



UNIVERSIDAD MICHOACANA DE  
SAN NICOLÁS DE HIDALGO

*DIVISIÓN DE ESTUDIOS DE POSGRADO  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA*

“Análisis de Redes Eléctricas Utilizando Técnicas  
de Procesamiento en Paralelo de Grano Fino  
Basadas en OpenMP”

TESIS

QUE PARA OBTENER EL GRADO DE  
Maestro en Ciencias en Ingeniería Eléctrica

PRESENTA

Ing. Omar Antonio Rico Hernández

DIRECTOR DE TESIS

Dr. Antonio Ramos Paz

MORELIA, MICHOACÁN      FEBRERO, 2012



---

.

# DEDICATORIA

Esta tesis está dedicada a mis padres Florentino Rico y Juana Hernández, a mi hija Estefanía, a mi esposa Karen Camarena y a mis hermanos Carlos y Nancy quienes siempre me han brindado su apoyo incondicional.

# AGRADECIMIENTOS

- A la Universidad Michoacana de San Nicolás de Hidalgo donde cursé mis estudios de Licenciatura y ahora Maestría.
- A la división de estudios de posgrado de la Facultad de Eléctrica donde recibí el mejor de los tratos como estudiante.
- Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo económico recibido y que fue de gran importancia.
- Agradezco especialmente al Dr. Antonio Ramos Paz por su total apoyo durante el tiempo en que curse mis estudios de maestría y su guía para la elaboración de esta tesis.
- A los profesores de este posgrado por transmitir su vasto conocimiento y que me han hecho crecer como persona y profesionista.
- A mis compañeros con quienes pasé agradables momentos y que me brindaron su apoyo en los momentos más difíciles, especialmente a Félix Guadalupe, José Roberto, José Pilar, Julio Cesar, Omar Beltrán y Víctor Armando.
- A mis padres y hermanos por el apoyo incondicional que recibí en todo momento.
- A mi esposa e hija que siempre me han motivado para seguir adelante.
- Y a todas las personas que contribuyeron directa o indirectamente para poder culminar esta tesis.

---

## RESUMEN

En esta tesis se presenta el uso de técnicas de procesamiento en paralelo de grano fino en el análisis de redes eléctricas de gran escala que contienen elementos lineales, no lineales y variantes en el tiempo. Los resultados obtenidos muestran el comportamiento en estado estacionario periódico, así como el análisis armónico de las formas de onda de corriente cuando se incorporan elementos de electrónica de potencia de la familia de los FACTS en especial el STATCOM. El procesamiento en paralelo que se implementó en esta tesis explota los avances recientes donde se utilizan múltiples núcleos de procesamiento en la solución de tareas paralelas.

Los resultados obtenidos mediante la aplicación del procesamiento en paralelo se comparan con los resultados obtenidos en forma secuencial en términos del *speedup*, donde se demuestra que las eficiencias más altas se obtienen cuando las dimensiones del caso de estudio se incrementan. La implementación fue hecha en el lenguaje de programación C/C++.

**Palabras clave:** Procesamiento en paralelo, OpenMP, grano fino, estado estacionario periódico, análisis transitorio, *speedup*.

---

## ABSTRACT

This thesis presents a fine-grained parallel processing technique applied to the fourth-order Runge-Kutta method in order to obtain the transient and steady state behavior in power electric systems containing linear and non linear time varying elements. This approach is based in the application of the parallel processing platform named OpenMP. This work presents six test cases showing the advantage of using the proposed technique compared with traditional sequential technique. The results obtained by the application of the parallel processing tool are compared with results obtained by sequentially in terms of speedup, which shows that higher efficiencies are obtained when the dimensions of the case study increases. The implementation was done in the programming language C / C ++.

# Índice general

<b>Dedicatoria</b>	<b>III</b>
<b>Agradecimientos</b>	<b>IV</b>
<b>Resumen</b>	<b>V</b>
<b>Abstract</b>	<b>VI</b>
<b>Índice general</b>	<b>VII</b>
<b>Índice de figuras</b>	<b>XI</b>
<b>Índice de tablas</b>	<b>XIV</b>
<b>Lista de símbolos y abreviaturas</b>	<b>XVI</b>
<b>Lista de Publicaciones</b>	<b>XVIII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Simulación de Circuitos Eléctricos . . . . .	2
1.2.1. Formación de las ecuaciones que modela la dinámica del circuito . