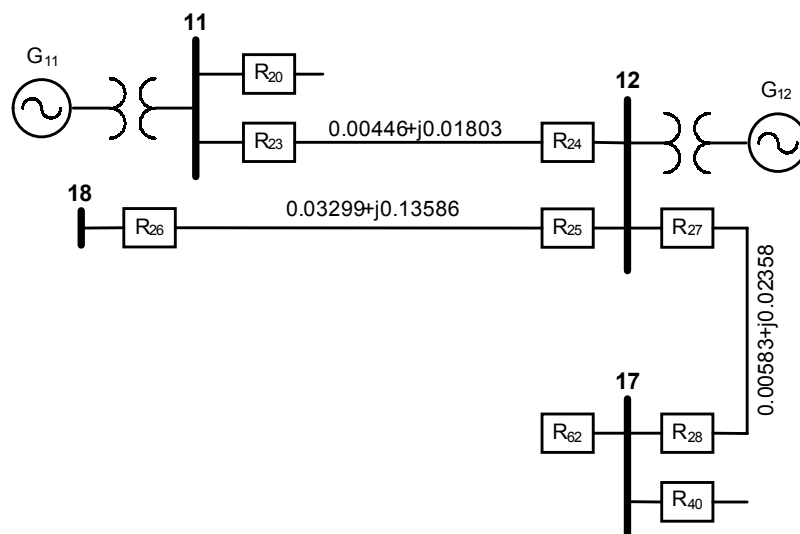


Figura 4.3 Porción de un sistema de potencia usado como ejemplo.

### **4.3. Cálculo de ajustes con esquemas de protección adaptable.**

El cálculo con ajustes con esquemas de protección adaptable se refiere a que dichos ajustes se realizan cuando las condiciones o configuración del sistema eléctrico de potencia ha cambiando. Los ajustes de protección adaptable únicamente son realizados para las zonas 2 y 3 de cada relevador, puesto que es en estas zonas en donde se presentan los problemas de sobrealcance y subalcance. En este cálculo se incluye un nuevo método propuesto para ajustar el alcance de la zona 2 para los relevadores de distancia [35].

En este trabajo solamente se implementa en el microcontrolador, el ajuste adaptable en zona 1 y zona 2.

#### **4.3.1. Ajuste adaptable para zona 1.**

El método para calcular el ajuste adaptable en zona 1 no presenta cambio alguno comparado con el método convencional, debido a que este ajuste siempre estará entre el 80% al 90% de la impedancia de secuencia positiva total de la línea de transmisión. Por esta razón, se recomienda revisar el tema 4.2.1.

#### **4.3.2. Ajuste adaptable para zona 2.**

Esta técnica propuesta calcula la impedancia vista por el relevador en zona 2 para las condiciones existentes en el sistema de potencia. Esto significa que el ajuste de la zona 2 del relevador de distancia en un sistema de protección adaptable, se determina por el cálculo de la impedancia aparente vista por el relevador en zona 2 cuando las fallas son simuladas en las líneas del nodo remoto, considerando las condiciones de operación prevalecientes del sistema de potencia.



El método adaptable para el ajuste de la zona 2 de los relevadores hace referencia al sistema de ejemplo de la Figura 4.4, y se describe en los siguientes pasos para determinar el ajuste de la zona 2 del relevador  $R_{BK}$ .

- 1) Obtener la condición de operación actual del sistema de potencia (topología de la red, flujos de potencia, y generación). Simular una falla trifásica ( $F_i$ ) al final de la zona 1 de la línea B-C<sub>i</sub> con el interruptor remoto CB<sub>i</sub> abierto.
- 2) Determinar el ajuste de la zona 2 del relevador RBK usando la impedancia aparente vista por el relevador  $R_{BK}$  para la falla  $F_i$  evaluando la ecuación 4.3.

$$Z2(R_{BK}, R_{Pri}) = Z_{AB} + K2 * [ZA(R_{BK}, F_i) - Z_{AB}] \quad (4.3)$$

Donde:

$R_{BK}$	relevador de respaldo cuyo ajuste en zona 2 debe ser determinado;
$R_{Pri}$	relevador primario protegiendo la línea B-C <sub>i</sub> ;
$Z_{AB}$	impedancia de secuencia positiva de la línea A-B;
$K2$	fracción de las líneas remotas protegidas por el relevador en zona 2 (este valor es 5% menos que el porcentaje la línea cubierto por la zona 1);
$ZA(R_{BK}, F_i)$	Impedancia aparente vista por el relevador RBK para la falla $F_i$ ;
$Z2(R_{BK}, R_{Pri})$	Posible ajuste de la zona 2 para el relevador $R_{BK}$

Sin embargo, si la impedancia aparente vista por el relevador para la falla es infinita, entonces es ajustado a  $1.20 * Z_{AB}$ .

- 3) Repetir los pasos 2 y 3 para fallas al final de la zona 1 de las otras líneas.
- 4) Seleccionar el menor ajuste para la zona 2 del relevador, calculados en los pasos 2 a 4.

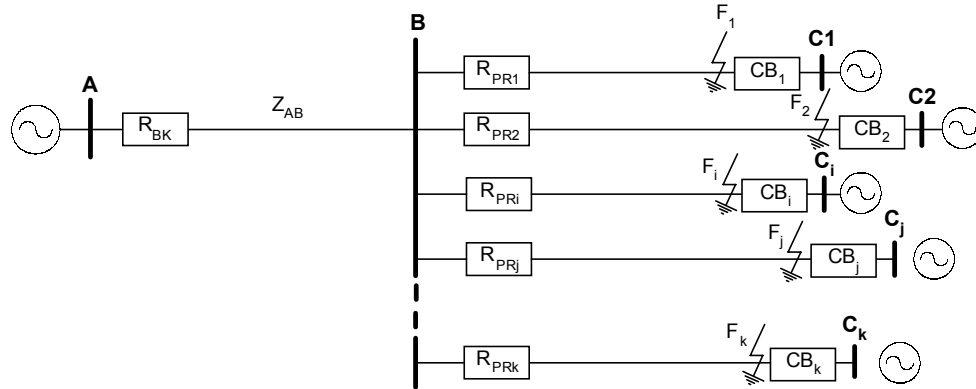


Figura 4.4 Localización de fallas para determinar el ajuste en zona 2.

- 5) Simular una falla en el alcance de la zona 1 del relevador primario denominado  $R_{BK}$ ,  $i = 1, 2, \dots, k$ , con todos los buses excepto el B. Todos los interruptores deben estar cerrados. Determinar la impedancia vista por el relevador  $R_{BK}$ .
- 6) Evaluar si la impedancia vista por el relevador  $R_{BK}$  para las fallas simuladas en el paso 5 cae dentro de la zona de operación definido por el ajuste obtenido en el paso 4.
- 7) Si el relevador opera, entonces se debe reducir el ajuste para la zona 2 en un alcance del 90% de la impedancia de falla menor vista por el relevador  $R_{BK}$ , que ocasiona la operación en el paso 6.
- 8) Repetir los pasos 1 al 8 para determinar el ajuste de la zona 2 de los relevadores restantes en el sistema.

Para estudiar con mayor detalle este algoritmo se recomienda ver [5,35,43].

Este algoritmo de protección, puede operar como un esquema de protección adaptable si es ejecutado cada vez que el sistema de potencia cambie en sus condiciones de operación, así los ajustes de los relevadores siempre estarán en el valor adecuado y de esta manera, se satisface con las características de confiabilidad evaluada en términos de la dependabilidad y seguridad, y con un máximo alcance de protección de respaldo de la zona 2.

#### **4.4. Comparación de ambos esquemas de protección ante distintas fallas en el sistema eléctrico.**

Para realizar la comparación entre estos métodos se hicieron simulaciones de distintos tipos de fallas en EMTDC/PSCAD y de dichas simulaciones se puede decir que el esquema de protección adaptable se comporta de manera más eficiente que el esquema de protección convencional, esto debido a que, como se mencionó antes, cuando un elemento del sistema queda fuera del sistema por alguna liberación de falla, o por mantenimiento, la configuración del sistema cambia, por lo tanto, también las corrientes de corto circuito y los flujos de potencia se ven afectados, y es cuando el sistema de protección necesita de un nuevo ajuste (siempre y cuando el nuevo estudio de fallas presente cambios suficientes como para realizar dicho ajuste). Si el ajuste es necesario y se realiza, se tendrá por consecuencia una protección de distancia deseada o calculada para proteger de la mejor manera posible al sistema de potencia para esas condiciones de operación.

Un ejemplo para hacer notar la diferencia existente entre los ajustes con métodos convencionales y los ajustes con métodos adaptables, puede verse al simular fallas al final de la zona 1 de las líneas remotas al relevador 26, descrito anteriormente en la Figura 4.3.

Los porcentajes de alcance obtenidos para condiciones de máxima generación por ambos métodos se pueden observar en la Figura 4.5., donde se puede ver claramente que se tiene un mayor alcance cuando el ajuste de la zona 2 del relevador  $R_{26}$  se obtiene mediante técnicas de protección adaptable [35].

De igual manera se observa el porcentaje de alcance para condiciones de mínima generación como se muestra en la Figura 4.6. en donde el alcance del relevador  $R_{26}$  con ajuste adaptable es de 49% en la línea  $L_{12-17}$ , mientras que con ajuste convencional solo alcanza el 4%.

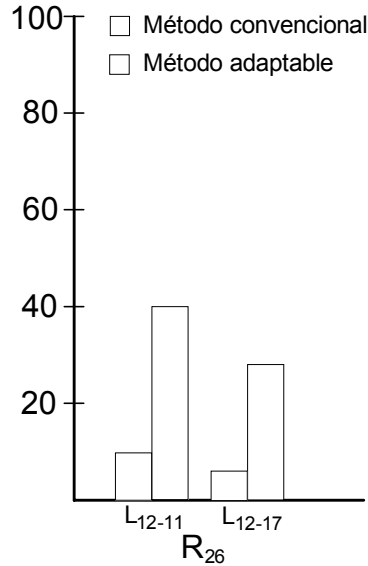


Figura 4.5. Porcentaje de alcance en zona 2 para líneas remotas en condición de máxima generación.

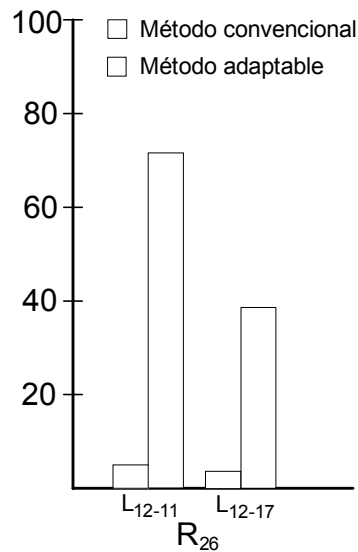


Figura 4.6. Porcentaje de alcance en zona 2 para líneas remotas en condición de mínima generación.

Con esto se demuestra la ventaja que presentan los ajustes con métodos adaptables comparados con los ajustes con métodos convencionales. Cabe aclarar que estos porcentajes son relativos, y dependen directamente de la configuración del sistema de potencia.

## CAPÍTULO V

### **5. ANÁLISIS DE RESULTADOS.**

#### **5.1. Introducción.**

Es importante mencionar que los sistemas de protección adaptable requieren de un centro de control de protecciones, donde se llevan a cabo los estudios para proponer los ajustes en los relevadores que así lo requieran cuando las condiciones del sistema hayan cambiado, para de esta manera, mantener el mejor alcance y coordinación entre los relevadores primarios y de respaldo [5,33,44,45].

En este trabajo, la protección adaptable solo se implementó para el ajuste de la zona 2 en el DSP, para evaluar la operación, o no operación de dicho relevador, en una determinada condición de operación del sistema. El ajuste se realiza de manera manual con ayuda del programa desarrollado en LabVIEW que es el encargado de interactuar con el DSP, en vez de ser realizado en forma automática respondiendo a las condiciones de operación del sistema. En este caso, los nuevos ajustes se actualizan en el DSP mediante el estudio fuera de línea para determinar dicho ajuste.

En este capítulo se presenta el análisis de resultados que fueron obtenidos al realizar distintos tipos de pruebas al algoritmo, como son: fallas trifásicas y monofásicas en distintos puntos de la línea a proteger. Los primeros resultados (prueba 1 y 2) muestran el comportamiento del algoritmo implementado en el DSP para comprobar el cálculo correcto de la impedancia de falla. En los siguientes resultados, únicamente se prueba el algoritmo simulado en MATLAB.

En algunos casos, debido a la configuración del sistema, se muestra una diferencia en la impedancia de falla o de secuencia positiva calculada por el algoritmo. De esta forma puede comprobarse cuándo es aplicable la filosofía de protección adaptable.

## 5.2. Representación esquemática del DSP como relevador de distancia.

Debido a las limitaciones del equipo para representar el sistema de protección completo, se recurre a la simulación de algunos de estos elementos del sistema; como es el caso del interruptor de la línea, la simulación de fallas, y el modelado del sistema eléctrico de potencia. Por ejemplo, la operación del interruptor de la línea, se simula con una señal de CD del microcontrolador DSP cuando se detecta una falla dentro de la zona de operación, esto quiere decir que cuando es detectada una falla, el DSP genera una señal de CD que enciende un led (diodo emisor de luz). En la simulación de los casos de estudio y el modelado del sistema de potencia, se utiliza un programa con el software comercial EMTDC/PSCAD.

La información de voltajes y corrientes de falla procesados por el microcontrolador DSP, es manejada con un programa desarrollado con el software comercial de instrumentación virtual (LabVIEW). Adicionalmente, se realizó un programa digital del algoritmo de protección en lenguaje de MATLAB, para comparar los resultados que se obtuvieron con el algoritmo desarrollado e implementado en el DSP.

Una representación esquemática de la protección de una línea de transmisión con el microcontrolador se presenta en la Figura 5.1.

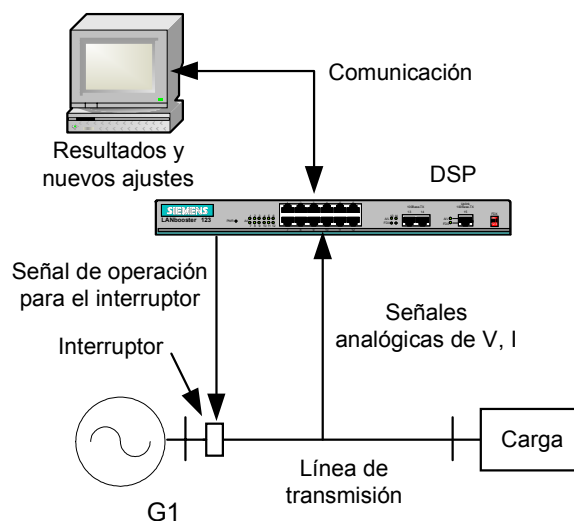


Figura. 5.1. Diagrama esquemático del DSP como protección.

En la Figura 5.1., se puede observar el diagrama del sistema de protección desarrollado en este trabajo, en donde se muestra al microcontrolador DSP, tomando las señales de voltaje y corriente del elemento a proteger (línea de transmisión), y la comunicación que este microcontrolador tiene con la computadora encargada de realizar los nuevos ajustes cuando sean requeridos.

### 5.3. Sistemas de prueba para el DSP.

En la simulación de las fallas trifásicas y monofásicas, se usó el software comercial EMTDC/PSCAD, el modelo del sistema eléctrico utilizado en las simulaciones se presenta en la Figura 5.2. que muestra dos fuentes equivalentes conectadas por medio de una línea de transmisión de 100 Km. de longitud (TIY). Las señales que son utilizadas para calcular la impedancia de secuencia positiva de la línea se toman de los medidores  $V_{as}$ ,  $V_{bs}$ ,  $V_{cs}$ ,  $I_{as}$ ,  $I_{bs}$  e  $I_{cs}$ .

Los resultados que se muestran en el primer caso son resultado de las simulaciones de una falla trifásica sólida al final de la línea (prueba 1) y una falla monofásica sólida (fase A a tierra), también al final de la línea (prueba 2). Las fallas son simuladas al final de la línea para que la impedancia de falla medida por el relevador, pueda ser comparada con la impedancia de secuencia positiva del modelo de la línea de transmisión, proporcionada por el software EMTDC/PSCAD y está en función de la longitud de la línea, para este caso es de 100 Km. con una impedancia de secuencia positiva total de  $37.88\angle 87.17$  ohms.

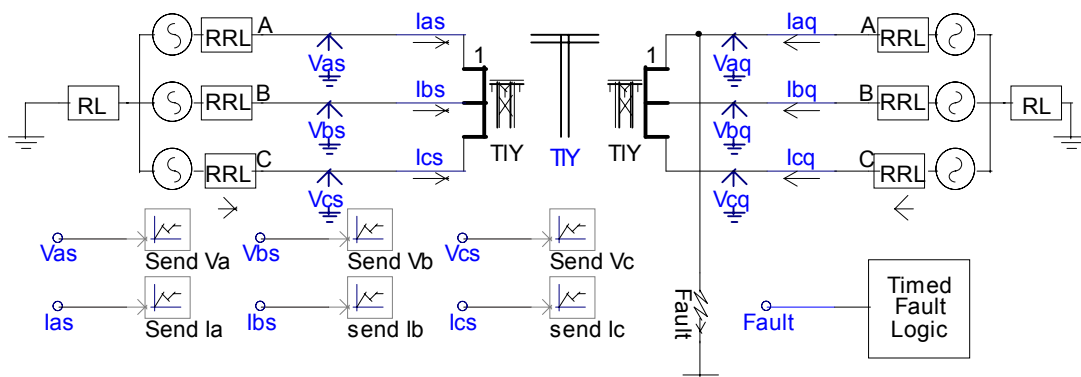


Figura 5.2. Programa gráfico de EMTDC/PSCAD

### 5.3.1 Resultados de la prueba 1

Se tomaron las muestras de voltajes y corrientes de las simulaciones realizadas en EMTDC/PSCAD de una falla trifásica al final de una línea de transmisión de 100 Km. (mostrada en la Figura 5.2). La simulación se realiza en el intervalo de 0 - 1 segundo, la falla es simulada en  $t = 0.5$  s. Se registraron dos ciclos antes de la falla y cuatro ciclos después, para estudiar el comportamiento del relevador durante el transitorio de la falla, porque es el periodo que presenta el mayor problema para los filtros digitales.

En esta prueba, las muestras procesadas de voltaje y corriente son de una sola fase (fase A). Posteriormente, se adecuaron y escalaron los resultados en MATLAB para simular la operación en tiempo real del convertidor analógico digital y para que las señales de entrada fueran similares a los valores que recibe el ADC, cuando está recibiendo señales analógicas en una medición real.

Los resultados se presentan en la Tabla 5.1, en las columnas se muestran los 6 ciclos de estudio y en las filas, las magnitudes y ángulos (en grados) de voltaje, corriente y la impedancia de falla medida. Se puede observar que durante el ciclo uno y dos la impedancia medida por el DSP es muy alta comparada con la impedancia real de la línea esto indica que se está midiendo la impedancia de la carga conectada, es decir, que aún no ocurre la falla. Sin embargo a partir del ciclo 3 esta impedancia medida cambia notoriamente hasta ser igual o muy semejante a la impedancia de secuencia positiva de la línea. Además se presenta un incremento en la magnitud de la corriente. Puede concluirse que se presentó una condición de falla.

Tabla 5.1. Resultados de la falla trifásica.

Resultados	Ciclo1	Ciclo2	Ciclo3	Ciclo4	Ciclo5	Ciclo6
Fasor de Voltaje en DSP	$3.02 \angle -162.82$	$3.02 \angle -162.83$	$2.61 \angle -160.32$	$2.60 \angle -160.39$	$2.60 \angle -160.42$	$2.60 \angle -160.42$
Fasor de Voltaje en LabVIEW	$3.02 \angle -165.85$	$3.02 \angle -162.87$	$2.59 \angle -160.38$	$2.60 \angle -160.42$	$2.60 \angle -160.44$	$2.60 \angle -160.47$



Tabla 5.1. continuación...

.Resultados	Ciclo1	Ciclo2	Ciclo3	Ciclo4	Ciclo5	Ciclo6
Fasor de corriente en DSP	$0.0185 \angle -164.8$	$0.0185 \angle -164.8$	$0.0688 \angle 112.4$	$0.0685 \angle 112.4$	$0.0686 \angle 112.4$	$0.0685 \angle 112.4$
Fasor de corriente en LabVIEW	$0.0187 \angle -164.8$	$0.0187 \angle -164.9$	$0.0685 \angle 112.4$	$0.0686 \angle 112.4$	$0.0687 \angle 112.4$	$0.0687 \angle 112.3$
Fasor de Impedancia DSP	$161.40 \angle 2.03$	$161.28 \angle 2.02$	$37.88 \angle -272.23$	$37.99 \angle -272.83$	$37.97 \angle -272.82$	$37.96 \angle -272.80$
Fasor de Impedancia LabVIEW	$161.39 \angle 2.01$	$161.26 \angle 2.03$	$37.91 \angle -272.83$	$38.01 \angle -272.88$	$37.98 \angle -272.86$	$37.96 \angle -272.85$

En la Figura 5.3, se presentan gráficamente los resultados obtenidos con el algoritmo desarrollado en LabVIEW. Puede observarse en la gráfica de la impedancia, que después de un ciclo completo de falla, se tiene el valor correcto de impedancia medida, aún cuando en la corriente se presenta un transitorio con señales de CD y de CD decreciente.

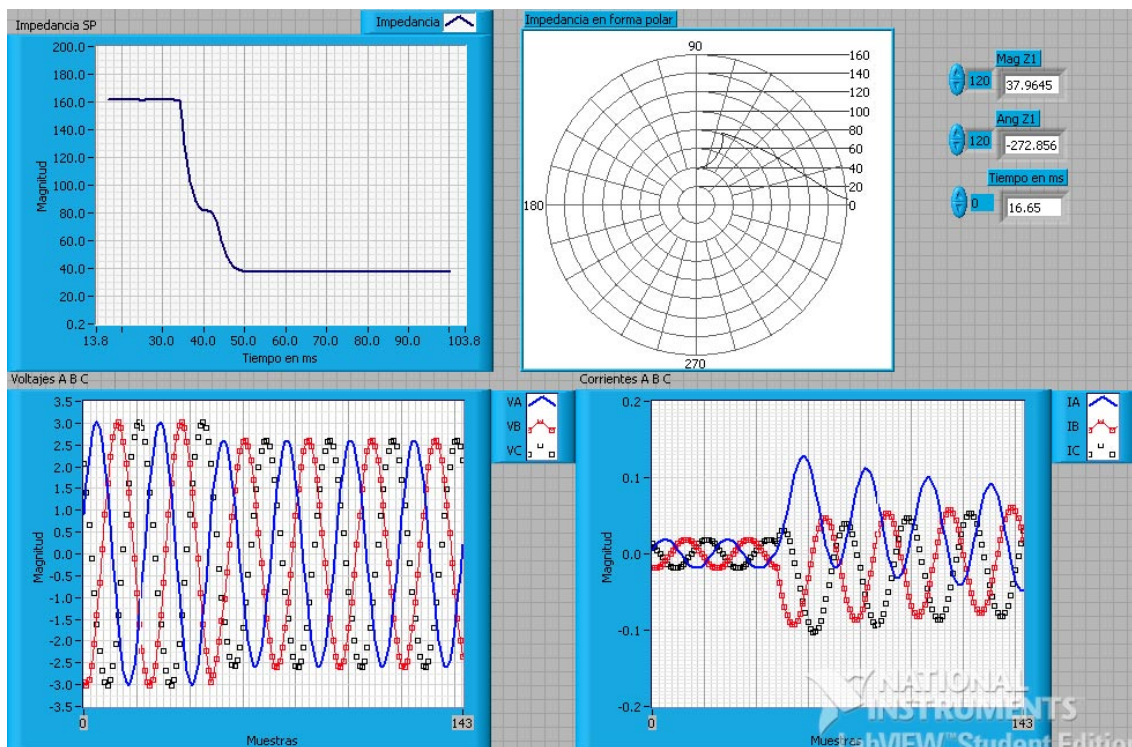


Figura. 5.3. Prueba al algoritmo con una falla trifásica.

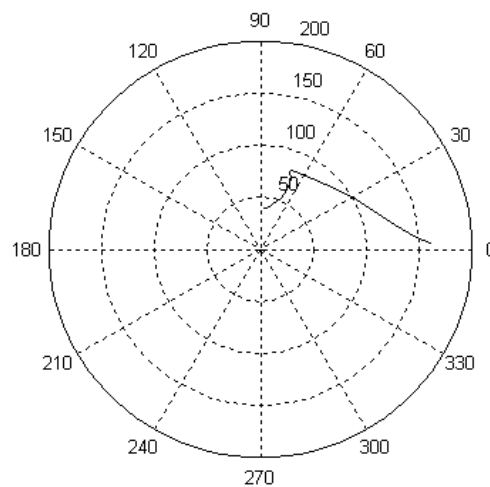
En estos resultados se observa una diferencia mínima que puede ser calculada como un error relativo de 0.002903, como se muestra en la ecuación 5.1.

$$Error = \frac{resPSCAD - resDSP}{resPSCAD} = \frac{37.88 - 37.99}{37.88} = 0.002903 \quad (5.1)$$

donde:

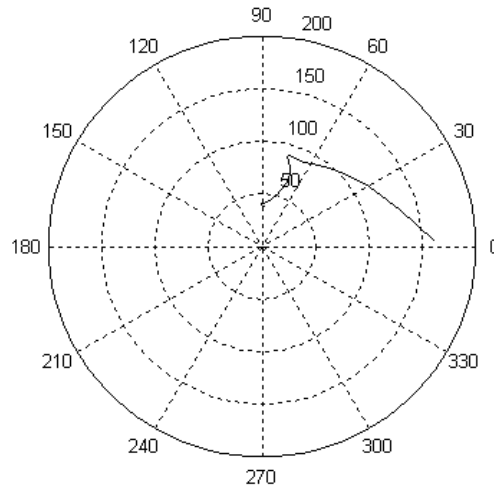
- Error* = es el error relativo.
- resPSCAD* = resultado arrojado por la simulación en EMTDC/PSCAD.
- resDSP* = resultado calculado por el Microcontrolador DSP.

En la Figura. 5.4. se presentan los resultados obtenidos mediante un programa desarrollado en MATLAB® para comparar con los obtenidos con el DSP. Puede observarse que son iguales con los presentados en la gráfica de la impedancia en forma polar de la Figura 5.3.



*Figura. 5.4. Impedancia con todas las muestras de la simulación.*

Para comprobar que se tienen los valores correctos de la impedancia de falla y en el menor tiempo posible por el algoritmo implementado, se realizaron simulaciones con el programa desarrollado en MATLAB durante dos ciclos de falla, uno antes de la falla y otro después de la falla, los resultados se presentan en la Figura. 5.5.



*Figura. 5.5. Falla trifásica con muestras de 2 ciclos, uno antes de la falla y uno después de la misma.*

Con estos resultados se prueba que es suficiente procesar un ciclo a la frecuencia fundamental después del inicio de la falla, para tener una medición correcta de la impedancia de falla medida durante la contingencia.

### **5.3.2. Resultados de la prueba 2**

Se simuló en EMTDC/PSCAD una falla monofásica (fase A) en un circuito trifásico, mostrado en la Figura 5.2. La simulación se realiza durante el intervalo de  $t = 0 - 1$  s., la falla es simulada en  $t = 0.5$  s.

En la Tabla 5.2 se presentan los resultados de 6 ciclos, 2 ciclos antes de la falla y 4 ciclos después de esta.

Tabla 5.2. Resultados de la falla monofásica.

Resultados	Ciclo1	Ciclo2	Ciclo3	Ciclo4	Ciclo5	Ciclo6
Fasor de Voltaje en DSP	$3.02 \angle -162.82$	$3.02 \angle -162.83$	$2.71 \angle -161.36$	$2.71 \angle -161.38$	$2.71 \angle -161.45$	$2.71 \angle -161.46$
Fasor de Voltaje en LabVIEW	$3.02 \angle -165.85$	$3.02 \angle -162.87$	$2.69 \angle -161.33$	$2.70 \angle -161.39$	$2.70 \angle -161.42$	$2.70 \angle -161.45$

Tabla 5.2. continuación...

Resultados	Ciclo1	Ciclo2	Ciclo3	Ciclo4	Ciclo5	Ciclo6
Fasor de corriente en DSP	$0.0184 \angle -164.8$	$0.0185 \angle -164.8$	$0.0685 \angle 111.7$	$0.0704 \angle 111.5$	$0.0705 \angle 111.3$	$0.0685 \angle 111.2$
Fasor de corriente en LabVIEW	$0.0187 \angle -164.8$	$0.0187 \angle -164.9$	$0.0685 \angle 111.7$	$0.0693 \angle 111.4$	$0.0702 \angle 111.3$	$0.0702 \angle 111.3$
Fasor de Impedancia DSP	$161.41 \angle 2.04$	$161.22 \angle 2.05$	$39.36 \angle -272.85$	$38.38 \angle -272.85$	$38.19 \angle -272.82$	$38.17 \angle -272.81$
Fasor de Impedancia LabVIEW	$161.39 \angle 2.01$	$161.26 \angle 2.03$	$39.30 \angle -273.07$	$38.40 \angle -272.86$	$38.21 \angle -272.80$	$38.18 \angle -272.80$

En la Figura 5.6, se presentan gráficamente los resultados obtenidos con el algoritmo desarrollado. Y como puede observarse en la gráfica de voltajes y corrientes, correspondientes a una falla monofásica, la impedancia de falla es obtenida en tan solo el primer ciclo después de iniciada la condición de falla.

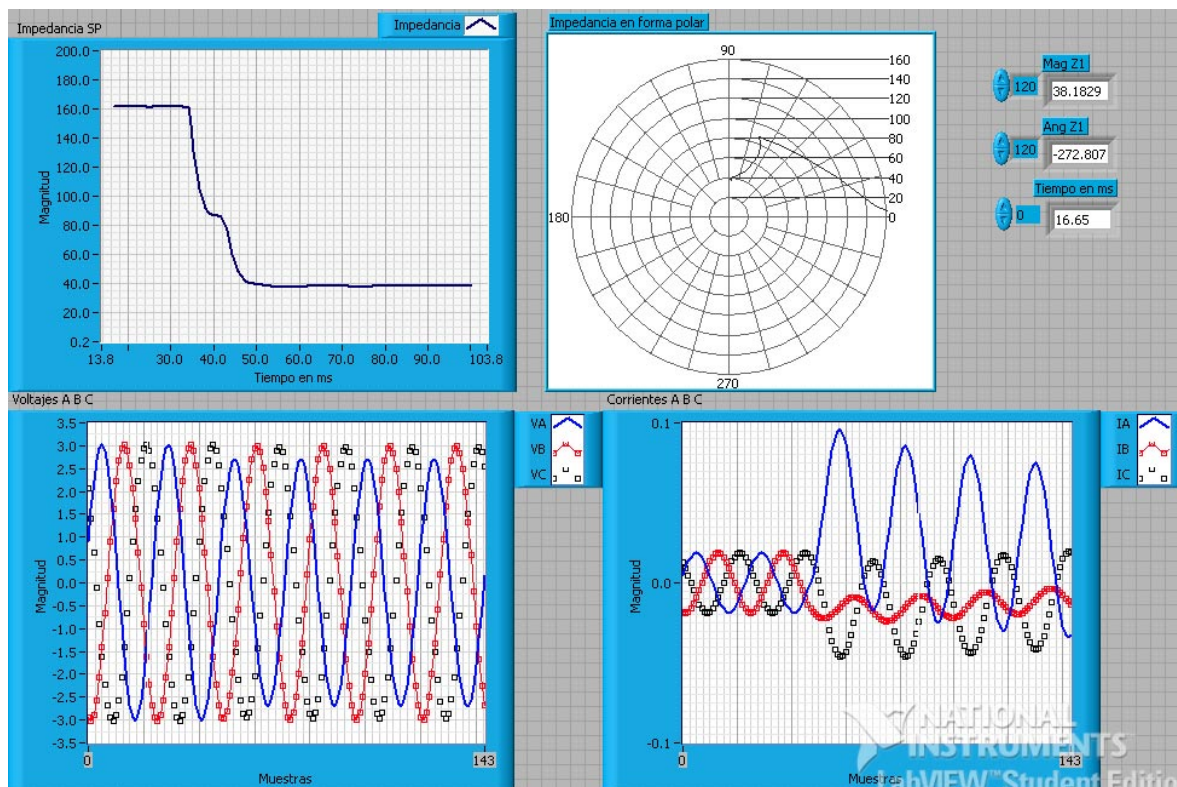
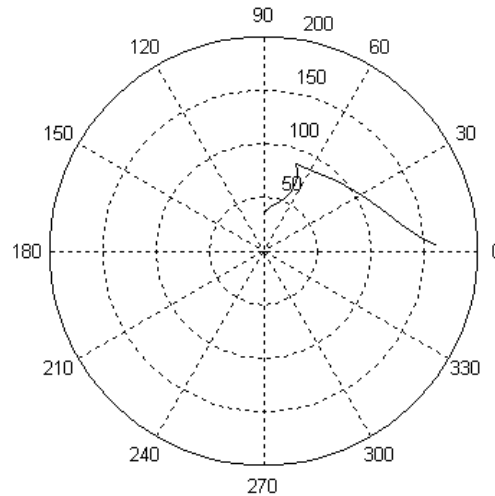


Figura 5.6. Prueba al algoritmo con una falla monofásica.

Los resultados obtenidos con el programa desarrollado en MATLAB se presentan en la Figura 5.7. y no se aprecian cambios, comparados con los resultados obtenidos con LabVIEW y el DSP.



*Figura 5.7. Comprobación de la falla monofásica en MATLAB.*

#### **5.4. Sistemas eléctricos de potencia utilizados.**

Las pruebas anteriores son realizadas para comprobar el comportamiento del algoritmo implementado en el DSP ante los diferentes tipos de fallas (trifásicas y monofásicas en zona 1), es decir, para verificar que la impedancia medida por el DSP sea la correcta, por esta razón únicamente se utilizó un sistema eléctrico de dos generadores interconectados por una sola línea de transmisión en donde se simularon fallas al final de la línea.

A continuación, se presentan simulaciones de fallas en sistemas eléctricos de potencia, en que pueden utilizarse esquemas de protección adaptable para evaluar su comportamiento ante las contingencias, o cambios en la topología del sistema de potencia o alguna condición de operación del sistema que requiera actualizar los ajustes en los relevadores; de tal manera se tenga una protección confiable, evaluada en términos de dependabilidad y seguridad, y así evitar la salida innecesaria o el retardo excesivo en la liberación de fallas, en los elementos del sistema eléctrico de potencia.

### 5.4.1. Simulación de fallas en el sistema eléctrico de prueba (S.E.P.).

El sistema de potencia modelado en EMTDC/PSCAD se puede observar en la Figura 5.8. El sistema tiene un voltaje de 230 KV. y está formado por 4 líneas de transmisión y 5 generadores de potencia. Este sistema de prueba, se usa para observar el comportamiento de las protecciones cuando ocurren cambios en su configuración, posteriormente si se realizan pruebas a sistemas reales.

Primeramente, se simulan fallas al 80 % de cada una de las líneas remotas al relevador, para evaluar la zona 2, se determina el menor factor de *infeed* y se continúa con el algoritmo de protección adaptable descrito en el capítulo 4. Una vez obtenidos los valores de ajustes para la zona 2 de los relevadores, se tiene la mayor distancia protegida para esa condición de operación. Para mantener esta condición en todo momento, es necesario monitorear al sistema continuamente, y si estas condiciones cambian, entonces será necesario realizar nuevamente las simulaciones para calcular los nuevos ajustes, y actualizarlos en los relevadores que lo requieran.

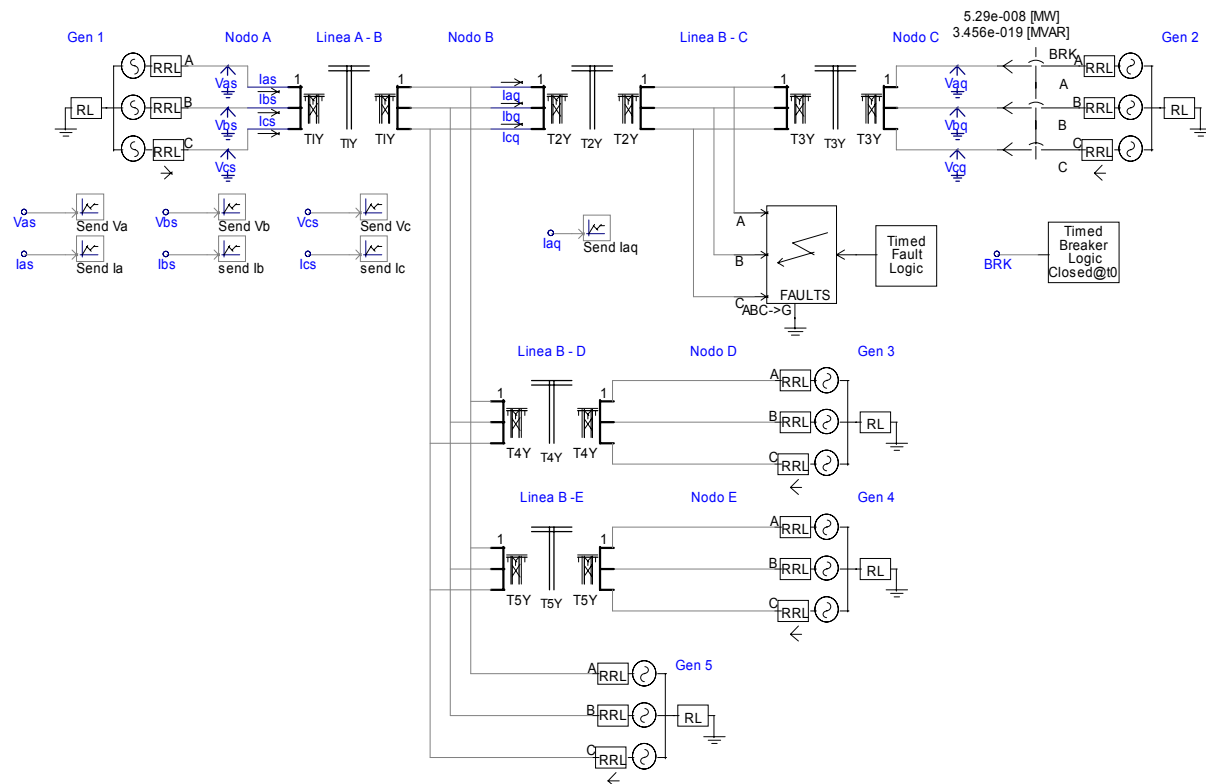


Figura 5.8. Sistema eléctrico de prueba usado en las simulaciones.

**Prueba S.E.P.1.:** Esta prueba consiste en realizar fallas al 80% de la línea B-C del sistema de potencia mostrado en la Figura 5.8. Se realizaron 4 fallas en esta línea haciendo variar la configuración del sistema. Los resultados de estas fallas se muestran en la Tabla 5.3. y también en el plano complejo en la Figura 5.9.

Tabla 5.3. Impedancias de falla de la prueba S.E.P.1.

No. de Falla	Impedancia de Secuencia Positiva	Elementos en el sistema
1	<b>206 <math>\angle</math> 66.29</b>	Con todos los elementos
2	175 $\angle$ 76.43	Con la línea B-D fuera
3	190 $\angle$ 77.1	Con la línea B-E fuera
4	<b>127 <math>\angle</math> 65.9</b>	Con el Generador 5 fuera

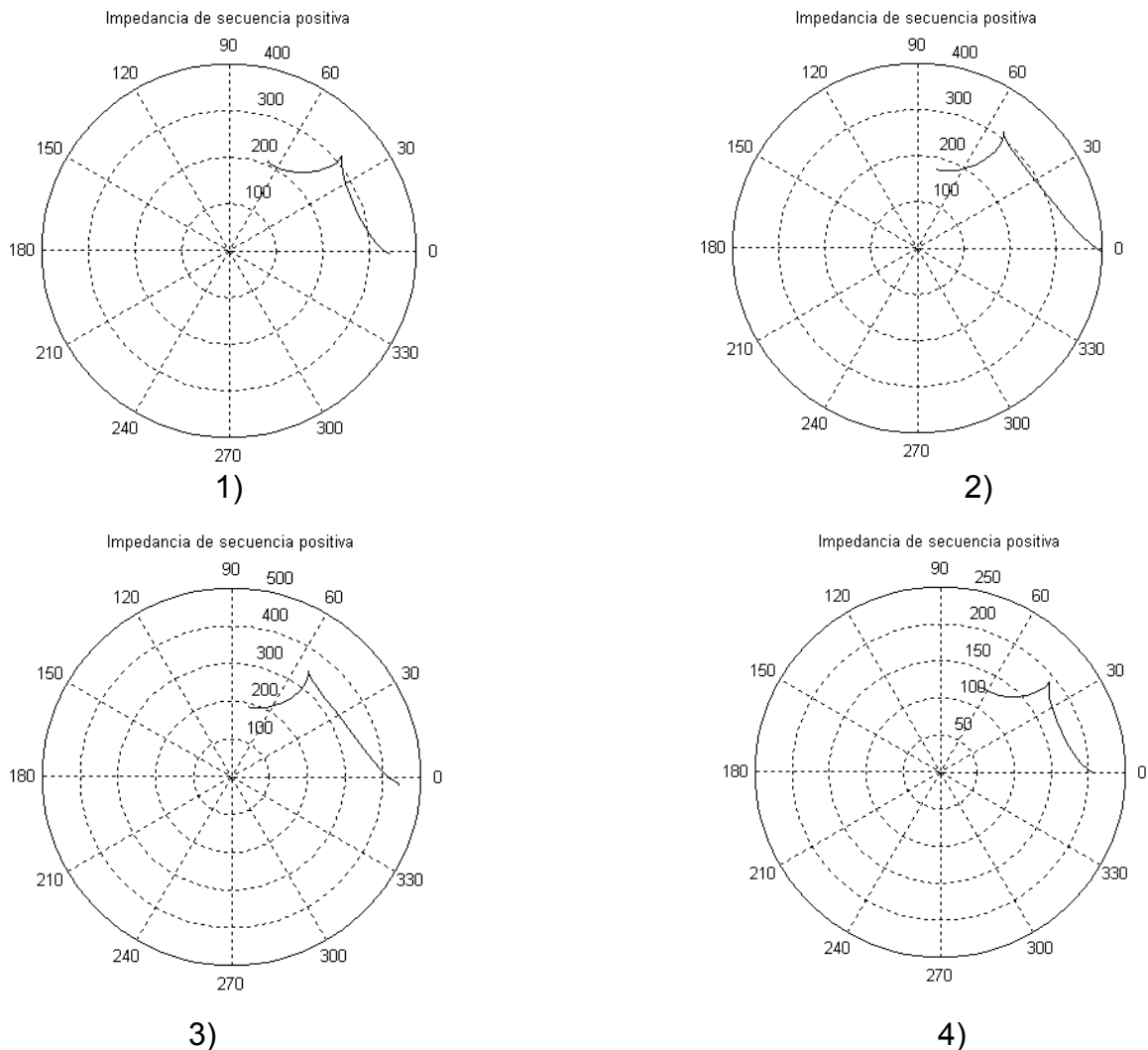


Figura 5.9. Gráficas de la impedancia de falla de la Tabla 5.3



**Prueba S.E.P.2.:** Esta prueba consiste en realizar fallas al 80% de la línea B-D del sistema de potencia mostrado en la Figura 5.8. Se realizaron 4 fallas en esta línea considerando modificaciones en la configuración del sistema. Los resultados de estas fallas se muestran en la Tabla 5.4., así como también se observa la trayectoria de la impedancia de falla en el plano complejo en le Figura 5.10

Tabla 5.4. Impedancias de falla de la prueba S.E.P.2.

No. de Falla	Impedancia de Secuencia Positiva	Elementos en el sistema
1	<b>93.5 <math>\angle</math> 79.6</b>	Con todos los elementos
2	86.9 $\angle$ 83.7	Con la línea B-C fuera
3	83.9 $\angle$ 83.7	Con la línea B-E fuera
4	<b>64.5 <math>\angle</math> 79.1</b>	Con el Generador 5 fuera

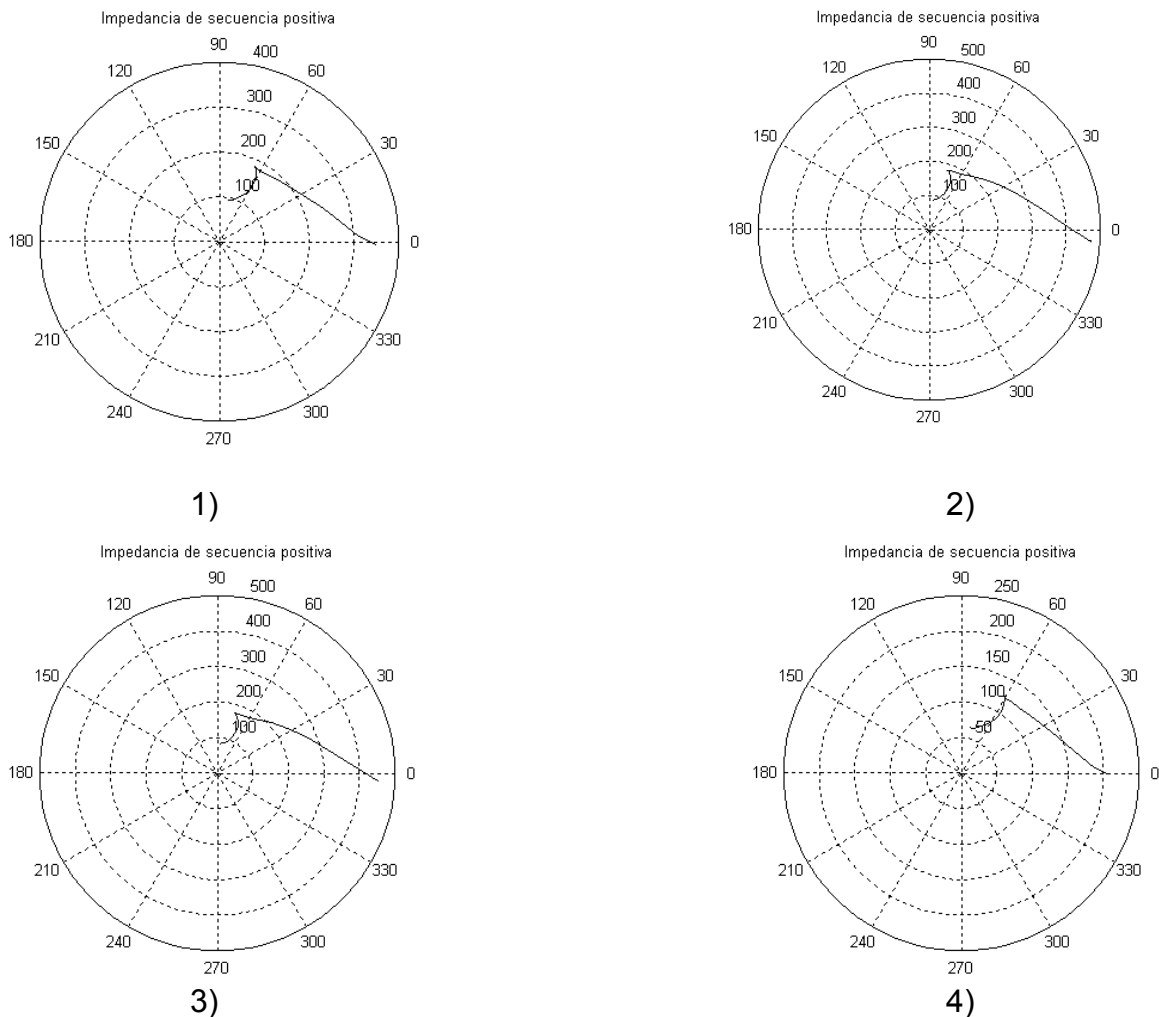


Figura 5.10. Gráficas de la impedancia de falla de la Tabla 5.4



**Prueba S.E.P.3.:** Esta prueba consiste en realizar fallas al 80% de la línea B-E del sistema de potencia mostrado en la Figura 5.8. Se realizaron 4 fallas en esta línea haciendo variar la configuración del sistema. Los resultados de las impedancias de posfalla en magnitud y ángulo se muestran en la Tabla 5.5., así como su trayectoria durante las condiciones de prefalla, falla y posfalla en la Figura 5.11.

Tabla 5.5. Impedancias de falla de la prueba S.E.P.3.

No. de Falla	Impedancia de Secuencia Positiva	Elementos en el sistema
1	<b>145.9 <math>\angle</math> 73.6</b>	Con todos los elementos
2	136.8 $\angle$ 80.9	Con la línea B-C fuera
3	121.7 $\angle$ 80.5	Con la línea B-D fuera
4	<b>93.39 <math>\angle</math> 73.16</b>	Con el Generador 5 fuera

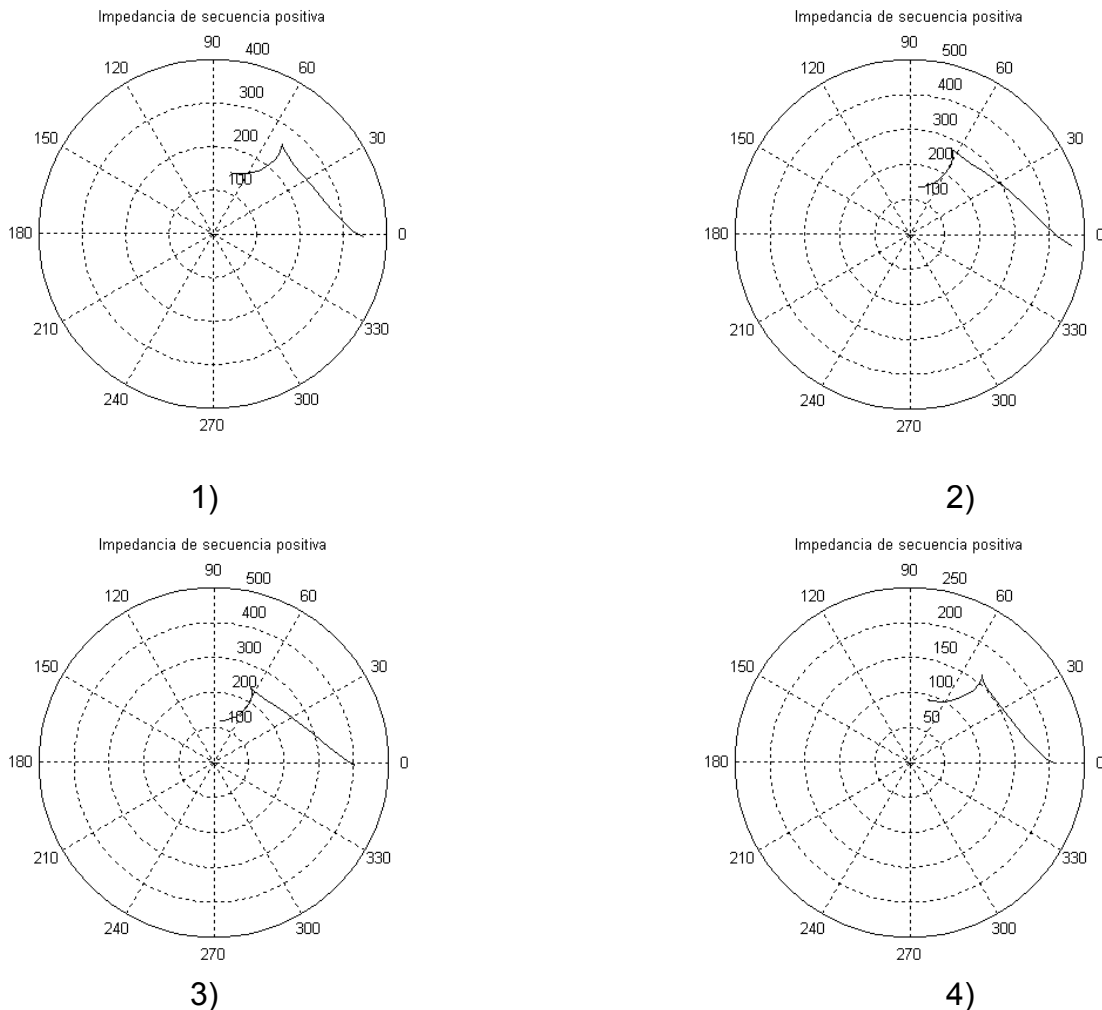


Figura 5.11. Gráficas de las impedancias de falla de la Tabla 5.5

Con los resultados anteriores puede analizarse el comportamiento de la impedancia de falla ante las condiciones de operación del SEP. Por ejemplo, la falla 1 de cada una de las pruebas (configuración del sistema con todos los elementos conectados), se puede deducir que si se desea cubrir el 80% de la línea remota sin tener traslapes, se debe elegir la menor impedancia, la cual es de  $93.5 \angle 79.66$  Ohms y cubre el 80% de la línea B-D (Prueba S.E.P.2, falla 1). Sin embargo, si por alguna razón (debido a alguna falla o mantenimiento) el generador 5 queda fuera del sistema (Prueba S.E.P.2, falla 4), entonces la impedancia de ajuste ya no es la adecuada porque en la simulación se obtiene una impedancia de falla de  $64.5 \angle 79.1$  Ohms, y como consecuencia se tienen problemas de sobrealcance y posiblemente la salida innecesaria de la línea B-D al presentarse una falla más allá de la zona de protección a la que fue ajustada en el relevador.

En el análisis de resultados anterior, se demuestra la necesidad de realizar un nuevo ajuste, cuando se presente esta nueva configuración del sistema, para que el relevador continúe ofreciendo la protección adecuada en todo momento.

En las Figuras 5.9.,5.10. y 5.11. se observa el comportamiento de cada una de las fallas realizadas en las pruebas. En estas figuras solo se tienen dos ciclos de prueba, uno de prefalla y uno de posfalla para mostrar que efectivamente después de un ciclo completo de falla, se obtiene el resultado correcto de la medición de impedancia de secuencia positiva que corresponde desde la ubicación del relevador al punto de falla, en el cual puede estar presente el efecto de *infeed*.

En estas simulaciones no se realiza la prueba con el DSP principalmente porque ya se mostró en la sección (5.3), que su comportamiento es similar a los resultados obtenidos por el programa desarrollado en MATLAB. Además, estas simulaciones son realizadas para mostrar la necesidad de realizar ajustes en las protecciones cuando ocurra un cambio en la configuración del sistema. Cabe mencionar que en algunas ocasiones y dependiendo de las características del sistema eléctrico, no es necesario realizar nuevos ajustes, pues aunque la configuración del sistema haya cambiado, aún se tiene el ajuste adecuado para seguir cubriendo el porcentaje establecido de la línea. Esto se puede observar en el siguiente sistema.

### 5.4.2. Simulación de fallas en el sistema eléctrico real 1 (S.E.R.1).

Las siguientes simulaciones son realizadas utilizando modelos de sistemas eléctricos reales, proporcionados por una empresa dedicada al estudio de protecciones [46].

Prueba S.E.R.1. Se simularon fallas al 80% de la línea B-C, para determinar el ajuste de la zona 2 del relevador ubicado en el nodo A del sistema eléctrico de la Figura 5.12. Se realizaron 8 fallas en esta línea considerando distintas configuraciones del sistema para observar los cambios en la impedancia de falla medida por el relevador. Los resultados de estas fallas se muestran en la Tabla 5.6. y el comportamiento de la impedancia en el plano complejo en la Figura 5.13.

En este sistema eléctrico, las simulaciones de fallas realizadas después de cambiar la configuración, no presentan cambios considerables en la impedancia de falla, por lo que se deduce que no es necesario reajustar al relevador.

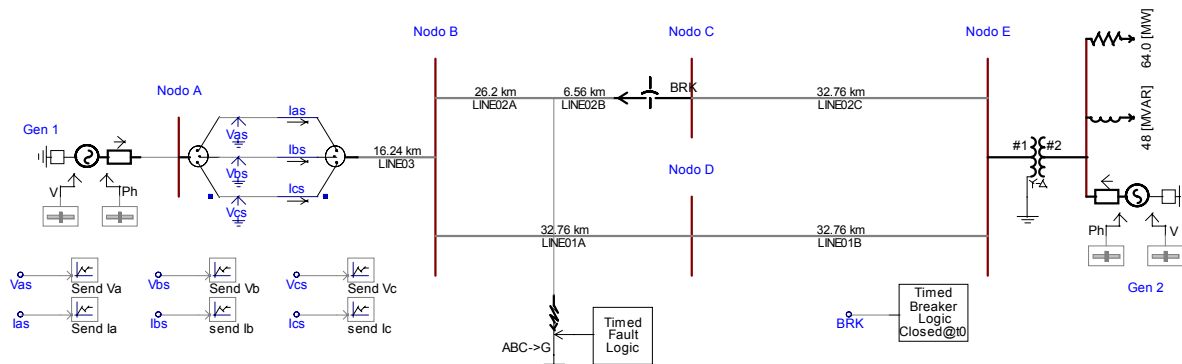


Figura 5.12. Sistema eléctrico real 1.

Tabla 5.6. Impedancias de falla de la prueba S.E.R.1.

No. de Falla	Impedancia de Secuencia Positiva	Elementos en el sistema
1	$22.1 \angle 66.1$	Con todos los elementos
2	$21.8 \angle 68.1$	Con la línea B-D fuera
3	$21.8 \angle 68.1$	Con la línea D-E fuera
4	$22.1 \angle 66.2$	Con la línea C-E fuera
5	<b><math>22.4 \angle 67.7</math></b>	Sin la carga de 64 MW
6	$22.4 \angle 65.1$	Sin carga de 48 MVAR
7	$23.1 \angle 66.9$	Sin cargas
8	<b><math>21.2 \angle 67.3</math></b>	Con el Generador 2 (Gen 2) fuera

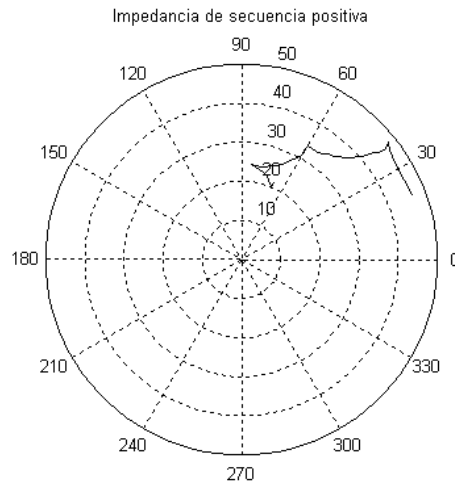


Figura 5.13. Gráfica de la impedancia de falla de la Tabla 5.6.

### 5.4.3. Simulación de fallas en el sistema eléctrico real 2 (S.E.R.2).

Prueba S.E.R.2. Se realizaron fallas simuladas al 80% de la línea B-C, que recibe una protección de respaldo en zona 2 por el relevador ubicado en el nodo A del sistema eléctrico de la Figura 5.14. Se realizaron 11 fallas en esta línea haciendo variar la configuración del sistema para observar los cambios en la impedancia de falla medida. Los resultados de estas fallas se muestran en la Tabla 5.7. y el comportamiento de la impedancia en el plano complejo en la Figura 5.15.

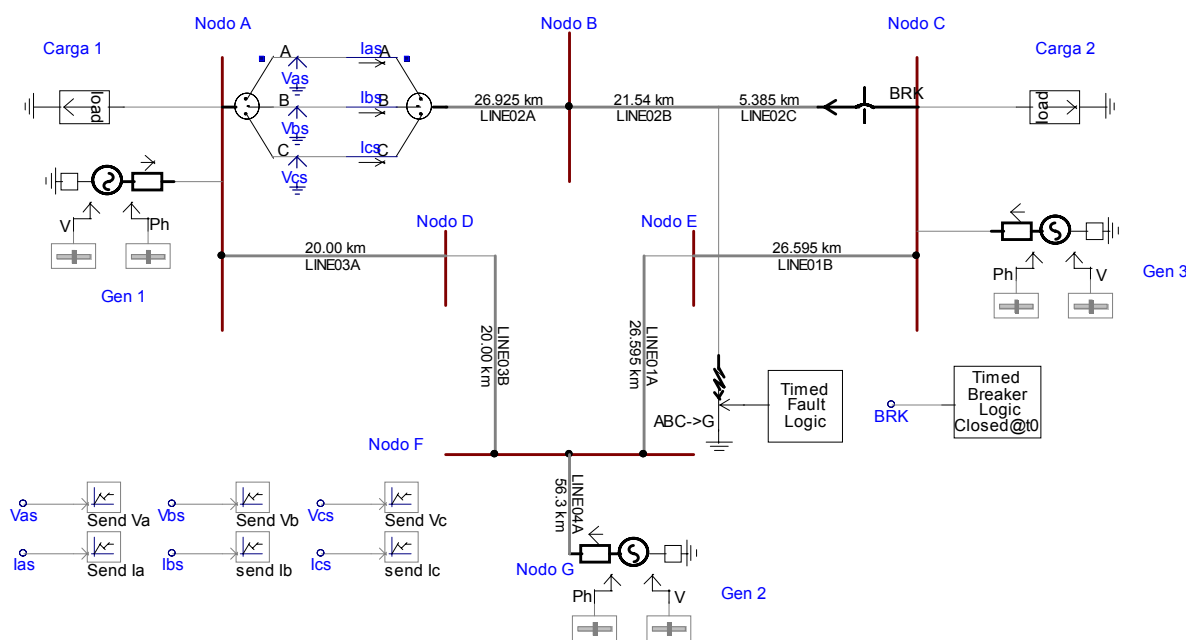


Figura 5.14. Sistema eléctrico real 2.

Tabla 5.7. Impedancias de falla de la prueba S.E.R.2.

No. de Falla	Impedancia de Secuencia Positiva	Elementos en el sistema
1	<b>26.2 <math>\angle</math> 71.1</b>	Con todos los elementos
2	<b>26.1 <math>\angle</math> 71.1</b>	Con la línea A-D fuera
3	26.1 $\angle$ 71.1	Con la línea D-F fuera
4	26.1 $\angle$ 71.1	Con la línea E-F fuera
5	26.1 $\angle$ 71.1	Con la línea C-E fuera
6	26.2 $\angle$ 71.1	Con la línea F-G fuera
7	26.2 $\angle$ 71.1	Con el Generador 2 (Gen 2) fuera
8	26.1 $\angle$ 71.1	Con el Generador 3 (Gen 3) fuera
9	26.2 $\angle$ 71.1	Con el Generador 1 (Gen 1) fuera
10	26.1 $\angle$ 71.1	Sin la carga 1
11	26.2 $\angle$ 71.1	Sin la carga 2

Con estos resultados se observa que no existe cambio en la impedancia de falla aunque las condiciones y configuración del sistema hayan cambiado, entonces se concluye que no es necesario realizar cambio en el ajuste del relevador ubicado en el nodo A.

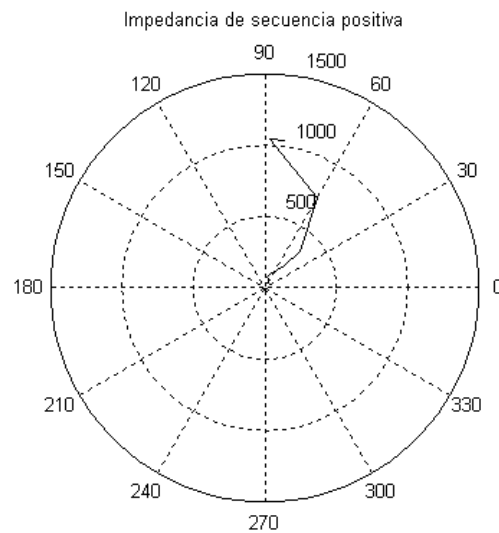


Figura 5.15. Gráfica de la impedancia de falla de la Tabla 5.7.

#### 5.4.4. Simulación de fallas en el sistema eléctrico real 3 (S.E.R.3).

Prueba S.E.R.3. Se realizaron fallas simuladas al 80% de la línea D-C, la cual es protegida en zona 2 por el relevador ubicado en el nodo A de la línea A-D del sistema eléctrico de la Figura 5.16. Se realizaron 7 fallas en esta línea tomando en cuenta distintas contingencias alrededor del nodo remoto del relevador que está siendo evaluado. Los resultados de estas fallas se muestran en la Tabla 5.8. y el comportamiento de la impedancia en el plano complejo en la Figura 5.17.

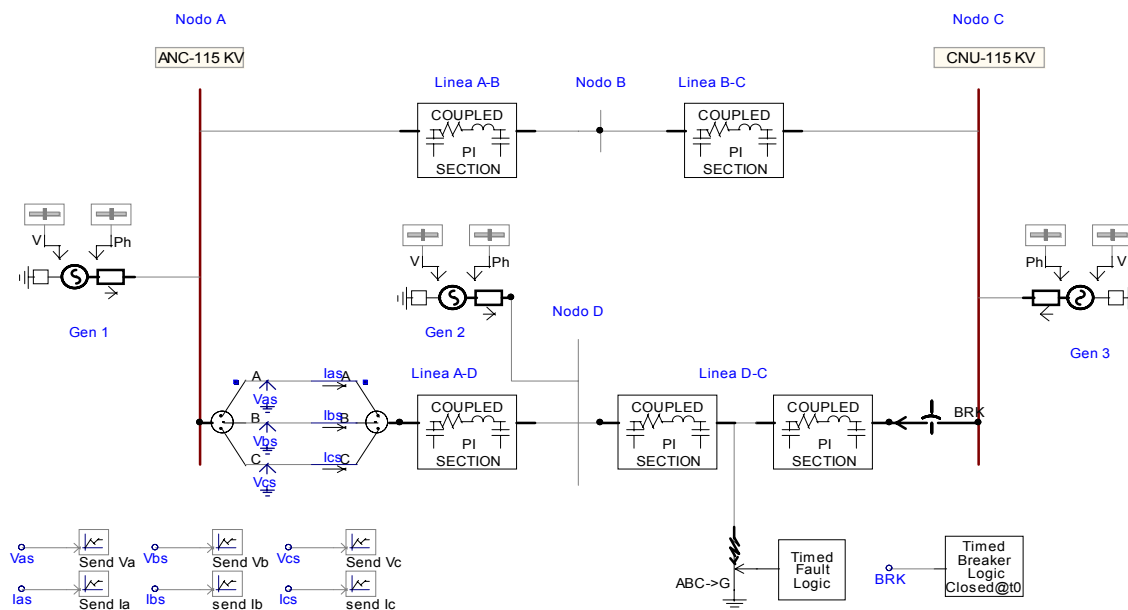


Figura 5.16. Sistema eléctrico real 3

Tabla 5.8. Impedancias de falla de la prueba S.E.R.3.

No. de Falla	Impedancia de Secuencia Positiva	Elementos en el sistema
1	$120 \angle 80.1$	Con todos los elementos
2	$66 \angle 79.5$	Sin el generador 2 (Gen 2)
3	$133 \angle 79.8$	Sin el generador 3 (Gen 3)
4	$415 \angle 84.4$	Sin el generador 1 (Gen 1)
5	$134 \angle 79.7$	Con la línea B-C fuera
6	$135 \angle 79.7$	Con la línea A-B fuera
7	$655 \angle 79.5$	Sin los generadores 2 y 3 (Gen2 y 3)

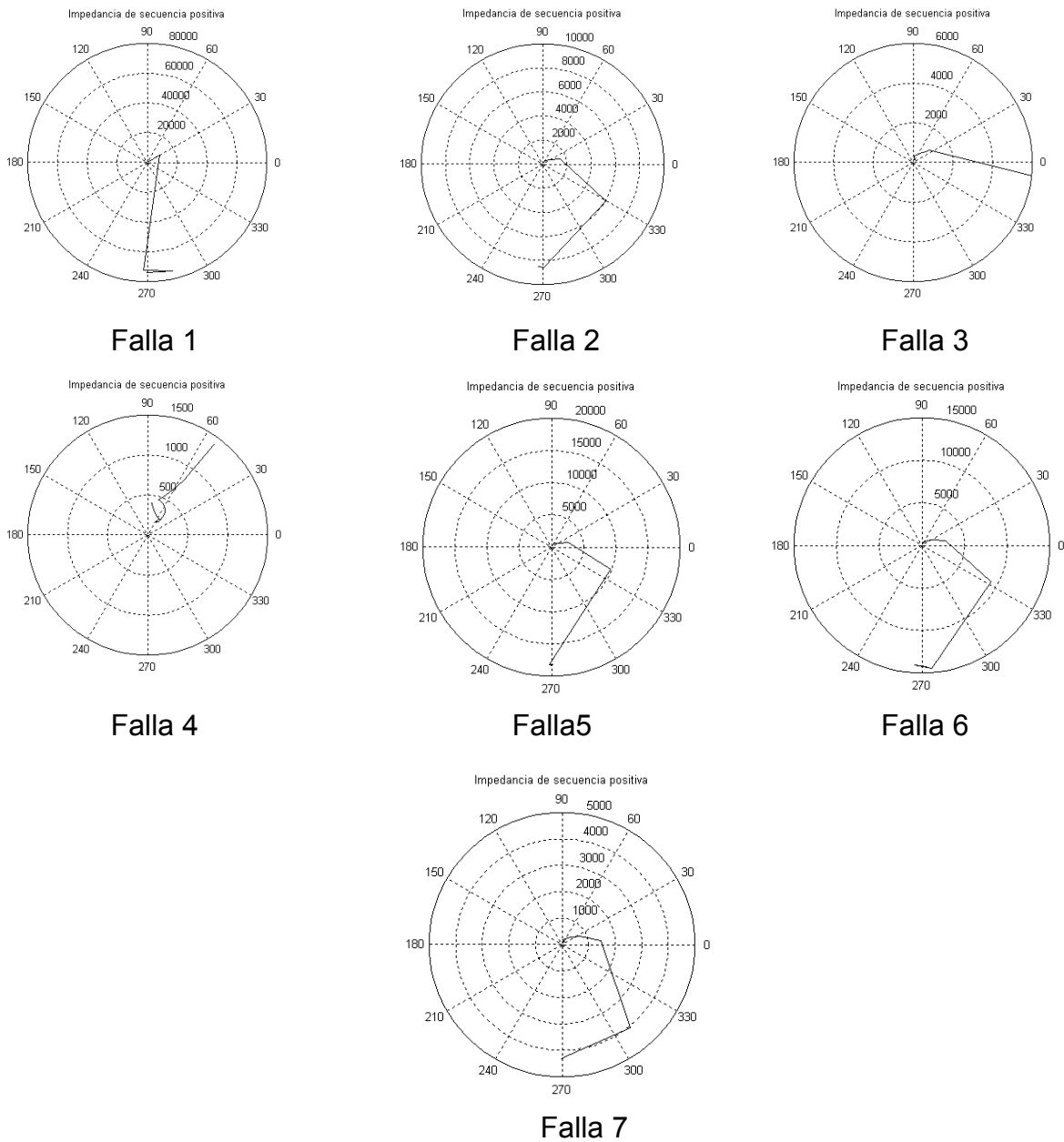


Figura 5.17. Gráficas de las impedancias de falla de la Tabla 5.8.

Para el caso de este sistema eléctrico de potencia, se tienen cambios considerables en la impedancia de falla. Entonces se puede sugerir la aplicación de técnicas de protección adaptable para lograr, de manera eficiente y segura, el mayor alcance posible de la zona 2 de los relevadores de distancia.

## CAPÍTULO VI

### **6. CONCLUSIONES Y RECOMENDACIONES PARA TRABAJOS FUTUROS.**

En este capítulo se presentan las conclusiones derivadas de este trabajo, así como algunas recomendaciones para posibles trabajos futuros, y las aportaciones que este trabajo ofrece.

#### **6.1. Conclusiones.**

De los resultados obtenidos puede concluirse que el filtro digital utilizado funciona correctamente, aún durante el transitorio debido a las fallas simuladas. Este algoritmo funciona correctamente para un ciclo completo de muestras a la frecuencia fundamental y se puede obtener cualquier componente de señales, siempre y cuando ésta haya sido modelada en el filtro de mínimos errores cuadrados. Los resultados muestran que solo es necesario un ciclo completo de falla para determinar si dicha falla ocurre dentro o fuera de la zona de protección del relevador.

Los resultados de esta implementación del algoritmo digital de protección de distancia en el DSP muestran que, en el peor de los casos, es necesario dos ciclos de frecuencia fundamental para determinar si ocurre una falla en la zona de operación programada en el DSP, esto se presenta cuando la falla ocurre a la mitad de ese ciclo (primer ciclo de muestreo del DSP), por lo tanto, hasta el siguiente ciclo, se tendrá la medición correcta de la impedancia de falla.

La modelación del sistema de potencia en el EMTDC/PSCAD es fundamental para poder simular las fallas y así probar al algoritmo, debido a que este software realiza las simulaciones en estado transitorio, y en estas condiciones, normalmente los algoritmos de filtros digitales pueden presentar problemas al calcular los fasores de las señales eléctricas.



El programa desarrollado en LabVIEW también es importante, debido a que facilita el manejo del puerto de comunicación serial de la PC, que es usado para transferencia de datos con el microcontrolador. Además, el programa fue utilizado para el manejo de controles en el panel frontal y para la presentación de gráficas, de tal manera que el diseño e implementación del algoritmo de protección de distancia en el DSP, represente un laboratorio virtual con el cual se puede obtener información del sistema de potencia.

Los resultados de la simulación de fallas muestran que los esquemas de protección adaptable tienen ventajas (mayor alcance de la protección) importantes comparados con los esquemas de protección convencionales. De estos resultados se deriva que el uso de esquemas adaptables sea una excelente alternativa para evitar problemas de coordinación, alcance o sobrealcance, en los sistemas de protección para líneas de transmisión. Enfatizando que no es posible generalizar que en todos los sistemas eléctricos sea indispensable la aplicación de esquemas de protección adaptable, porque depende de la configuración del sistema y de las condiciones de operación. Tal como se observa en los resultados, donde se muestra que no siempre son necesarios estos esquemas de protección.

La comunicación por el puerto serie existente entre el DSP y la computadora, es aceptable, pero es importante desarrollar un protocolo de comunicación eficiente, por ejemplo; DNP3 o algún otro, para evitar problemas de sincronía en el momento de transmitir una gran cantidad de datos.

## **6.2. Recomendaciones para trabajos futuros.**

Probar el algoritmo en un sistema de potencia simulado con el equipo RTP-DOBLE adquirido por el departamento de la SEPI.

Desarrollar un sistema de protección en el que se incluyan varios microcontroladores para extender el concepto de esquemas de protección adaptable y comprobar, a nivel de laboratorio, el desempeño de estos algoritmos de protección.

Desarrollar un protocolo de comunicación serial para mejorar la transferencia de datos entre el microcontrolador y la PC. Con esto se puede lograr enviar datos con valores flotantes para los ajustes en el relevador, y así se evitaría las limitaciones en cuanto al rango de envío de datos.

Implementar una tarjeta electrónica diseñada especialmente para esta aplicación incluyendo a los TC's y TP's secundarios. Con esta plataforma podrían desarrollarse nuevos algoritmos de protección como: de sobrecorriente, diferenciales, piloto o cualquier otro que se requiera.

Probar otros microcontroladores con capacidades de realizar operaciones en punto flotante, para evitar hacer conversiones que afectan la velocidad de procesamiento y por consecuencia no limitar a un ciclo completo de posfalla para determinar la impedancia de falla.

Probar este algoritmo con tarjetas de adquisición de datos para el manejo con LabVIEW y evaluar las ventajas de usar una PC con velocidades de procesamiento de más de 2.4 Ghz, y así evitar el uso del microcontrolador. Mediante esta plataforma, se pueden tener algunas ventajas. Por ejemplo, la PC cuenta con tarjeta de red, que puede ser usada para comunicación con el centro de control de protecciones, y LabVIEW tiene la capacidad de manejar archivos y programas por medio de este canal de comunicación.

Posiblemente un problema de esta implementación es el uso de una computadora para interactuar con el DSP, ya que ésta solo se utiliza precisamente para esto, y se podría considerar como un esclavo (desperdiciando la capacidad de la PC), pero con ella podría realizarse una comunicación por medio de una red, hacia el centro de control de protecciones, y así obtener un esquema de protección adaptable y jerarquizado que realice toda la coordinación del sistema de potencia.

### **6.3. Aportaciones de la tesis.**

Diseño e implementación de un algoritmo digital de protección para relevadores de distancia en el microcontrolador DSP56807.

Desarrollo de programas computacionales en MATLAB y LabVIEW para simular la operación de relevadores de distancia.

Coordinación entre el microcontrolador y la PC mediante un programa en LabVIEW para el intercambio de información por medio del puerto de comunicación serial.

Desarrollo de un programa en MATLAB para el cálculo de los coeficientes del filtro digital de errores mínimos cuadrados para 24 muestras, incluyendo en el modelo, señales de frecuencia fundamental, 2<sup>a</sup>, 3<sup>a</sup> y 4<sup>a</sup> armónica, CD y CD decreciente.

Cálculo e implementación del circuito eléctrico para adecuar las señales, de voltaje y corriente, para ser muestreadas por el ADC.

Desarrollo de subinstrumentos virtuales en LabVIEW, para la formación de fasores de voltajes y corrientes trifásicos con 24 muestras de cada ciclo completo de la señal de 60 Hz., aplicando el filtro digital de errores mínimos cuadrados. Así como del cálculo de la impedancia de secuencia positiva de estas señales.

**REFERENCIAS:**

- [1] C. Russell Mason, "El Arte y la Ciencia de la Protección por Relevadores". Editorial Continental S.A. de C.V. México D.F. Noviembre de 1982.
- [2] J. L. J. Lewis Blackburn, "Protective Relaying Principles and Applications." New York: Marcel Dekker, 1987.
- [3] D. Álvarez Hernández, Tesis de Grado de Maestría, "Protección Adaptable de Sobrecorriente en Sistemas de Transmisión." SEPI-ESIME-IPN, México D.F. Septiembre de 1998.
- [4] D. Sebastián Baltazar, Tesis de Grado de Maestría. "Coordinación Ajuste y Simulación de Protecciones de Distancia en Sistemas de Transmisión". SEPI-ESIME-IPN, México D.F. Septiembre de 1993.
- [5] D. Sebastián Baltazar, Tesis de Grado de Doctor. "Protección Adaptable de Sistemas de Transmisión: Ajuste y Coordinación de Relevadores Mediante Sistemas Expertos". SEPI-ESIME-IPN, México D.F. Noviembre de 1999.
- [6] S. H. Horowitz, A.G. Phadke, J.S. Thorp, "Adaptive Transmission System Relaying" IEEE Transactions on Power Delivery, Vol. 3, No 4, pp. 1436-1445, October 1988.
- [7] The MathWorks, Inc., Software "MATLAB" Versión 6.5, 2002.
- [8] M. Kesunovic, B. Kastenny, Z. Galijasevic. "Modeling, Developing and Testing Protective Relays Using MATLAB, Programmable Relays and Digital Simulators", 2000.
- [9] J. Schilleci, G. Breaux, M. Kezunovic, Z. Galijasevic, T. Popovic, "Use of Advanced Digital Simulators for Distance Relay Design and Application Testing", 2001.
- [10] K. Zimmerman, "Microprocessor-Based Distribution Relay Applications". Schweitzer Engineering Laboratories, Inc. 1996.
- [11] J. Mooney, "Microprocessor-Based Transmission Line Relay Applications". Schweitzer Engineering Laboratories, Inc. 1996.
- [12] L. Gross, "10 Ways to Reduce Project Costs by Using Microprocessor-Based Relays for Power System Protection". Schweitzer Engineering Laboratories, Inc. 1996.
- [13] M.S. Sachdev, T.S. Sidhu, B. Chattopadhyay, B. Talukdar, G. McDonald, R. Chan, "Design and Evaluation of an Adaptive Protection System for a Distribution Network", CIGRÉ Study Committee 34 Colloquium, Power System Protection and Local Control, Stockholm, Sweden, June 11-18, 1995.

- [14] J. H. Díaz, D. Sánchez, R. D. González. “Sistemas PCYM Integrados en Base a Relevadores Multifuncionales y Procesadores de Comunicación”. VII Simposio Iberoamericano sobre Protección de Sistemas Eléctricos de Potencia, presentada en Monterrey, Nuevo León, México. Noviembre de 2004.
- [15] M. D. Diehl, R. A. Crognale, J. A. Schwenk. “Microprocessor Relay Capabilities Improve Protection, SCADA, and Maintenance”. 1999.
- [16] G. Rosas Ortiz, “Detección y clasificación de fallas en sistemas de transmisión empleando una red neuronal con retropropagación del error”, Tesis de Maestría, SEPI-ESIME-IPN, México, 2000.
- [17] B. Reyes López, “Monitoreo y Protección Digital de un Generador de Tipo Experimental”, Tesis de Maestría, SEPI-ESIME-IPN, México, 1999.
- [18] R. A. Cortés Mateos, Tesis de grado de Doctor, “Control de excitación difuso de un generador síncrono”, SEPI-ESIME-IPN, México, 1997.
- [19] R. A. Cortés Mateos, Tesis de grado de Maestría, “Diseño de un sistema de control con microprocesador para una planta de emergencia generadora de energía eléctrica”, SEPI-ESIME-IPN, México, 1980.
- [20] D. Torres, Tesis de Licenciatura, “Prototipo de un Watorímetro Digital Basado en Microcontrolador”. ESIME-IPN, México, 2003
- [21] Maturano Rodríguez Juvenal, “Algoritmos Para la Protección del Transformador de Potencia de la Central Termoeléctrica de Tula Hidalgo en Base al Microcontrolador 8086”, Tesis de Maestría, SEPI-ESIME-IPN, México, 1993.
- [22] H. Becerra Zarraga, “Desarrollo de un Relevador de Multifunciones Basado en Microprocesadores para la Protección de un Generador.”
- [23] National Instruments, Software “LabVIEW 7 Express” Student Edition Versión 7.0, 2003.
- [24] U. Uribe Fernández, Tesis de Maestría, “Instrumentación Virtual de un Generador Síncrono de Laboratorio con LabVIEW”, SEPI-ESIME-IPN, México, 2003.
- [25] A. Pascasio de los Santos, “Prototipo para la detección de corto-circuitos entre laminaciones del núcleo del estator en generadores de potencia mediante corrientes de Eddy”, Tesis de Maestría, SEPI-ESIME-IPN, México, 2003.
- [26] Willibald Winkler, Andrzej Wiszniewski “Adaptive Protection–Potential and Limitations” CIGRÉ Study Committee 34 Colloquium, Power System Protection and Local Control, Stockholm, Sweden, June 11-18, 1995.
- [27] Unidad de Ingeniería Especializada de la Comisión Federal de Electricidad, “Protección de líneas de transmisión”. Curso de protecciones, 1990.

- [28] G.D. Rockefeller, C.L. Wagner, J.R. Linders “Adaptive Transmission Relaying Concepts for Improved Performance”
- [29] B. Stedall, P. Moore, A. Johns, J. Goody, M. Burt, “An Investigation into the Use of Adaptive Setting Techniques for Improved Distance Back-up Protection” IEEE Transactions on Power Delivery, Vol. 11, No. 2, pp. 757 – 762, April 1996.
- [30] A.K. Jampala , S.S. Venkata, M.J. Damborg. “Adaptive Transmission Protection Concepts and Computational Issues” IEEE Transaction on Power Delivery, Vol. 4 No. 1, pp. 177-185, January 1989.
- [31] G. Enriquez Harper, “Fundamentos de Protección de Sistemas Eléctricos por Relevadores”, México, Ed. Limusa, 1984.
- [32] SIEMENS, “Manual del relevador 7SL24 aplicado a sistemas de alta tensión.”
- [33] J.D. Codling, S.A. House, et. Al. “Adaptive Relaying: a New Direction in Power System Protection” IEEE Potentials, pp. 28-33, February – March – 1 April 1996.
- [34] Booth, J.R. McDonald, R.W. Stewart, W.J. Laycock, A. Bennett. “Coordinated Control and Protection: Functionality Enhancements Via Information Technology” IEE 368 Developments in Power System Protection, Fifth International Conference, The University of York, UK, pp. 25-28, 30 March- 1 April 1993.
- [35] T.S. Sidhu, D. S. Baltazar, R. M. Palomino, M. S. Sachdev. “A New Approach for Calculating Zone-2 Setting of Distance Relays and Its Use in an Adaptive Protection System”. IEEE Transaction on Power Delivery, Vol. 19, No. 1, pp. 70-77, Enero 2004.
- [36] Manitoba HVDC Research Centre inc., Software “PSCAD” Versión 4.0, 2004.
- [37] Manitoba HVDC Research Centre inc., “Manual de EMTDC y PSCAD”, 2004.
- [38] “DSP56F801/803/805/807 Manual del Usuario”, Procesador de Señales Digitales de 16 Bits (16-Bits Digital Signal Processor) Motorola Inc. 2001.
- [39] “Manual del DSP56800 16-Bit Digital Signal Processor”, Manual de la Familia Motorola DSP56800; DSP56800FM/D Rev. 1.00,01/2000.
- [40] Sachdev, M.S. and Baribeau, M.A., “A New Algorithm for Digital Impedance Relays”, IEEE Trans. On Power Apparatus and Systems, Vol. PAS-98, Nov., Dec. 1979, pp. 2232-2240.
- [41] A. Caballero, D. Sebastián, “Prototipo de protección adaptable para líneas de transmisión con relevadores de distancia microprocesados”, RVP-AI 2004, Acapulco Gro., México 2004.
- [42] Metrowerks Corporation, Software “CodeWarrior IDE” Versión 5.2, 2002.

- [43] D. S. Baltazar, R. M. Palomino. "Protección Adaptable de Sistemas de Transmisión: Alternativas para ajustar esquemas de protección de distancia".
- [44] Z. Zhizhe and C. Deshu, "An Adaptive Approach in Digital Distance Protection" IEEE Transaction on Power Delivery, Vol. 6, No. 1, pp. 135-142, January 1991.
- [45] Sachdev, T.S. Sidhu, B.K. Talukdar. "Topology Detection for Adaptive Protection of Distribution Networks". IEEE Energy Management and Power Delivery, Proceeding of EMPD'95, pp. 445-450, 1995.
- [46] IVESCO, S.A. de C.V. Tokio 102, 4º piso, col Juárez, Del. Cuauhtémoc CP. 06600, México, D.F. Website: <http://www.ivesco.com.mx>

## APÉNDICE A

### A. CARACTERÍSTICAS Y OPERACIÓN DE LOS MICROCONTROLADORES DSP56800 DE MOTOROLA.

Los procesadores de señales digitales han evolucionado casi todos los aspectos de nuestras vidas, estos ya se encuentran en todas partes, en cualquier equipo eléctrico se tiene uno o se ha utilizado uno para su diseño, ha mejorado la funcionalidad de los sistemas actuales y ha traído otros al mundo.

El DSP56800 es un Procesador de Señales Digitales que tiene la característica de realizar sumas de productos en un ciclo de máquina. El núcleo del DSP56800 tiene una Unidad Central de Procesamiento (CPU) como propósito general, diseñada para un eficiente procesamiento de señales digitales y una variedad de operaciones de control.

El DSP56800 tiene un Procesador de Señales Digitales programable de 16 BITS, este consiste de una Unidad de Aritmética y Lógica de 16 Bits (DATA ALU), una Unidad Generadora de Direcciones de 16 Bits (AGU), un Decodificador de Programas, On-Chip Emulation (ONCE), Buses y grupo de Instrucciones, como se puede observar en la figura A.1.

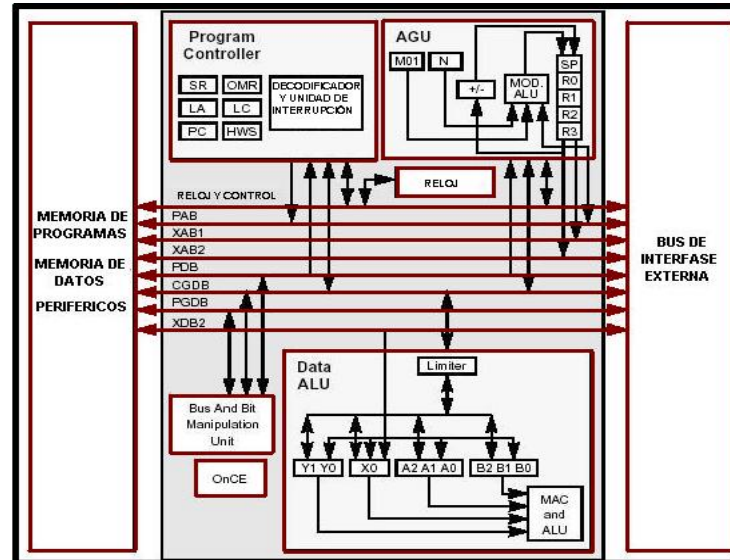


FIGURA A.1. DIAGRAMA DE BLOQUES DEL NÚCLEO DEL DSP56800.

La arquitectura del DSP56800 es de tipo Harvard, esto significa que tiene de manera separada la memoria de datos con la memoria de programas, indicado en la figura A.2.



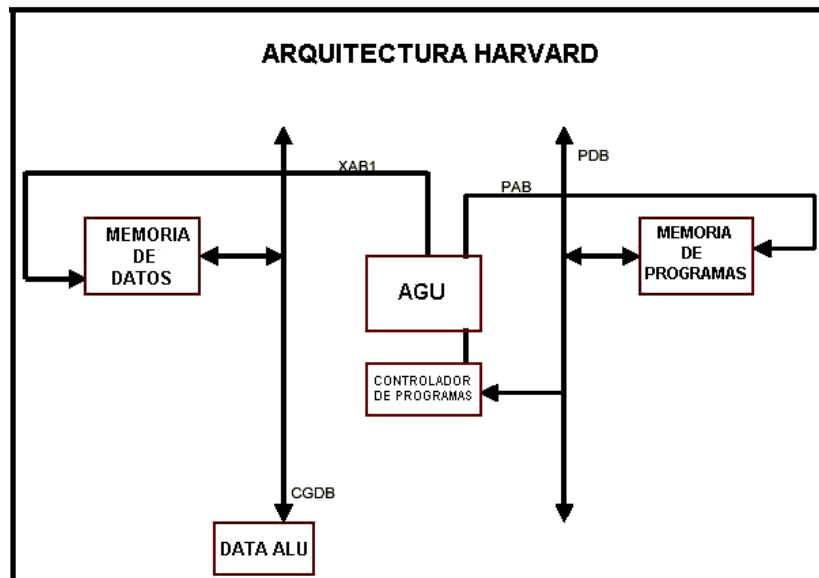


FIGURA A.2. ARQUITECTURA HARVARD.

Las principales características del DSP56800 son las siguientes:

- Capacidad de procesamiento de 35 millones de instrucciones por segundo (MIPS) a 70 MHz.
- Requiere solo 2.7-3.6 Volts.
- Paralelamente multiplica y acumula en un simple ciclo de instrucción 16 Bits x 16 Bits.
- Registros de corrimientos de simple instrucción de 16 Bits.
- Tiene dos acumuladores de 36 Bits que incluyen extensión de Bits.
- Grupo de instrucciones paralelas con único modo de direccionamiento del DSP.
- Modo de direccionamiento paralelo.
- Instrucciones de bucles mediante Hardware DO y REP.
- Requiere dos pines externos de interrupción.
- Tres buses de datos de 16 Bits.
- Tres buses de direccionamiento de 16 Bits.
- Opera con frecuencias bajas cercanas a CD.
- Grupo de instrucciones y funciones de control que soporta el DSP.
- Control estilo modo de direccionamiento e instrucciones para pequeños tamaños de código.
- Eficiente compilador en Lenguaje C y soporte de variables locales.
- Subrutinas de software e interruptor de apilamiento.
- Procesador de velocidad independiente para depuración.
- Simple repuesto de alimentación.

El núcleo del procesador del DSP56800 está diseñado como un procesador de la familia Motorola DSP's. El ejemplo de un chip que puede ser construido con este núcleo es mostrado en la figura A.3:

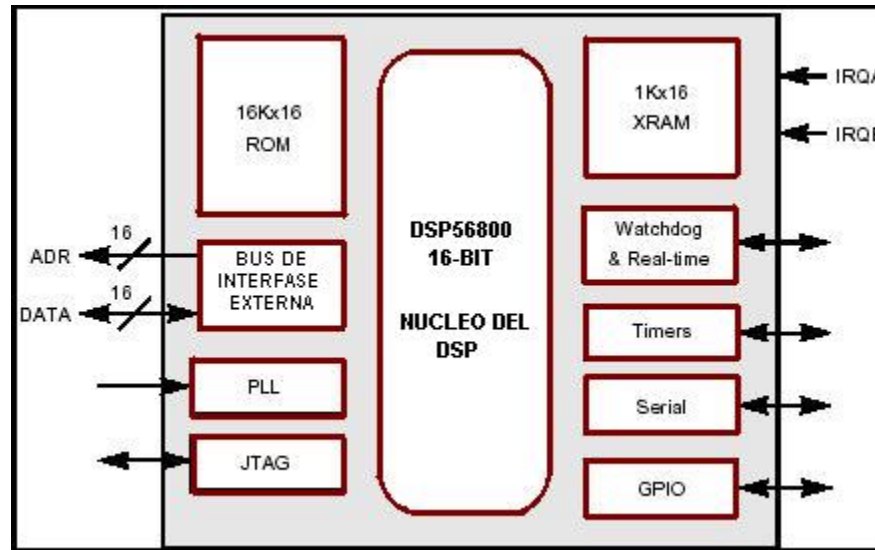


FIGURA A.3. EJEMPLO DE LA CONSTRUCCIÓN ALREDEDOR DEL NÚCLEO DEL CHIP DSP56800

El DSP tiene el procesamiento aritmético de señales en tiempo real mostradas en intervalos regulares y digitalizados, ejemplos del procesamiento de DSP incluyen lo siguiente:

- Filtración.
- Convolución (mezclando las señales).
- Correlación (comparando dos señales).
- Rectificación, amplificación y transformación.

La salida del DSP es procesada por un convertidor de señales digital / analógico (D/A) y es filtrada para remover los efectos de digitalización.

Algunos de los atributos del DSP son los siguientes:

- Realiza operaciones (MAC) multiplica y acumula.
- Realiza ciclos de lectura de dos operaciones para el MAC.
- Realiza control de programas que provee versátiles operaciones.
- Movimientos de datos en entrada/salida fuera del DSP.

La operación MAC es la operación fundamental usada en el DSP. El DSP tiene una Arquitectura tipo Harvard optimizada para operaciones MAC como se muestra en la figura A.4.

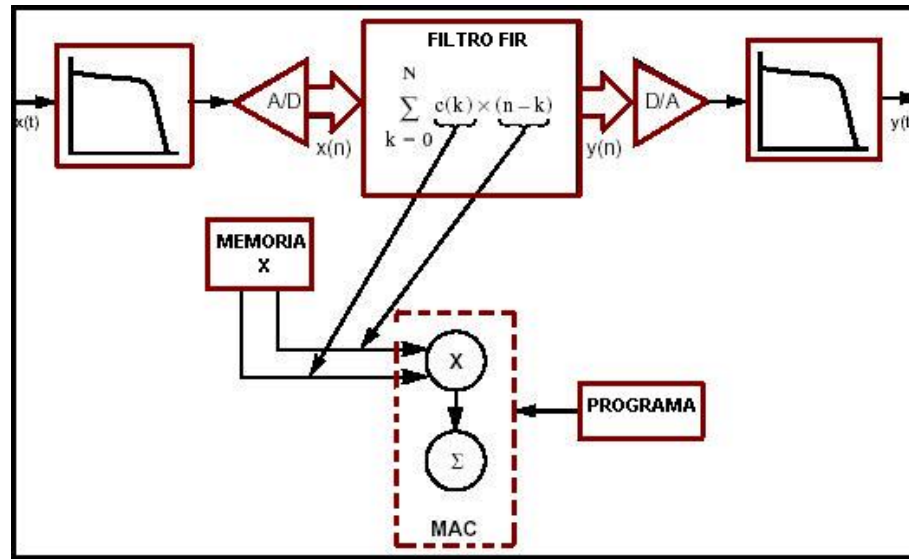


FIGURA A.4. ALGORITMOS MAPEADOS DENTRO DEL HARDWARE DEL DSP.

Los dos operandos  $C()$  y  $X()$ , realizan la operación directa a multiplicar y el resultado es sumado, además puesto que la memoria y la unidad de MAC son independientes, el DSP puede ejecutar dos movimientos de memoria, una multiplicación y una acumulación y dos actualizaciones de direccionamiento en una simple operación.

El núcleo del DSP56800 está compuesto por unidades funcionales que operan paralelamente para aumentar la eficiencia de la máquina. El Program Controller, la AGU y la Data ALU, cada unidad contiene sus propios grupos de registros y controles lógicos, de esta manera cada unidad puede operar independientemente y en forma paralela con las otras unidades.

Los componentes del núcleo del DSP56800 son los siguientes:

- Unidad de Aritmética y Lógica de Datos ( DATA ALU).
- Unidad Generadora de Direcciones (AGU).
- Controlador de Programas (PROGRAM CONTROLLER).
- Unidad de Bus y Unidad de Manipulación de BIT (BUS AND BIT MANIPULATION UNIT).
- On Chip Emulation (ONCE).
- Buses de Direccionamiento ( ADDRESS BUSES).
- Buses de Datos (DATA BUSES).

Estos componentes se muestran en la figura A.5.

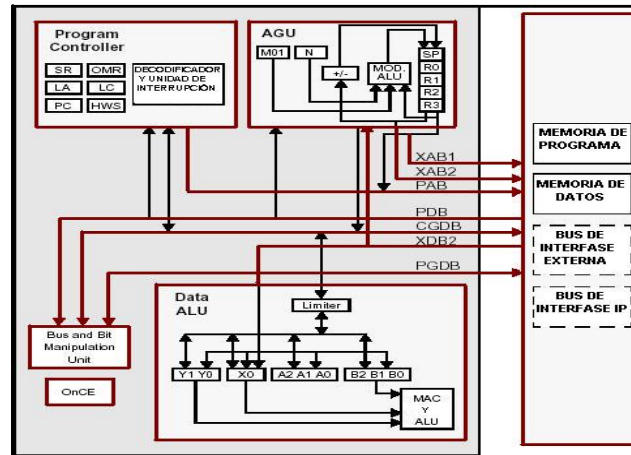


FIGURA A.5. COMPLEMENTO DEL DIAGRAMA DE BLOQUES DEL NÚCLEO DEL DSP56800.

La Data ALU se observa en la figura A.6. y realiza todas las operaciones Aritméticas y Lógicas de datos. Esta unidad consiste de lo siguiente:

- Tres Registros de Entrada de 16 Bits ( $X_0$ ,  $Y_0$ ,  $Y_1$ ).
- Dos Acumuladores de 32 Bits (A, B).
- Dos Registros de Extensión de Bits de los Acumuladores de 4 Bits (A2, B2).
- Un Acumulador de Corrimiento (AS).
- Un Limitador de Datos.
- Un Registro de Corrimiento de 16 Bits.
- Una Unidad que Multiplica y Acumula (MAC).

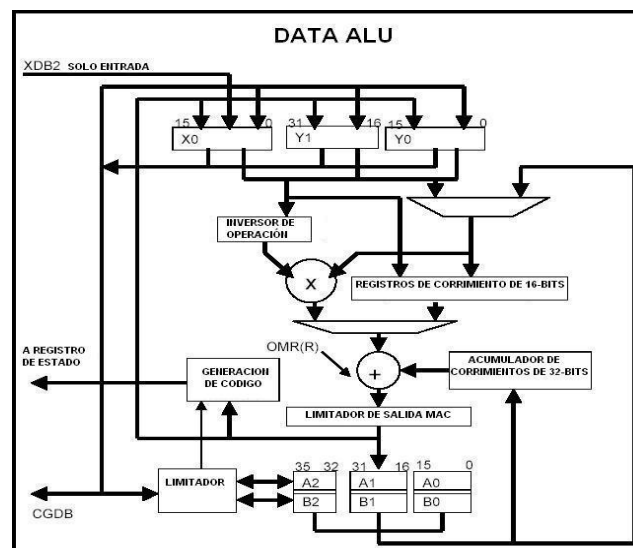


FIGURA A.6. UNIDAD ARITMÉTICA LÓGICA.

La Data ALU tiene la capacidad de multiplicar, multiplicar y acumular (con acumulación positiva o negativa), suma, resta, corrimientos y operaciones lógicas en un ciclo de instrucción.

Los operandos fuentes de la Data ALU pueden ser de 16, 32 o 36 Bits y se pueden originar individualmente desde los registros de entrada, locaciones de memoria, datos inmediatos. Los resultados son almacenados en uno de los acumuladores. En una suma algunas instrucciones aritméticas almacenan sus resultados de 16Bits en uno de los tres registros de entrada de la Data ALU o directamente en la memoria. Las operaciones aritméticas y los corrimientos pueden tener resultados de 16 Bits o de 36 Bits.

Los valores de los registros de la Data ALU pueden ser transferidos (lectura o escritura) a través del Bus de Datos de Núcleo Global (CGDB), como operaciones de 16 Bits, los valores del registro X0 puede ser escrito por un Bus externo de datos (XDB2) como operaciones de 16 Bits.

La AGU se observa en la figura A.7. y realiza todos los cálculos efectivos de direccionamiento y almacena direcciones necesarias para operandos de direccionamiento de datos en la memoria. La AGU funciona en paralelo con otras fuentes del chip.

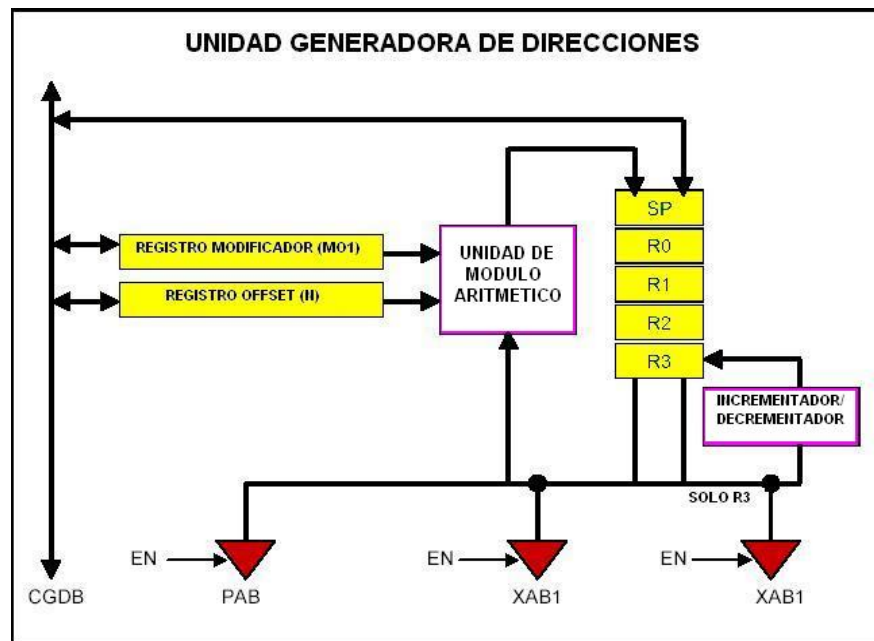


FIGURA A.7. UNIDAD GENERADORA DE DIRECCIONES.

Tiene dos Buses de Direccionamiento de 16 Bits (XAB1) y (XAB2) y un Bus de Direccionamiento de Programas (PAB). La AGU puede directamente direccionar

65,536 locaciones en el XAB1 o XAB2 y 65,536 locaciones en el PAB, un total de 131,072 datos de palabras de 16 Bits.

La AGU contiene los siguientes registros:

- Cuatro registros de direccionamiento (R0, R1, R2, R3).
- Un registro STACK POINTER (SP).
- Un registro OFFSET (N).
- Un registro modificador (MO1).
- Una unidad de modulo aritmético.
- Una unidad de incremento / decremento.

Los registros de direccionamiento son registros de 16 Bits estos pueden contener un dato o una dirección.

El controlador de programas se presenta en la figura A.8. ejecuta las siguientes instrucciones:

Instrucción de pre-ciclo de lectura (PREFECTCH).

- Instrucción de decodificación.
- Control de Loop por Hardware.
- Interrupción de procesamiento.

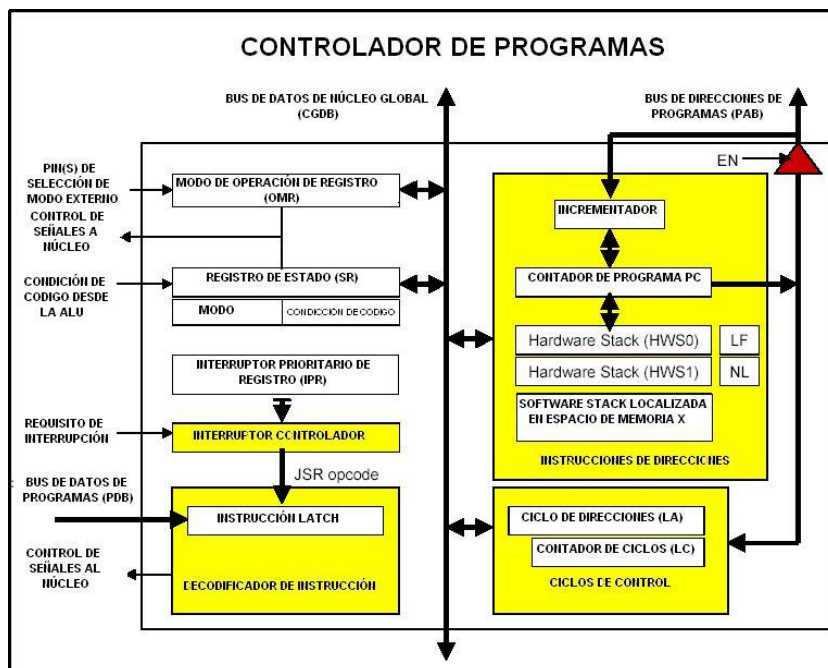


FIGURA A.8. CONTROLADOR DE PROGRAMAS.

La ejecución de las instrucciones es conducida afuera en otras unidades como la Unidad de Aritmética y Lógica de Datos, Unidad Generadora de Direcciones, Unidad de Manipulación de Bit presentadas en la figura A.9.

El controlador de programas consiste de las siguientes partes:

- Una unidad de PROGRAM COUNTER.
- Instrucciones LATCH (atrapa) y decodifica.
- Control lógico de Looping de Hardware.
- Interruptor de control lógico.
- Registros de estado y de control.

Las localizaciones dentro del controlador de programas son los siguientes:

- Cuatro registros de uso accesible.
- Loop Address Register (LA)
- Loop Count Register (LC)
- Status Register (SR)
- Operación de modo Register (OMR)
- Un Program Counter (PC).
- Un Hardware Stack (HWS).

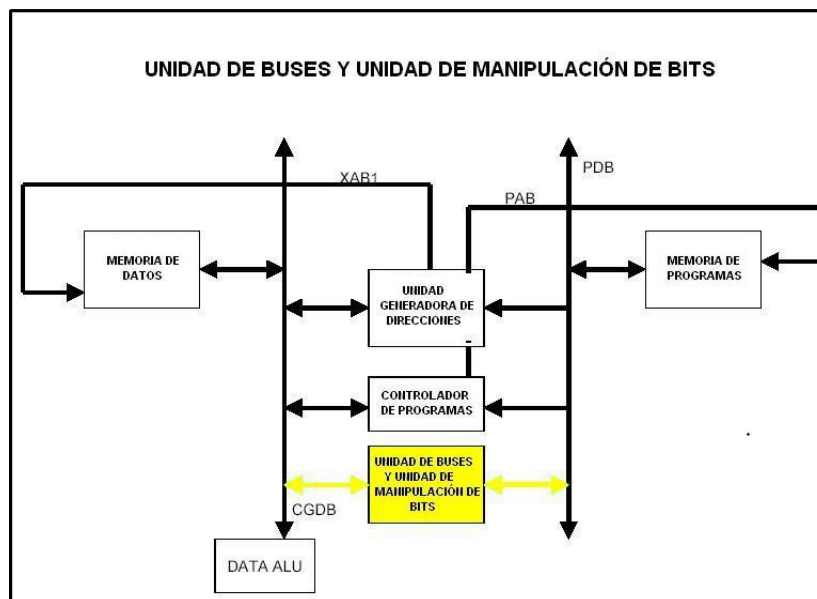


FIGURA A.9. UNIDAD DE BUSES Y MANIPULACIÓN DE BITS.

Las transferencias entre los buses internos son terminadas en la Unidad de Bus. La unidad de Bus es similar como un SWITCH y puede conectar cualquiera de los tres buses internos de datos junto con otros introduciendo retardos. Estos datos pueden ser movidos desde la memoria de datos o de programa. La unidad de Bus es usada para transferir datos por el PGDB.

La Unidad de Manipulación de Bit ejecuta manipulaciones de Bits de campo en la memoria de palabras de datos X, registros periféricos y con todos los registros dentro del núcleo del DSP56800.

La unidad ONCE permite al usuario la interacción en un momento de depuración con el núcleo del DSP56800 y se observa en la figura A.10.

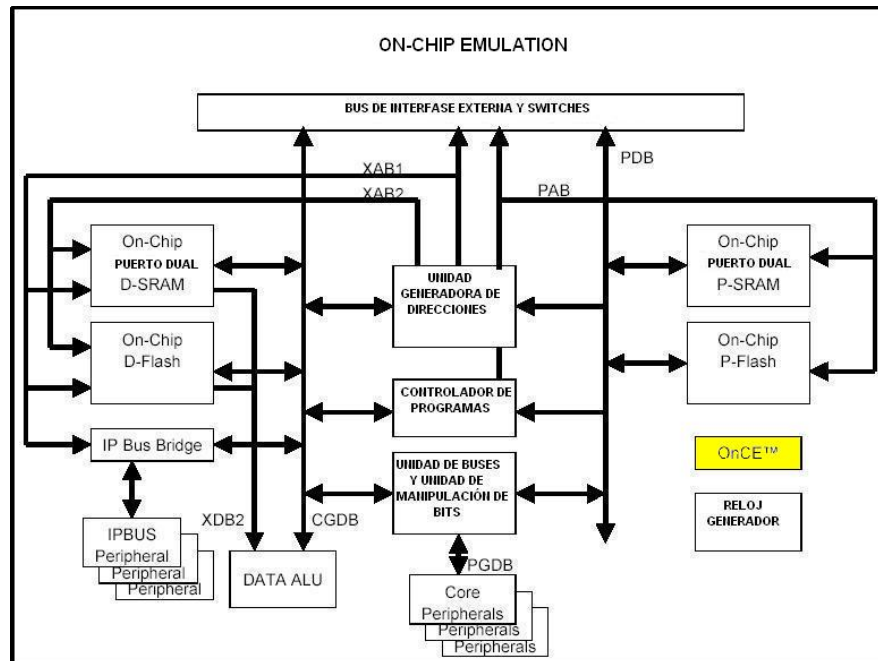


FIGURA A.10. UNIDAD ONCE.



## APENDICE B

### B. METODOLOGÍA PARA LA PROGRAMACIÓN DEL DSP56800 CON EL SOFTWARE “CODEWARRIOR”

¿Como se instala el programa?

Al introducir el CD. De instalación aparece la pantalla principal mostrada en la figura B.1.



Figura B.1. Menú del CD de instalación

Del menú que presenta se elije “Launch CodeWarrior Setup”, después de la elección, aparece la pantalla de bienvenida en la figura B.2.

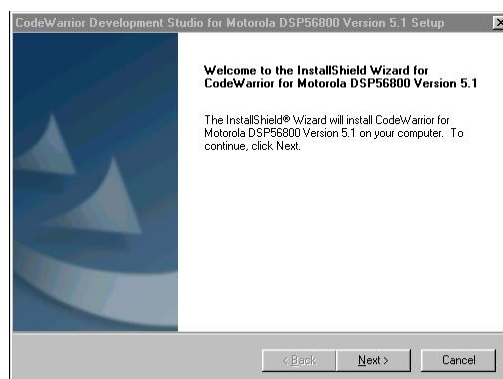


Figura B.2. Bienvenida para la instalación.

Después aparece la pantalla de licencia, para leer los términos y condiciones del programa, en la figura B.3.



Figura B.3. Licencia, términos y condiciones.

Posteriormente tenemos algunas descripciones del programa CodeWarrior 5.1., tal como aparece en la figura B.4.

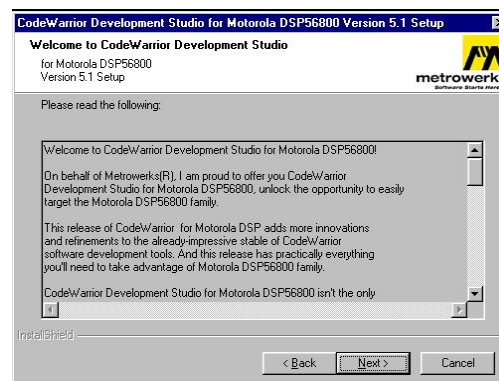


Figura B.4. Descripciones.

Ahora elegimos el tipo de instalación, completa o la opción de elegir componentes, esto se muestra en la figura B.5., se recomienda la instalación completa a menos que ya se haya manejado antes este programa y no se deseen algunos componentes.

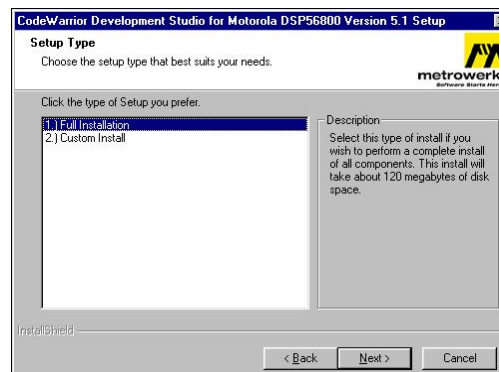


Figura B.5. Tipo de instalación.

Después en la figura B.6, se realiza la selección del folder donde se colocarán los iconos de acceso al programa

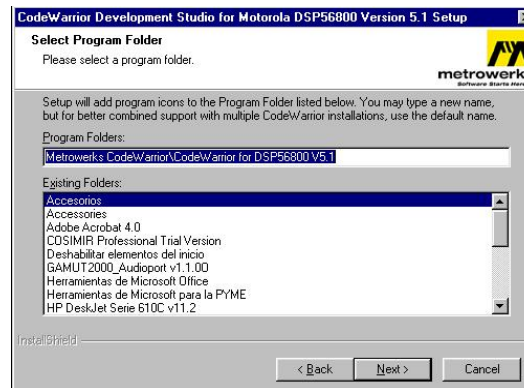


Figura B.6. Elección del folder.

La selección de archivos asociados, es decir, archivos que pueden funcionar con CodeWarrior 5.1., se ven en la figura B.7.

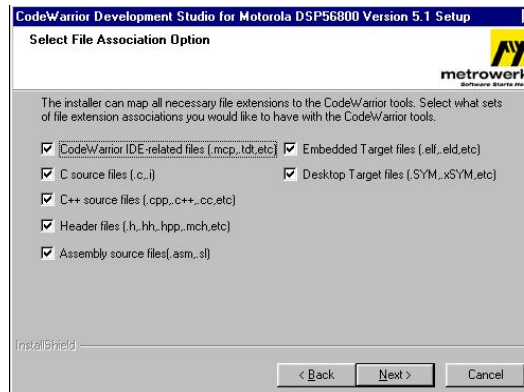


Figura B.7. Archivos asociados.

En las figuras B.8 y B.9 se presenta el inicio de la copia de archivos al disco duro de la computadora, y el proceso de copia de dichos archivos.

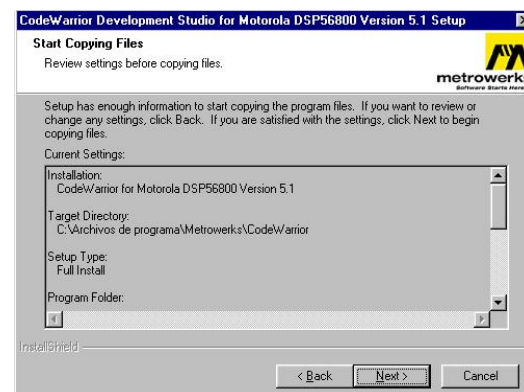


Figura B.8. Preparado para copiar archivos.

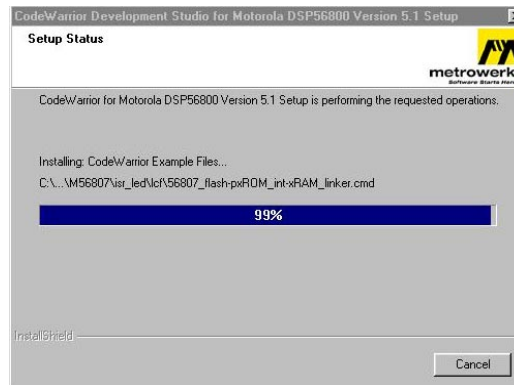


Figura B.9. Progreso de la copia de archivos.

Al término de la copia de archivos, figuras B.10 y B.11, la instalación pregunta si se desea registrar el programa en la página de Metrowerks, para esto es necesario estar en línea (conectado a internet) y que el software no haya sido registrado antes.



Figura B.10. Pregunta de registro.

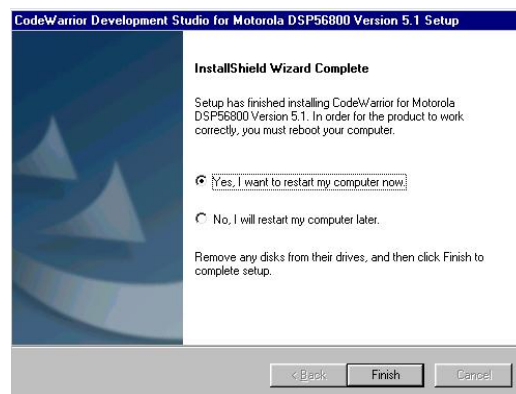


Figura B.11. Fin de la instalación.

Para que el programa pueda ejecutarse es necesario tener la clave o “key” de acceso, e insertarse en el archivo de texto license.dat creado en el fólдер del programa (c:\Archivos de programa\Metrowerks\CodeWarrior\license.dat por ejemplo).

Aquí se presenta un ejemplo de la clave para ejecutar el programa:

```
#FEATURE Win32_CWIDE_Limited metrowks 5.0 permanent uncouned
25DCD0C24654 HOSTID=ANY
#####
# ( 356): DSP V5.1 8k limited
#####
FEATURE Win32_Plugins_DSP56800_Demo metrowks 5.1 permanent uncouned
DA3EF17C54C0 VENDOR_STRING="DSP plugins 8k limited" HOSTID=ANY ck=142
```

Después de realizar esto, se está listo para crear y ejecutar un programa para el DSP 56800 de motorola.

## B.2. Realización de un programa para el DSP56807.

El primer paso es ir al menú INICIO de windows, elegir Todos los programas, después, Metrowerks CodeWarrior, CodeWarrior for DSP56800 V5.1 y CodeWarrior IDE. Una vez abierto el programa elegimos New en el menú File de la barra de herramientas.

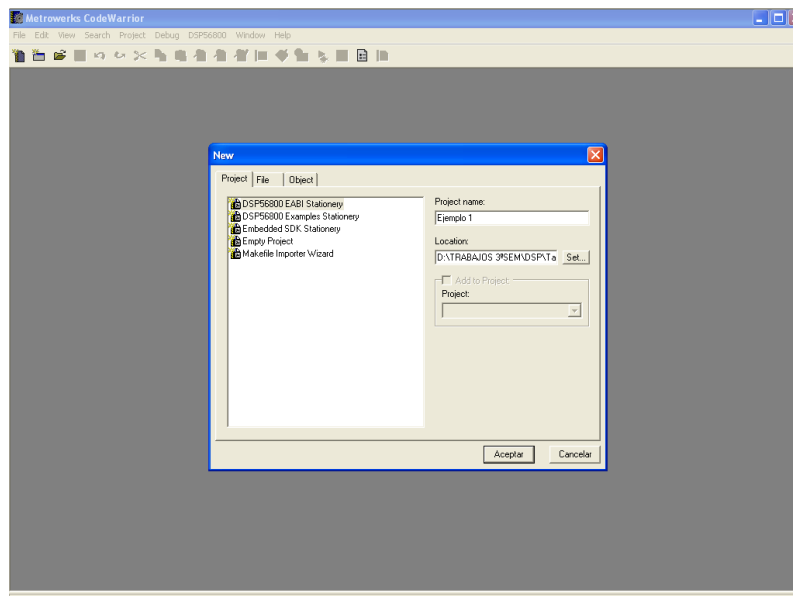


Figura B.12. Menú para crear un programa

En la Figura B.12 podemos elegir el tipo de proyecto que deseamos crear, los más utilizados dependiendo de la aplicación son DSP56800 EABI stationery y Embedded SDK stationery, el primero se conoce como modo estacionario, porque crea archivos básicos o mínimos para la ejecución y carga en el microcontrolador DSP, el segundo crea una serie de archivos para aplicación con SDK que es un software auxiliar donde se encuentran definidas muchas funciones de varios tipos: para multiplicaciones, sumas, restas, divisiones, funciones trigonométricas inversas, etc,

con las características de no presentar desbordamientos y ser mas eficientes para los cálculos. El único inconveniente de el software SDK es que se tiene que pedir por e-mail la licencia o key de instalación.

Para crear el proyecto únicamente tenemos que escribir el nombre que se desea dar y elegir la dirección o la carpeta donde queremos guardarlo.

En este ejemplo escribiremos el nombre, Ejemplo 1, en modo estacionario (DSP56800 AEBI stationery).

Después de esto, tenemos la pantalla mostrada en la figura B.13, que nos permite elegir que tipo de microcontrolador DSP vamos a utilizar, ó si se desea usar el simulador.

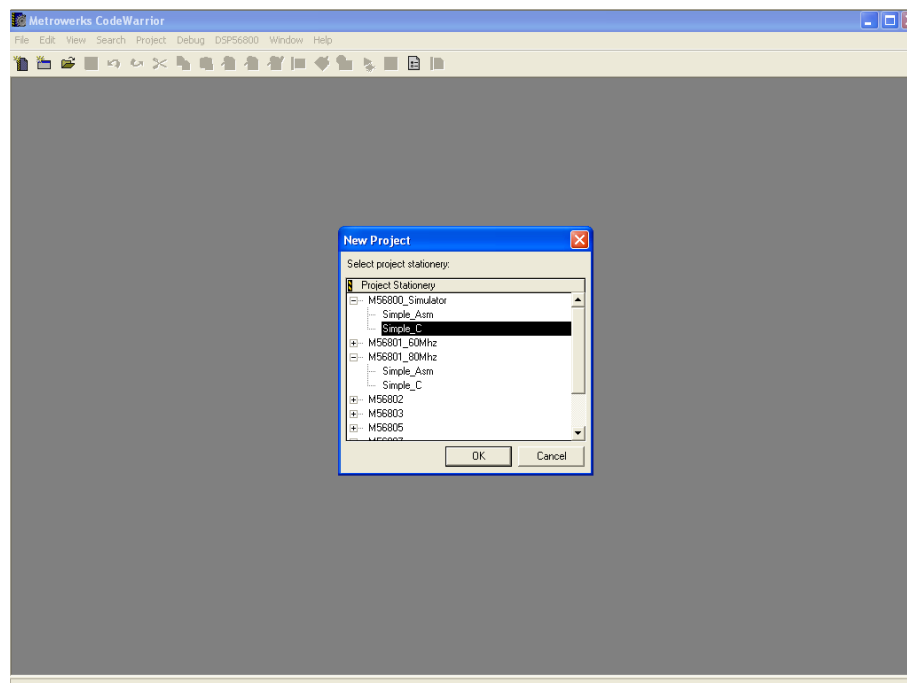


Figura B.13. Menú para la elección del Microcontrolador DSP

Como podemos observar en la figura B.13, cada uno de los microcontroladores a elegir tiene dos modos de operación.

1. Modo Simple\_Asm para la creación de programas con ensamblador únicamente.
2. Modo Simple\_C para la creación de programas en lenguaje C o mezclado con ensamblador.

Al dar aceptar se crea el proyecto y la pantalla es mostrada en la figura B.14.:

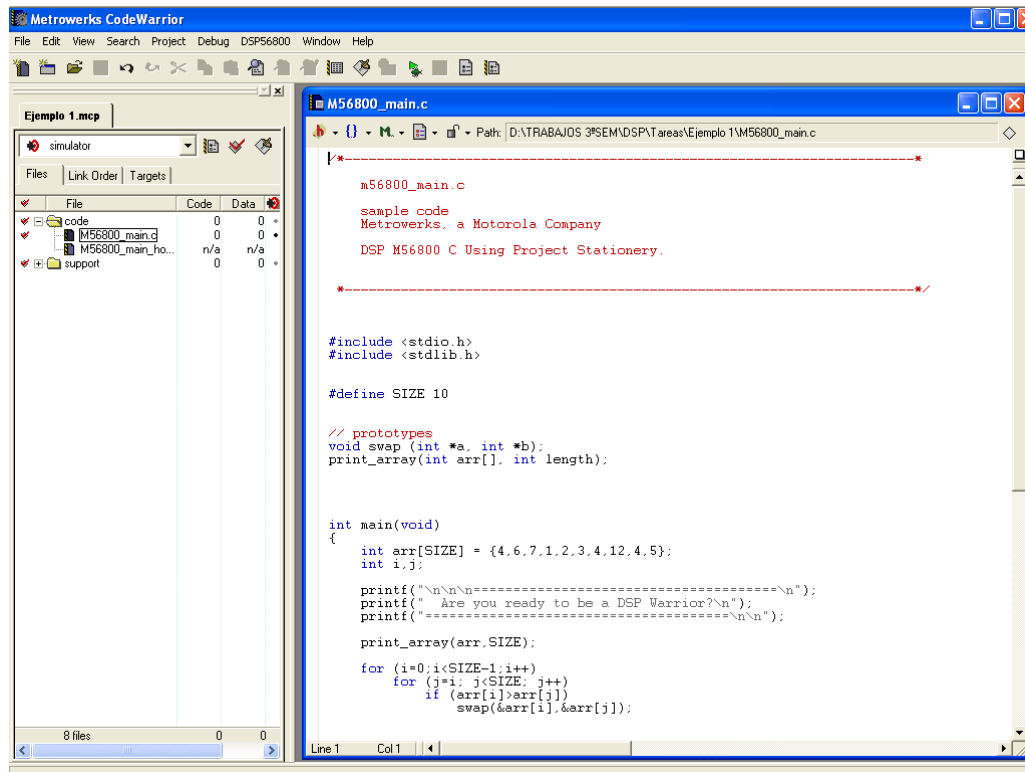


Figura B.14. Proyecto creado en modo estacionario en lenguaje C.

Automáticamente se crean las carpetas con los archivos necesarios para la descarga al micro, el programa en la Figura B.14 es también creado automáticamente, ahora podemos modificar el programa para la aplicación que se requiera.

Algunos inconvenientes del C de CodeWarrior es que algunas funciones de C++ no están disponibles, como por ejemplo scanf, funciones trigonométricas. Los botones para compilar (compile), hacer (make) y suprimir errores (debug) son los mostrados en la Figura B.15, los cuales se usan para enviar el archivo a la memoria del microcontrolador.



Figura B.15. Botones de compilación, hacer y debug

El botón "compile", compila el programa y muestra si se tienen errores de compilación. El botón "Make", crea los archivos que se van a mandar al micro. El botón "Debug", hace la transferencia de los archivos por el puerto paralelo hacia el micro, con la característica de que se puede oprimir este botón únicamente puesto que al mismo tiempo, compila, crea archivos y los manda al micro.

Estos son algunos procedimientos básicos para poder cargar y ejecutar un programa en el DSP 56800 de Motorola.



## APÉNDICE C

### C. PROGRAMA DIGITAL PARA RELEVADORES IMPLEMENTADO EN EL DSP.

/\*-----\*/

**Algoritmo para Relevador de Distancia.**

**Captura de los ADC\_A0,A1,A2,A4,A5,A6 por pulso de sincronía del TimerC2 con un muestreo de 24 muestras por ciclo a 60Hz. Cálculo fasorial e impedancia de sec. posi.**

**Ing. Adrián Caballero Rodríguez.**

**DSP M56807 C Using Project Stationery.**

\*-----\*/

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "56807_pll.h"
#include "tfr16.h"
#include "mfr32.h"
#include <cmath>
typedef unsigned short WORD;
#define ADC_A_STAT          0x1286
#define TIMERC2_SCR        0x1157 //Registro de control de estado para el Timer C2
#define TIMERC2_CTRL      0x1156 //Registro de control para el Timer C2
#define ADC_A_CTRL        0x128.0 //Registro de control para el ADC A
#define ipr                (WORD *) 0xFFFFB // Interrupt priority register
#define ADCA_ADCR1        (WORD *) 0x1280
#define ADCA_ADSDIS      (WORD *) 0x1285
#define ADCA_ADRSLT0     (WORD *) 0x1289 //ADC Result Register 0
#define ADCA_OFFSET0     (WORD *) 0x12A1 //ADC offset Register 0
#define ADCA_ADRSLT4     (WORD *) 0x128D //ADC Result Register 4
#define ADCA_OFFSET4     (WORD *) 0x12A5 //ADC offset Register 4
#define ADCA_ADRSLT1     (WORD *) 0x128A //ADC Result Register 1
#define ADCA_OFFSET1     (WORD *) 0x12A2 //ADC offset Register 1
#define ADCA_ADRSLT5     (WORD *) 0x128E //ADC Result Register 5
#define ADCA_OFFSET5     (WORD *) 0x12A6 //ADC offset Register 5
#define ADCA_ADRSLT2     (WORD *) 0x128B //ADC Result Register 2
#define ADCA_OFFSET2     (WORD *) 0x12A3 //ADC offset Register 2
#define ADCA_ADRSLT6     (WORD *) 0x128F //ADC Result Register 6
#define ADCA_OFFSET6     (WORD *) 0x12A7 //ADC offset Register 6
#define ITCN_BASE        (WORD *) 0x1260 //0x0E60 Base address for ITCN's GPR registers
#define TmrC2_CNTRL      (WORD *) 0x1156 //Registro de control para el Timer C2
#define TmrC2_CMP1       (WORD *) 0x1150 //Registro de comparacion para el Timer C2
#define TmrC2_LOAD       (WORD *) 0x1153 //Registro de carga para el Timer C2
#define TmrC2_CNTR      (WORD *) 0x1155
#define SCIO_CTRL        0x1301 // SCIO control register
#define SCIO_SR          0x1302 // SCIO status register
#define SCIOBR           (WORD*) 0x1300 // Velocidad de transimision
#define SCIOCR           (WORD*) 0x1301 // Registro de control del SCIO
#define SCIODR           (WORD*) 0x1303 // Registro de dato
#define GPR12            (WORD*) 0x126C // Prioridad del registro GPR12
#define GPR13            (WORD*) 0x126D // Prioridad del registro GPR13
#define PMOUT            0x1203
#define pwma_PWMVAL1    0x1207
#define GPIO_B_DR        0x13C1
#define GPIO_B_DDR       0x13C2
#define GPIO_B_PER       0x13C3
#define muestras        144
// prototypes
void init(void);
adca_setup(void);
timer_setup(void);
void irqISR(void);
```





```
void adcaISR(void);
void delay(void);
sci0_setup(void);
void sci0_tdre(void);
void sci0_rdrf(void);
void opcion(void);
void initLED();
void ul2str(unsigned long val, char * buff);
char dest[8];
char mensaje;
//Variables globales
int siempre=1,cont=0,n,k,m,aux=0,hace,nuevo = 1;
int pimag=17,lee,escribe;
double preal=1,preal1;
int conta=0,imprime=0,envio=0,scan=0,elige;
//Variables globales para VA
int VAmuestras[muestras];
int VAresult;
long VAreal,VAimag;
double VAmag,VAang;
double VArealp,VAimagp;
//Variables globales para IA
int IAmuestras[muestras];
int IAresult;
long IAreal,IAimag;
double IAmag,IAang;
double IArealp,IAimagp;
//Variables globales para VB
int VBmuestras[muestras];
int VBresult;
long VBreal,VBimag;
double VBmag,VBang;
double VBrealp,VBimagp;
//Variables globales para IB
int IBmuestras[muestras];
int IBresult;
long IBreal,IBimag;
double IBmag,IBang;
double IBrealp,IBimagp;
//Variables globales para VC
int VCMuestras[muestras];
int VCresult;
long VCreal,VCimag;
double VCmag,VCang;
double VCrealp,VCimagp;
//Variables globales para IC
int ICMuestras[muestras];
int ICresult,Datos[muestras];
long ICreal,ICimag;
double ICmag,ICang;
double ICrealp,ICimagp;
//Variables globales para impedancias
double ZA,angZA,ZB,angZB,ZC,angZC,I0real,I0imag,re,im,falla;
double Mreal=2.2311,Mimag=-0.8002,I0xreal,I0ximag,ajuste;
double IAcompreal,IAcompimag,IBcompreal,IBcompimag,ICcompreal,ICcompimag;
//Variables globales para el filtro LES.
long se[24] = {FRAC16(0.4171),FRAC16(0.0019),FRAC16(-0.1701),FRAC16(-0.1318),FRAC16(-0.0338),
FRAC16(-0.0072), FRAC16(-0.0656),FRAC16(-0.1275),FRAC16(-0.1192),FRAC16(-0.0504),FRAC16(0.0076),
FRAC16(0.0062),FRAC16(-0.0278),FRAC16(-0.0277),FRAC16(0.0331),FRAC16(0.1060),FRAC16(0.1192),
FRAC16(0.0628),FRAC16(0.0100),FRAC16(0.0421),FRAC16(0.1450),FRAC16(0.1874),FRAC16(0.0182),
FRAC16(-0.3956)};
long co[24] = {FRAC16(-0.0284),FRAC16(-0.0774),FRAC16(-0.0891),FRAC16(-0.0685),FRAC16(-0.0366),
```



FRAC16(-0.0119),FRAC16(0.0023),FRAC16(0.0154),FRAC16(0.0355),FRAC16(0.0601),FRAC16(0.0787),  
FRAC16(0.0842),FRAC16(0.0797),FRAC16(0.0740),FRAC16(0.0710),FRAC16(0.0651),FRAC16(0.0479),  
FRAC16(0.0192),FRAC16(-0.0097),FRAC16(-0.0266),FRAC16(-0.0321),FRAC16(-0.0420),FRAC16(-0.0753),  
FRAC16(-0.1354);

/\*/ **MUESTRAS DE UNA FALLA MONOFÁSICA AL FINAL DE UNA LÍNEA DE 100 KM //**

int muvola[144] = {20813,24374,27390,29657,31018,31383,30726,29091,26591,23395,19721,15820,11957, 8396, 5380,  
3113, 1750, 1385, 2041, 3675, 6175, 9370,13044,16945,20808,24369,27386,29654,31017,31384,30728,29095,26595,  
23400,19727,15826,11963, 8401, 5383, 3115, 1751, 1384, 2039, 3672, 6170, 9365,13038,16939,20802,23849,26448,  
28416,29576,29829,29154,27604,25289,22370,19045,15541,12095, 8942, 6296, 4338, 3200, 2961, 3636, 5180, 7486,  
10398,13718,17218,20661,23811,26454,28410,29546,29783,29107,27562,25254,22341,19020,15518,12074, 8922, 6277,  
4320, 3183, 2943, 3619, 5162, 7469,10382,13702,17203,20647,23799,26444,28402,29540,29779,29105,27562,25256,  
22343,19024,15522,12078, 8926, 6281, 4322, 3184, 2944, 3618, 5161, 7466,10379,13698,17200,20644,23797,26443,28402,  
29542,29783,29109,27568,25263,22351,19032,15531,12087, 8934, 6288, 4328, 3188, 2947, 3620, 5161, 7466,10377,  
13697,17198};

int mucorA[144] =

{16627,16851,17043,17191,17283,17314,17282,17188,17040,16847,16623,16382,16141,15917,15725,15577,15485,15454,  
15486,15579,15728,15921,16145,16386,16627,16851,17043,17191,17283,17314,17282,17189,17041,16848,16623,16382,  
16142,15917,15725,15578,15485,15454,15486,15579,15727,15920,16145,16386,16626,17000,17499,18093,18736,19381,  
19981,20490,20872,21097,21148,21019,20717,20259,19676,19005,18289,17576,16913,16342,15901,15619,15513,15590,  
15841,16250,16786,17412,18083,18754,19376,19907,20309,20554,20623,20512,20225,19783,19215,18557,17854,17152,  
16500,15939,15508,15235,15138,15222,15482,15897,16440,17073,17751,18428,19056,19593,20000,20250,20324,20218,  
19936,19499,18934,18280,17581,16883,16233,15676,15248,14978,14884,14970,15232,15650,16196,16831,17511,18190,  
18821,19360,19770,20023,20099,19995,19716,19280,18718,18066,17368,16672,16025,15469,15042,14774,14681,14769};

int muvolB[144] = { 1749, 1385, 2043, 3677, 6178,

9374,13048,16949,20812,24372,27389,29656,31018,31383,30727,29092,26592,23397,19723,15822,11959, 8398, 5381,  
3113, 1750, 1384, 2040, 3674, 6173, 9369,13042,16943,20806,24368,27385,29653,31017,31384,30729,29096,26597,23402,  
19729,15828,11965, 8403, 5385, 3116, 1751, 1591, 2381, 4081, 6622, 9843,13514,17392,21208,24698,27625,29790,31045,  
31304,30550,28834,26273,23041,19358,15474,11656, 8162, 5230, 3061, 1801, 1538, 2288, 4000, 6558, 9787,13468,17348,  
21165,24657,27588,29756,31014,31277,30526,28813,26255,23024,19343,15461,11643, 8150, 5218, 3048, 1788, 1524,2273,  
3984, 6542, 9771,13451,17333,21150,24644,27575,29745,31006,31270,30522,28810,26253,23024,19343,15462,11644,  
8150, 5217, 3046, 1785, 1519, 2267, 3977, 6534, 9762,13442,17323,21141,24636,27568,29740,31001,31268,30521,  
28811,26255,23027,19347,15465,11647, 8152, 5219, 3047};

int mucorB[144] =

{15485,15454,15486,15580,15728,15921,16145,16386,16627,16851,17043,17191,17283,17314,17282,17189,17040,16847,1  
6623,16382,16141,15917,15725,15577,15485,15454,15486,15579,15728,15920,16145,16386,16626,16851,17043,17190,172  
83,17314,17282,17189,17041,16848,16623,16382,16142,15918,15725,15578,15485,15397,15332,15295,15283,15296,15335  
,15394,15470,15556,15647,15735,15813,15877,15920,15941,15937,15908,15856,15785,15699,15604,15507,15414,15331,1  
5264,15218,15197,15201,15230,15283,15356,15445,15544,15645,15743,15831,15903,15955,15983,15986,15964,15918,158  
52,15771,15681,15588,15498,15419,15356,15313,15293,15300,15331,15386,15462,15552,15652,15755,15854,15943,16016  
,16068,16097,16101,16079,16034,15968,15887,15797,15704,15615,15535,15472,15429,15409,15415,15447,15501,15576,1  
5666,15765,15868,15966,16055,16128,16180,16208,16211,16189,16143,16077,15996,15905,15811,15721};

int muvolC[144] = {26589,23393,19719,15818,11956, 8395, 5378, 3112, 1750, 1385, 2042, 3676, 6176, 9372,13046,16947,

20810,24371,27388,29655,31018,31384,30727,29094,26594,23398,19725,15824,11961, 8400, 5382, 3114, 1751, 1384,  
2040, 3673, 6172, 9367,13040,16941,20804,24366,27384,29652,31017,31384,30729,29097,26599,23611,20074,16241,  
12420, 8884, 5864, 3572, 2160, 1719, 2281, 3809, 6196, 9281,12854,16670,20470,23994,27003,29290,30701,31138,30573,  
29043,26652,23564,19988,16169,12366, 8839, 5828, 3537, 2124, 1683, 2246, 3774, 6162, 9248,12822,16640,20441,23967,  
26978,29268,30681,31121,30558,29031,26642,23555,19981,16163,12361, 8833, 5821, 3530, 2116, 1674, 2236, 3763, 6150,  
9236,12809,16627,20429,23956,26968,29260,30675,31117,30555,29029,26642,23556,19983,16165,12362, 8835, 5822,  
3530, 2114, 1671, 2231, 3757, 6143, 9228,12801,16619,20421,23949,26962,29255,30671,31114,30555,29030};

int mucorC[144] =

{17040,16847,16623,16382,16141,15917,15725,15577,15485,15454,15486,15579,15728,15921,16145,16386,16627,16851,1  
7043,17191,17283,17314,17282,17189,17040,16847,16623,16382,16141,15917,15725,15578,15485,15454,15486,15579,157  
27,15920,16145,16386,16626,16851,17043,17190,17283,17314,17282,17189,17041,16791,16470,16098,15697,15293,14915  
,14586,14329,14160,14090,14123,14257,14482,14783,15137,15522,15910,16276,16593,16840,17001,17064,17025,16887,1  
6659,16356,16001,15616,15228,14864,14549,14305,14147,14088,14132,14275,14509,14817,15179,15570,15965,16336,166  
59,16912,17077,17144,17109,16975,16750,16451,16098,15716,15330,14968,14655,14412,14256,14198,14242,14387,14621  
,14930,15292,15685,16080,16451,16774,17027,17193,17261,17226,17091,16867,16568,16214,15832,15446,15084,14770,1  
4526,14370,14311,14355,14499,14732,15041,15403,15794,16189,16560,16883,17135,17300,17367,17332};/\*/\*

// **MUESTRAS DE UNA FALLA TRIFÁSICA AL FINAL DE UNA LÍNEA DE 100 KM //**

int muvola[144] = {20813,24374,27390,29657,31018,31383,30726,29091,26591,23395,19721,15820,11957, 8396, 5380,  
3113, 1750, 1385, 2041, 3675, 6175, 9370,13044,16945,20808,24369,27386,29654,31017,31384,30728,29095,26595,23400,  
19727,15826,11963, 8401, 5383, 3115, 1751, 1384, 2039, 3672, 6170, 9365,13038,16939,20802,23708,26152,27984,29054,



```

29261,28578,27048,24780,21936,18712,15330,12021, 9010, 6501, 4665, 3626, 3455, 4164, 5705, 7973,10812,14031,17409,
20717,23729,26239,28077,29117,29289,28581,27041,24774,21935,18717,15339,12031, 9020, 6509, 4671, 3630, 3458, 166,
5705, 7972,10811,14029,17408,20715,23727,26239,28078,29119,29293,28586,27047,24781,21943,18725,15347,12039,
9027, 6515, 4676, 3634, 3460, 4166, 5705, 7970,10809,14027,17405,20713,23726,26238,28078,29121,29296,28590,27052,
24788,21950,18732,15354,12046, 9033, 6521, 4680, 3637, 3461, 4166, 5703, 7968,10806,14023,17401};
int mucorA[144]= {16627,16851,17043,17191,17283,17314,17282,17188,17040,16847,16623,16382,16141,15917,15725,
15577,15485,15454,15486,15579,15728,15921,16145,16386,16627,16851,17043,17191,17283,17314,17282,17189,17041,
16848,16623,16382,16142,15917,15725,15578,15485,15454,15486,15579,15727,15920,16145,16386,16626,17028,17611,18
334,19148,19996,20816,21548,22140,22547,22740,22702,22433,21950,21282,20473,19575,18647,17749,16941,16275,1579
4,15528,15494,15691,16104,16702,17442,18273,19134,19966,20709,21311,21729,21932,21905,21647,21175,20517,19719,
18831,17912,17024,16225,15568,15096,14839,14814,15019,15441,16047,16796,17635,18505,19345,20096,20707,21133,21
344,21325,21076,20611,19961,19170,18289,17378,16497,15705,15054,14589,14338,14319,14531,14958,15571,16326,1717
0,18046,18892,19650,20267,20699,20916,20903,20659,20200,19556,18771,17895,16989,16113,15326,14680,14219,13973,
13958};
int muvoIB[144] = { 1749, 1385, 2043, 3677, 6178, 9374,13048,16949,20812,24372,27389,29656,31018,31383,30727,
29092,26592,23397,19723,15822,11959, 8398, 5381, 3113, 1750, 1384, 2040, 3674, 6173, 9369,13042,16943,20806,
24368,27385,29653,31017,31384,30729,29096,26597,23402,19729,15828,11965, 8403, 5385, 3116, 1751, 2562, 4357, 6437,
8702,11277,14212,17402,20633,23646,26192,28068,29135,29318,28610,27065,24792,21948,18727,15349,12042, 9031,
6522, 4684, 3644, 3472, 4181, 5721, 7988,10827,14046,17424,20731,23743,26253,28092,29132,29305,28597,27057,24790,
21951,18733,15354,12047, 9035, 6524, 4685, 3644, 3470, 4178, 5717, 7983,10822,14040,17418,20725,23738,26249,28088,
29130,29304,28597,27059,24793,21955,18737,15359,12051, 9038, 6526, 4686, 3644, 3469, 4175, 5713, 7978,10816,14034,
17412,20720,23733,26245,28085,29129,29304,28599,27061,24796,21959,18741,15363,12055, 9042, 6529, 4688};
int mucorB[144] =
{15485,15454,15486,15580,15728,15921,16145,16386,16627,16851,17043,17191,17283,17314,17282,17189,17040,16847,1
6623,16382,16141,15917,15725,15577,15485,15454,15486,15579,15728,15920,16145,16386,16626,16851,17043,17190,172
83,17314,17282,17189,17041,16848,16623,16382,16142,15918,15725,15578,15485,14529,13669,12945,12376,11981,11783
,11801,12042,12496,13137,13921,14797,15705,16583,17371,18017,18478,18724,18739,18522,18090,17474,16714,15866,1
4986,14137,13376,12757,12322,12102,12112,12354,12810,13452,14235,15107,16010,16883,17667,18309,18767,19009,190
20,18800,18365,17745,16983,16131,15248,14395,13630,13007,12569,12345,12352,12590,13043,13681,14461,15331,16231
,17101,17882,18521,18976,19216,19225,19003,18566,17943,17178,16324,15438,14583,13816,13190,12749,12522,12527,1
2762,13213,13848,14626,15493,16391,17259,18038,18676,19129,19367,19375,19151,18712,18087,17321};
int muvoIC[144] = {26589,23393,19719,15818,11956, 8395, 5378, 3112, 1750, 1385, 2042, 3676, 6176, 9372,13046,16947,
20810,24371,27388,29655,31018,31384,30727,29094,26594,23398,19725,15824,11961, 8400, 5382, 3114, 1751, 1384,
2040, 3673, 6172, 9367,13040,16941,20804,24366,27384,29652,31017,31384,30729,29097,26599,22882,18643,14731,
11397, 8614, 6363, 4702, 3738, 3570, 4249, 5754, 7996,10824,14041,17422,20734,23749,26260,28098,29138,29308,
28599,27058,24791,21951,18732,15354,12047, 9035, 6525, 4687, 3646, 3474, 4182, 5722, 7988,10828,14046,17424,
20731,23743,26254,28092,29133,29306,28599,27059,24793,21954,18736,15358,12050, 9038, 6526, 4687, 3645, 3472,4178,
5717, 7983,10821,14039,17417,20725,23737,26249,28089,29131,29305,28599,27061,24796,21957,18740,15361,12053,
9041,6528, 4688, 3645, 3470, 4175, 5713, 7977,10815,14033,17411,20719,23732,26245,28085,29129,29305,28600,27063};
int mucorC[144] =
{17040,16847,16623,16382,16141,15917,15725,15577,15485,15454,15486,15579,15728,15921,16145,16386,16627,16851,1
7043,17191,17283,17314,17282,17189,17040,16847,16623,16382,16141,15917,15725,15578,15485,15454,15486,15579,157
27,15920,16145,16386,16626,16851,17043,17190,17283,17314,17282,17189,17041,17595,17872,17873,17627,17175,16553
,15803,14970,14109,13275,12529,11921,11497,11287,11308,11560,12027,12679,13472,14355,15268,16150,16944,17595,1
8062,18313,18334,18123,17696,17085,16331,15487,14613,13768,13012,12397,11967,11752,11767,12012,12473,13119,139
06,14783,15690,16567,17356,18002,18464,18710,18725,18510,18078,17462,16703,15855,14976,14126,13366,12746,12311
,12090,12101,12342,12798,13439,14222,15095,15998,16871,17655,18298,18756,18999,19011,18792,18357,17737,16975,1
6123,15240,14387,13623,13000,12561,12337,12344,12581,13034,13672,14451,15321,16221,17091,17873};/*
/*****
////////////////////////////////////main()////////////////////////////////////
int main(void)
{
    init();
    delay();
    initLED();
    *ADCA_OFFSET0 = 0x3FF8;
    *ADCA_OFFSET4 = 0x3FF8;
    *ADCA_OFFSET1 = 0x3FF8;
    *ADCA_OFFSET5 = 0x3FF8;
    *ADCA_OFFSET2 = 0x3FF8;
    *ADCA_OFFSET6 = 0x3FF8;
    adca_setup();
}

```



```

timer_setup();
sci0_setup();
// Enable timer interrupt and start timers
asm(bfset #2000, x:TIMERC2_CTRL); // Enable timer C2 (turn it on)
// Enable ADCA adc
asm( bfcclr #4000, x:ADC_A_CTRL); // normal operation ADC stop
// Enable SCI0 transmitter & receiver
*ipr = 0xFE3F;//0x0A3F; // Enable interrupt level and IQRA & IRQB
// Enable maskable (Level 0) interrupts
asm(bfset #0100,sr);
asm(bfcclr #0200,sr);
while (siempre==1)
{
    if (hace==1)
    {
        if(nuevo==1)
        {
            ajuste = (mfr32Sqrt((preal*preal)+(pimag*pimag))*1.4142;
            //ajuste = pow(((preal*preal)+(pimag*pimag)),0.5);
            nuevo = 0;
        }
        /** VOLTAJE FASE A **/
        VAmag = (mfr32Sqrt(VArealp*VArealp+VAimagp*VAimagp))*1.4142;
        VAang = tfr16AsinOverPI(FRAC16(VAimagp/VAmag));
        if (VArealp<0)
            VAang = 32768-VAang;
        /** VOLTAJE FASE B **/
        VBmag = (mfr32Sqrt(VBrealp*VBrealp+VBimagp*VBimagp))*1.4142;
        VBang = tfr16AsinOverPI(FRAC16(VBimagp/VBmag));
        if (VBrealp<0)
            VBang = 32768-VBang;
        /** VOLTAJE FASE C **/
        VCmag = (mfr32Sqrt(VCrealp*VCrealp+VCimagp*VCimagp))*1.4142;
        VCang = tfr16AsinOverPI(FRAC16(VCimagp/VCmag));
        if (VCrealp<0)
            VCang = 32768-VCang;
        /** CORRIENTE DE SECUENCIA CERO **/
        I0real = (IArealp+IBrealp+ICrealp)/3;
        I0imag = (IAimagp+IBimagp+ICimagp)/3;
        I0xreal = (Mreal*I0real)-(Mimag*I0imag);
        I0ximag = (I0real*Mimag)+(I0imag*Mreal);
        /** CORRIENTE FASE A COMPENSADA **/
        IAcompreal = IArealp+I0xreal;
        IAcompimag = IAimagp+I0ximag;
        IAmag = (mfr32Sqrt(IAcompreal*IAcompreal+IAcompimag*IAcompimag))*1.4142;
        IAang = tfr16AsinOverPI(FRAC16(IAcompimag/IAmag));
        if (IAcompreal<0)
            IAang = 32768-IAang;
        /** CORRIENTE FASE B COMPENSADA **/
        IBcompreal = IBrealp+I0xreal;
        IBcompimag = IBimagp+I0ximag;
        IBmag = (mfr32Sqrt(IBcompreal*IBcompreal+IBcompimag*IBcompimag))*1.4142;
        IBang = tfr16AsinOverPI(FRAC16(IBcompimag/IBmag));
        if (IBcompreal<0)
            IBang = 32768-IBang;
        /** CORRIENTE FASE C COMPENSADA **/
        ICcompreal = ICrealp+I0xreal;
        ICcompimag = ICimagp+I0ximag;
        ICmag = (mfr32Sqrt(ICcompreal*ICcompreal+ICcompimag*ICcompimag))*1.4142;
        ICang = tfr16AsinOverPI(FRAC16(ICcompimag/ICmag));
        if (ICcompreal<0)
            ICang = 32768-ICang;
    }
}

```



```

//IMPEDANCIA MEDIDA DE SECUENCIA POSITIVA CON LA FASE A
ZA      = (VAmag/IAmag)*10;
angZA   = (VAang-IAang);
//IMPEDANCIA MEDIDA DE SECUENCIA POSITIVA CON LA FASE B
ZB      = (VBmag/IBmag)*10;
angZB   = (VBang-IBang);
//IMPEDANCIA MEDIDA DE SECUENCIA POSITIVA CON LA FASE C
ZC      = (VCmag/ICmag)*10;
angZC   = (VCang-ICang);
re = (tfr16CosPIx(angZA))*ZA;
re = re/32768-preal;
im = (tfr16SinPIx(angZA))*ZA;
im = im/32768-pimag;
falla = (mfr32Sqrt(re*re+im*im))*1.4142;
if (falla<ajuste)
    {
        asm(bfset      #$0004,X:GPIO_B_DR);
    }

hace=0;
/*Datos[0]=ZA;
Datos[1]=angZA;
Datos[2]=ZB;
Datos[3]=angZB;
Datos[4]=ZC;
Datos[5]=angZC;*/
}
asm(nop);
asm(nop);
}
return(0);
}
//////////////////////////////////////////////////end main//////////////////////////////////////
//////////////////////////////////////////////////init()//////////////////////////////////////
void init ( void )
{
    // ----- SET TIMER C2
    *TmrC2_CMP1 = 868; // Set the wait loop count, counter counts up
    *TmrC2_LOAD = 0x0000; // value to initialize the counter
    *TmrC2_CNTR = 0x0000; // counter set to 0
    // ----- SET ITCN
    *GPR12 = 0x4500; //prioridad para Transmitter ready y Transmitter complete
    *GPR13 = 0x7060; //prioridad para ADC y receiver full
}
//////////////////////////////////////////////////end init()//////////////////////////////////////
//////////////////////////////////////////////////SCI0 SET UP//////////////////////////////////////
sci0_setup(void)
{
    *SCI0BR = 260; // BAUD RATE 9600
    *SCI0CR = 0x0024;//00AC // [15] = 0 LOOP disabled, normal operation
    // [14] = 0 SWAI enabled, SCI enabled in wait mode
    // [13] = 0 RSRC, normal operation
    // [12] = 0 M 8 data characters, start bit and stop bit
    // [11] = 0 WAKE idle line wake up
    // [10] = 0 Non invert bits
    // [9] = 0 parity disabled
    // [8] = 0 even parity
    // [7] = 1 TEIE enable
    // [6] = 0 TIDLE disabled
    // [5] = 1 RIE enabled
    // [4] = 0 REIE disabled
    // [3] = 0 TE transmitter disabled
    // [2] = 1 RE receiver disabled
}

```



```

//[1] = 0 RWU normal operation
//[0] = 0 SBK no break characters transmitted
}
//////////////////////////////////end of SCI0 SET UP//////////////////////////////////
////////////////////////////////// ISR for SCI0 transmit complete//////////////////////////////////
#pragma interrupt saveall
void sci0_tdre(void)
{
    asm(move x:SCI0_SR,x0);
    asm(move x0,x:SCI0_SR);
if(aux==1) //ASCII retorno de carro (enter) 0x0D
    {
        n=0;
        aux=0;
        k=0;
        opcion();
    }
if (imprime==1)
    {
        *SCI0DR = dest[n]; //envio de dato
        n++;
        if (dest[n]=='$')
            {
                if(k==muestras)
                {
                    asm(bfclr #$0008, x:SCI0_CTRL);
                    asm(bfset #$1000,x:ADC_A_CTRL);
                    imprime=0;
                    k=0;
                    envio = 0;
                    asm(bfclr #$0001,X:GPIO_B_DR);
                }
                else
                {
                    opcion();
                    n=0;
                }
            }
    }
}
//////////////////////////////////end of SCI0 ISR transmit complete//////////////////////////////////
////////////////////////////////// RECEIVER FULL ISR//////////////////////////////////
#pragma interrupt saveall
void sci0_rdrf(void)
{
    asm(move x:SCI0_SR,x0);
    asm(move x0,x:SCI0_SR);
    mensaje= *SCI0DR;
    envio = mensaje;
    scan = mensaje;
    lee = mensaje;
    escribe = mensaje;
    if (mensaje!=10&&mensaje!=13)
        {
            elije++;
            if (elije==1)
                {
                    preal1=(mensaje-25)*0.1;
                    asm(bfset#$0002,X:GPIO_B_DR);
                }
            if (elije==2)
                {

```



```

        preal=preal1;
        pimag=mensaje-25;
        elije=0;
        nuevo=1;
        asm(bfclr#$0002,X:GPIO_B_DR);
    }
}
}
// end of SCIO receiver full ISR
// Timer C, Counter 2 setup()
timer_setup( void )
{
    *TmrC2_CNTRL = 0x1823;
        // [15:13]      =000    No operation, timer is disabled
        // [12:09]      = 1100 Count Source 40/16 = 2.5Mhz
        // [8:7]        = 00
        // [6]          = 0 ONCE Count repeatedly
        // [5]          = 1 LENGTH Countuntil compare then reinitialize
        // [4]          = 0 DIR Count up
        // [3]          = 0 Co-channel timers can't reinit counter
        // [2:0]       = 011 toggle OFLAG on successful compare

// Timer is running at 0.3125MHZ:
//          IP BUS          = 40 MHz
//          Prescaler (128) = 40/16 = 2.5 MHz
//          1 / 2.5MHz      = 400 nS
// Timer C2 interval is 400 nS, TmrC2_CMP1 = 868 -> Interval is 347.22 usec = 1.44KHz
// 400 nsec * 868 = 347.22 usec, 1/(2*347.2 usec) thus frequency is 1.44Khz
// 1440/60 = 24 muestras por ciclo.
return (0);
}
// end timer_setup()
// ADCA setup()
adca_setup( void )
{
    *ADCA_ADCR1 = 0x5805;    // 0x5805;
                            // [15] = 0 not used
                            // [14] = 1 stop, ADCA stopped
                            // [13] = 0 start, conversion initialitated by a write to START bit
                            // [12] = 1 sync, conversion initialitated by a sync pulse
                            // [11] = 1 eosie, end of scan interrupt enable
                            // [10] = 0 zcie, zero crossing interrupt disable
                            // [9]  = 0 LLMTIE, low limit interrupt disable
                            // [8]  = 0 HLMTIE, high limit interrupt disable
                            // [7:4]= 000 CHNCHG, single ended input
                            // [3]  = 0 not used
                            // [2:0]= 101 SMODE, triggered simultaneous
    *ADCA_ADSDIS = 0x0088;  // 0x0088;
                            // [15:14] = 00    TEST, normal mode
                            // [13:8]  = 000000 reserved
                            // [7:1]  = 1000100 disable samples 7,3
                            // [0]    = 0 enable sample 0-2,4-6

asm( bfset #$0800,x:ADC_A_STAT);
return(0);
}
// end adca_setup()
// ISR for adc interrupt()
#pragma interrupt saveall
void adcaISR(void)
{
    // Algoritmo para VA 24 muestras
    Datos[conta] = muvoIA[cont];/*ADCA_ADRSLT2 + 16376;
    VAreult = muvoIA[cont];/*ADCA_ADRSLT2;

```



```
VAreal = VAreal + se[conta]*VAresult;
VAimag = VAimag + co[conta]*VAresult;
Datos[conta+24] = mucorA[cont];/*ADCA_ADRSLT6 + 16376;
IAresult = mucorA[cont];/*ADCA_ADRSLT6;
IAreal = IAreal + se[conta]*IAresult;
IAimag = IAimag + co[conta]*IAresult;
Datos[conta+48] = muvolB[cont];/*ADCA_ADRSLT2 + 16376;
VBresult = muvolB[cont];/*ADCA_ADRSLT2;
VBreal = VBreal + se[conta]*VBresult;
VBimag = VBimag + co[conta]*VBresult;
Datos[conta+72] = mucorB[cont];/*ADCA_ADRSLT6 + 16376;
IBresult = mucorB[cont];/*ADCA_ADRSLT6;
IBreal = IBreal + se[conta]*IBresult;
IBimag = IBimag + co[conta]*IBresult;
Datos[conta+96] = muvolC[cont];/*ADCA_ADRSLT2 + 16376;
VCresult = muvolC[cont];/*ADCA_ADRSLT2;
VCreal = VCreal + se[conta]*VCresult;
VCimag = VCimag + co[conta]*VCresult;
Datos[conta+120] = mucorC[cont];/*ADCA_ADRSLT6 + 16376;
ICresult = mucorC[cont];/*ADCA_ADRSLT6;
ICreal = ICreal + se[conta]*ICresult;
ICimag = ICimag + co[conta]*ICresult;
conta++;
cont++;
if (conta == 24)
    {
        conta = 0;
        VArealp = VAreal/32768;
        VAimagp = VAimag/32768;
        IArealp = IAreal/32768;
        IAimagp = IAimag/32768;
        VAreal=0;
        VAimag=0;
        IAreal=0;
        IAimag=0;
        VBrealp = VBreal/32768;
        VBimagp = VBimag/32768;
        IBrealp = IBreal/32768;
        IBimagp = IBimag/32768;
        VBreal=0;
        VBimag=0;
        IBreal=0;
        IBimag=0;
        VCrealp = VCreal/32768;
        VCimagp = VCimag/32768;
        ICrealp = ICreal/32768;
        ICimagp = ICimag/32768;
        VCreal=0;
        VCimag=0;
        ICreal=0;
        ICimag=0;
        asm(bfclr #$0008, x:SCIO_CTRL);
        hace=1;
    }
if (cont == muestras)
    {
        if(envio == 10||envio == 13)
            {
                asm(bfclr #$1000,x:ADC_A_CTRL);
                imprime = 1;
                aux = 1;
                envio = 0;
            }
    }

```





```

asm(bfset #0080, x:SCIO_CTRL);
asm(bfset #0008, x:SCIO_CTRL);
asm(bfset #0001,X:GPIO_B_DR);
    }
    cont = 0;
}
//////////////////////////////////fin algoritmo 24 muestras//////////////////////////////////
asm(move x:ADC_A_STAT,x0);
asm(move x0,x:ADC_A_STAT);
}
////////////////////////////////// end adcaISR//////////////////////////////////
////////////////////////////////// ISR for IRQA interrupt//////////////////////////////////
#pragma interrupt saveall
void irqISR(void)
{
    asm(bfclr #0004,X:GPIO_B_DR);
    asm(nop);
}
////////////////////////////////// end irqISR()//////////////////////////////////
//////////////////////////////////delay//////////////////////////////////
void delay( void )
{
    int i;
    for ( i = 0; i < 0x1fff; i++)
    {
        asm ( nop );
        asm ( nop );
        asm ( nop );
        asm ( nop );
        asm ( nop );
    }
}
//////////////////////////////////end delay//////////////////////////////////
////////////////////////////////// opcion () ////////////////////////////////////
void opcion(void)
{
    /*** Conversion: Unsigned long to string***/
    ul2str(Datos[k],&dest[0]);
    dest[5]='\n';
    dest[6]='\r';
    dest[7]='$';
    k++;
}
//////////////////////////////////end opcion ()//////////////////////////////////
//////////////////////////////////initLED//////////////////////////////////
#pragma interrupt
void initLED()
{
    /* disable SSI peripheral */
    asm(bfclr #0001,X:GPIO_B_PER);
    asm(bfclr #0002,X:GPIO_B_PER);
    asm(bfclr #0004,X:GPIO_B_PER);
    /* output direction */
    asm(bfset #0001,X:GPIO_B_DDR);
    asm(bfset #0002,X:GPIO_B_DDR);
    asm(bfset #0004,X:GPIO_B_DDR);
    /* off at first */
    asm(bfclr #0001,X:GPIO_B_DR);
    asm(bfclr #0002,X:GPIO_B_DR);
    asm(bfclr #0004,X:GPIO_B_DR);
}
//////////////////////////////////end intLED//////////////////////////////////

```



**//Funcion de converscion en Ensamblador**

```

void ul2str(unsigned long val, char * buff)
{
asm{
Ful2str:
    clr    X0                ; Reset digits counter

Ful2str_01:
    ;*****
    ;* val = val/10, *SP++ = val%10
    ;*****
    clr    B                ; Clear dividend register
    move   A1,B0            ; Load high byte of dividend
    movei  #10,Y0          ; Load divisor
    bfclr  #$0001,SR
    rep    #17
    div   Y0,B              ; Positive quotient in B0
    move  B0,A1            ; Save quotient in A1
    add  Y0,B              ; Restore remainder in B1
    asr  B                ; Required for correct integer remainder
    move A0,B0
    bfclr  #$0001,SR
    rep    #17
    div   Y0,B              ; Positive quotient in B0
    move  B0,A0            ; Save quotient in A0
    add  Y0,B              ; Restore remainder in B1
    beq  Ful2str_02        ; if 0, then stop cycle of conversion
    asr  B                ; Required for correct integer remainder
    add  #48,B              ; Convert number to char (add the '0')
    inc  X0                ; Increment cycle counter
    push B1                ; Store converted symbol to the stack
    jmp  Ful2str_01        ; Continue conversion

Ful2str_02:
    ;*****
    ;* Add the '0' to the first place (if its were empty)
    ;*****
    cmp   #0,X0            ; if number of the digit is > 0 then skip
    bgt  Ful2str_03
    movei #48,X:(R2+0)
    lea  (R2)+
    jmp  Ful2str_04

Ful2str_03:
    ;*****
    ;* Copy converted chars from the stack to the memory location,
    ;* that was pointered by the second parameter of the function
    ;*****
    pop  B1
    move B1,X:(R2)+
    dec  X0
    bne  Ful2str_03

Ful2str_04:
    ;*****
    ;* Add the to the last place
    ;*****
    movei #0,X:(R2+0)
    rts

}
}

```

## APÉNDICE D

### D. PROGRAMA DESARROLLADO EN LabVIEW PARA INTERACTUAR CON EL DSP.

Para comenzar a explicar la operación del programa desarrollado en el lenguaje gráfico de LabVIEW se observa la figura D.1. donde se muestra como se realiza la programación para le lectura de datos.

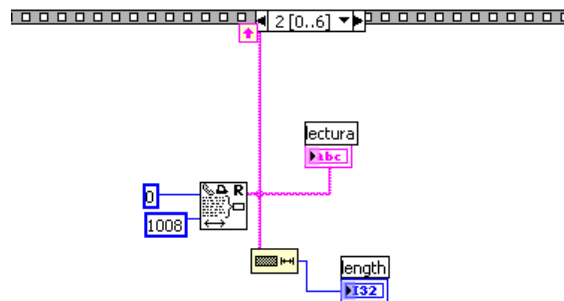


Figura D.1. Lectura de datos del DSP.

Después de obtener los datos en una variable tipo string se procede a separarlos y acomodarlos en los respectivos arreglos de variables numéricas de tipo flotante correspondientes a voltaje y corriente de cada una de las fases, esto se puede observar en la figura D.2.

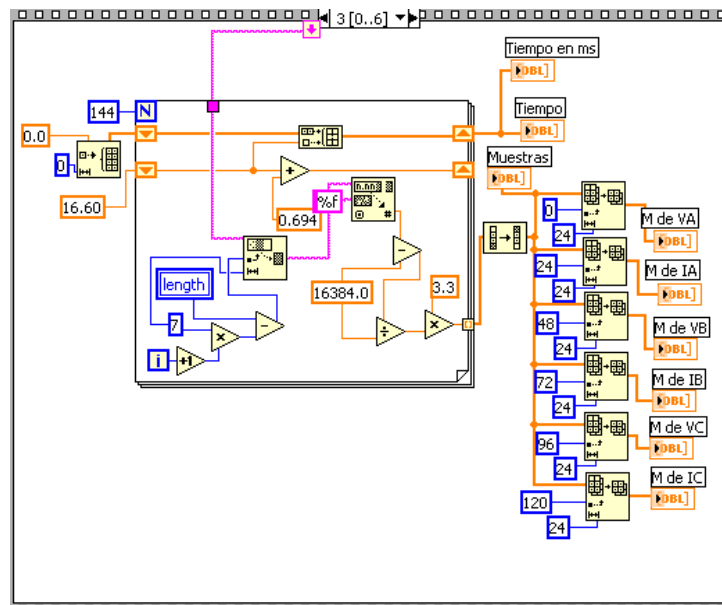


Figura D.2. Formación de arreglos de muestras correspondientes a voltajes y corrientes.

Una vez formados los arreglos, se realiza el cálculo de los fasores a frecuencia fundamental de cada señal, mediante el subinstrumento (SI) mostrado en la figura D.3. Aquí se puede observar la aplicación del filtro digital de errores mínimos cuadrados. En el arreglo “F.P. real”, se encuentran los coeficientes, de dicho filtro, para formar la parte real, y en “F.P. imag” para formar la parte imaginaria. En este mismo subinstrumento, se realiza la conversión del fasor de forma rectangular a forma polar. Teniendo los fasores, se divide el voltaje, entre la corriente compensada de la fase fallada, esto se realiza con el SI de la figura D.4.

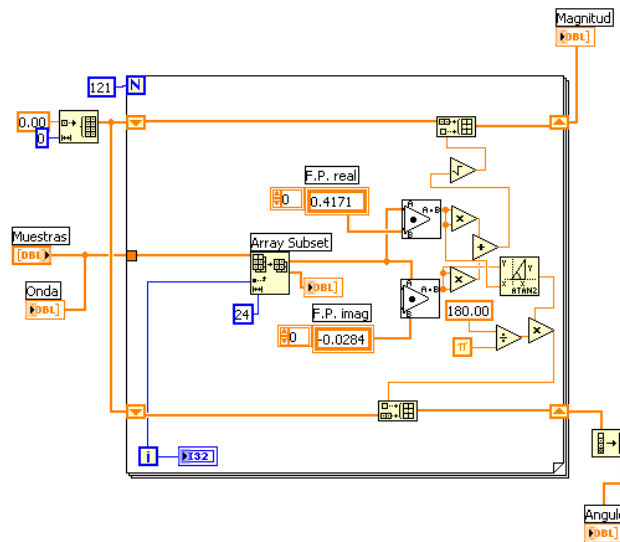


Figura D.3. Subinstrumento FasoresV3.vi

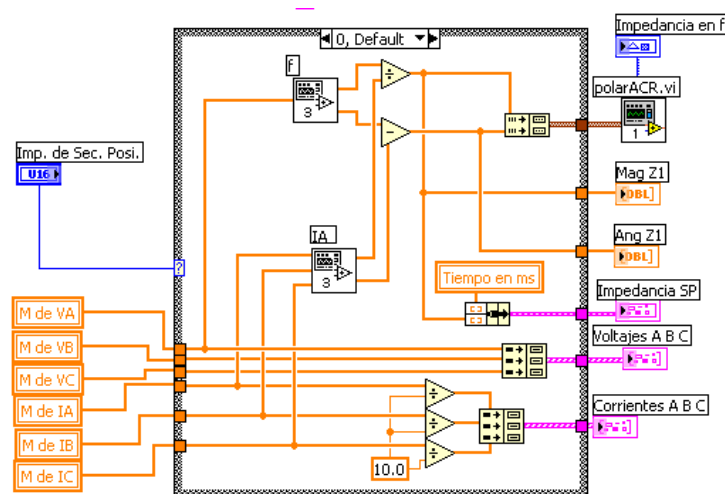


Figura D.4. Cálculo de la impedancia de secuencia positiva con IA de falla compensada.

Para compensar la corriente de la fase fallada, es necesario tener la información de los fasores de corriente de todas las fases (IA, IB, IC), y el valor de impedancia de secuencia positiva de la línea, dada por su magnitud (es una constante), para esto se utiliza el SI de la figura D.5.

Posteriormente, se calcula si la impedancia medida es menor a la impedancia de ajuste, como se muestra en la figura D.6., esto con la finalidad de detectar una falla.

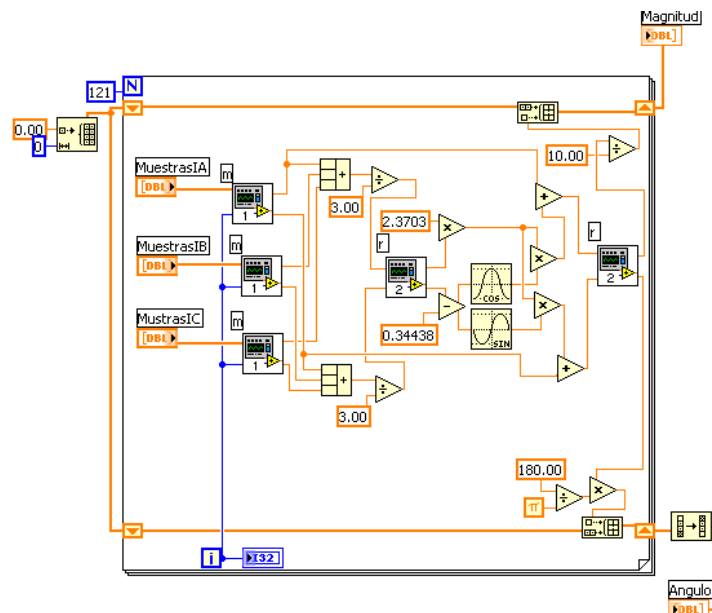


Figura D.5. Subinstrumento IFcompensada.vi

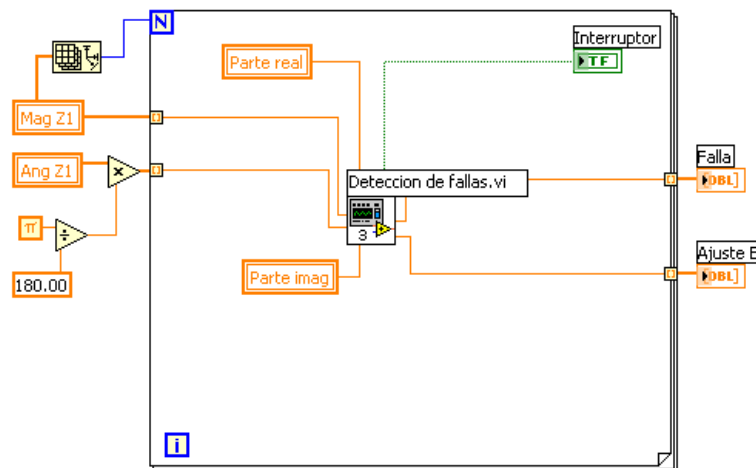


Figura D.6. Uso del subinstrumento para detección de fallas.

El SI para detectar fallas en el sistema, se muestra en la figura D.7. en este algoritmo se incluye la señal de alerta cuando se cumple la desigualdad, es decir, la impedancia medida es menor a la impedancia de ajuste.

La gráfica de la impedancia medida en forma polar, se puede realizar con el algoritmo mostrado en la figura D.8.

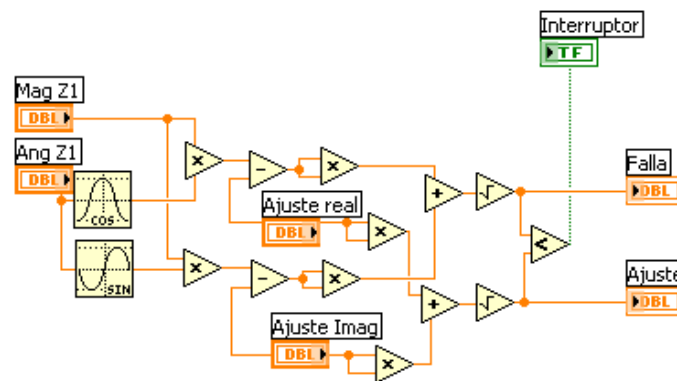


Figura D.7. Subinstrumento “Detección defallas.vi”

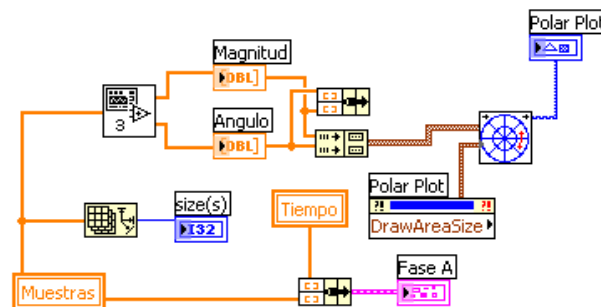


Figura D.8. Programación para graficar la impedancia en forma polar.

Por último, el algoritmo empleado para el envío de datos, es decir, para la actualización de los ajustes, se muestra en la figura D.9.

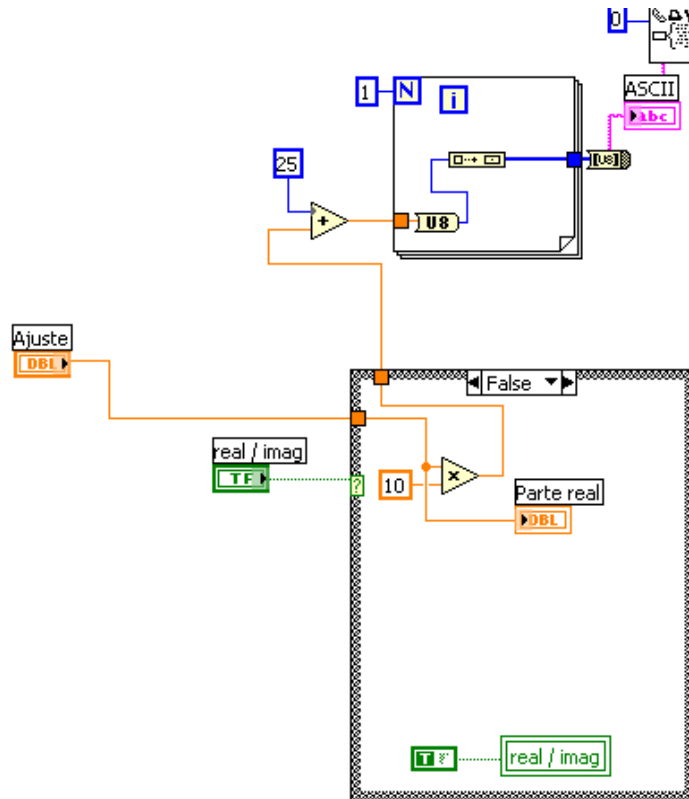


Figura D.9. Algoritmo para el envío de datos (nuevos ajustes).

## APÉNDICE E

### E. PROGRAMA EN MATLAB DEL FILTRO DE MÍNIMOS ERRORES CUADRADOS.

#### E.1. Código del programa.

```
clear all;
clc;
fm=1440; %frecuencia de muestreo 1440(24 muestras)
fn=60; %frecuencia del sistema
nm=24; %numero de muestras
nf=10; %# de funciones, 10 para fundamental, 2ª, 3ª, 4ª armónica, CD, CD decreciente.
aux=(nm)/2+1;
aux1=fn*2;
WAt=pi*aux1/fm;
for i=1:nm
    for j=1:nf
        if j==1
            A(i,j) = cos((i-aux)*WAt);
        end
        if j==2
            A(i,j) = sin((i-aux)*WAt);
        end
        if j==3
            A(i,j) = cos(2*(i-aux)*WAt);
        end
        if j==4
            A(i,j) = sin(2*(i-aux)*WAt);
        end
        if j==5
            A(i,j) = cos(3*(i-aux)*WAt);
        end
        if j==6
            A(i,j) = sin(3*(i-aux)*WAt);
        end
        if j==7
            A(i,j) = cos(4*(i-aux)*WAt);
        end
        if j==8
            A(i,j) = sin(4*(i-aux)*WAt);
        end
        if j==9
            A(i,j) = 1;
        end
        if j==10
            A(i,j) = i-aux;
        end
    end
end
```





end  
A  
Z=pinv(A)

**E.2. Corrida del programa.**

A =

-1.0000	-0.0000	1.0000	0.0000	-1.0000	-0.0000	1.0000	0.0000	1.0000	-12.0000
-0.9659	-0.2588	0.8660	0.5000	-0.7071	-0.7071	0.5000	0.8660	1.0000	-11.0000
-0.8660	-0.5000	0.5000	0.8660	0.0000	-1.0000	-0.5000	0.8660	1.0000	-10.0000
-0.7071	-0.7071	-0.0000	1.0000	0.7071	-0.7071	-1.0000	-0.0000	1.0000	-9.0000
-0.5000	-0.8660	-0.5000	0.8660	1.0000	0.0000	-0.5000	-0.8660	1.0000	-8.0000
-0.2588	-0.9659	-0.8660	0.5000	0.7071	0.7071	0.5000	-0.8660	1.0000	-7.0000
0.0000	-1.0000	-1.0000	-0.0000	-0.0000	1.0000	1.0000	0.0000	1.0000	-6.0000
0.2588	-0.9659	-0.8660	-0.5000	-0.7071	0.7071	0.5000	0.8660	1.0000	-5.0000
0.5000	-0.8660	-0.5000	-0.8660	-1.0000	-0.0000	-0.5000	0.8660	1.0000	-4.0000
0.7071	-0.7071	0.0000	-1.0000	-0.7071	-0.7071	-1.0000	-0.0000	1.0000	-3.0000
0.8660	-0.5000	0.5000	-0.8660	0.0000	-1.0000	-0.5000	-0.8660	1.0000	-2.0000
0.9659	-0.2588	0.8660	-0.5000	0.7071	-0.7071	0.5000	-0.8660	1.0000	-1.0000
1.0000	0	1.0000	0	1.0000	0	1.0000	0	1.0000	0
0.9659	0.2588	0.8660	0.5000	0.7071	0.7071	0.5000	0.8660	1.0000	1.0000
0.8660	0.5000	0.5000	0.8660	0.0000	1.0000	-0.5000	0.8660	1.0000	2.0000
0.7071	0.7071	0.0000	1.0000	-0.7071	0.7071	-1.0000	0.0000	1.0000	3.0000
0.5000	0.8660	-0.5000	0.8660	-1.0000	0.0000	-0.5000	-0.8660	1.0000	4.0000
0.2588	0.9659	-0.8660	0.5000	-0.7071	-0.7071	0.5000	-0.8660	1.0000	5.0000
0.0000	1.0000	-1.0000	0.0000	-0.0000	-1.0000	1.0000	-0.0000	1.0000	6.0000
-0.2588	0.9659	-0.8660	-0.5000	0.7071	-0.7071	0.5000	0.8660	1.0000	7.0000
-0.5000	0.8660	-0.5000	-0.8660	1.0000	-0.0000	-0.5000	0.8660	1.0000	8.0000
-0.7071	0.7071	-0.0000	-1.0000	0.7071	0.7071	-1.0000	0.0000	1.0000	9.0000
-0.8660	0.5000	0.5000	-0.8660	0.0000	1.0000	-0.5000	-0.8660	1.0000	10.0000
-0.9659	0.2588	0.8660	-0.5000	-0.7071	0.7071	0.5000	-0.8660	1.0000	11.0000

Z =

Columns 1 through 14

<b>-0.0284</b>	<b>-0.0774</b>	<b>-0.0891</b>	<b>-0.0685</b>	<b>-0.0366</b>	<b>-0.0119</b>	<b>0.0023</b>	<b>0.0154</b>	<b>0.0355</b>	<b>0.0601</b>	<b>0.0787</b>	<b>0.0842</b>	<b>0.0797</b>	<b>0.0740</b>
<b>0.4171</b>	<b>0.0019</b>	<b>-0.1701</b>	<b>-0.1318</b>	<b>-0.0338</b>	<b>-0.0072</b>	<b>-0.0656</b>	<b>-0.1275</b>	<b>-0.1192</b>	<b>-0.0504</b>	<b>0.0076</b>	<b>0.0062</b>	<b>-0.0278</b>	<b>-0.0277</b>
0.0284	0.0691	0.0586	0.0096	-0.0467	-0.0818	-0.0857	-0.0660	-0.0355	-0.0011	0.0352	0.0685	0.0870	0.0787
-0.2049	0.0301	0.1353	0.1191	0.0533	0.0056	-0.0087	-0.0185	-0.0490	-0.0875	-0.0964	-0.0553	0.0137	0.0659
-0.0284	-0.0558	-0.0169	0.0493	0.0884	0.0686	0.0023	-0.0651	-0.0895	-0.0578	0.0065	0.0626	0.0797	0.0524
0.1326	-0.0515	-0.1242	-0.0821	0.0122	0.0822	0.0890	0.0440	-0.0150	-0.0562	-0.0677	-0.0501	-0.0088	0.0433
0.0284	0.0386	-0.0248	-0.0737	-0.0467	0.0320	0.0810	0.0479	-0.0355	-0.0845	-0.0482	0.0380	0.0870	0.0482
-0.0951	0.0668	0.1015	0.0166	-0.0809	-0.0889	-0.0040	0.0829	0.0829	-0.0020	-0.0834	-0.0785	0.0063	0.0834
0.0142	0.0401	0.0501	0.0465	0.0391	0.0368	0.0405	0.0448	0.0448	0.0411	0.0384	0.0398	0.0435	0.0449
-0.0549	-0.0031	0.0169	0.0096	-0.0051	-0.0097	-0.0023	0.0062	0.0062	-0.0011	-0.0065	-0.0037	0.0037	0.0065

Columns 15 through 24

<b>0.0710</b>	<b>0.0651</b>	<b>0.0479</b>	<b>0.0192</b>	<b>-0.0097</b>	<b>-0.0266</b>	<b>-0.0321</b>	<b>-0.0420</b>	<b>-0.0753</b>	<b>-0.1354</b>
<b>0.0331</b>	<b>0.1060</b>	<b>0.1192</b>	<b>0.0628</b>	<b>0.0100</b>	<b>0.0421</b>	<b>0.1450</b>	<b>0.1874</b>	<b>0.0182</b>	<b>-0.3956</b>
0.0428	-0.0062	-0.0479	-0.0698	-0.0737	-0.0671	-0.0513	-0.0169	0.0448	0.1271
0.0764	0.0602	0.0490	0.0504	0.0360	-0.0228	-0.1080	-0.1465	-0.0606	0.1633
-0.0011	-0.0527	-0.0771	-0.0613	-0.0097	0.0539	0.0929	0.0758	-0.0031	-0.1138
0.0806	0.0739	0.0150	-0.0646	-0.1066	-0.0711	0.0232	0.0998	0.0759	-0.0737
-0.0405	-0.0895	-0.0479	0.0440	0.0930	0.0467	-0.0513	-0.1002	-0.0386	0.0966
0.0741	-0.0107	-0.0829	-0.0681	0.0167	0.0809	0.0556	-0.0293	-0.0668	0.0229
0.0422	0.0386	0.0386	0.0428	0.0465	0.0442	0.0369	0.0332	0.0432	0.0691
0.0011	-0.0062	-0.0062	0.0023	0.0097	0.0051	-0.0096	-0.0169	0.0031	0.0549

>>

Las filas en negrita representan al filtro para obtener los fasores a frecuencia fundamental.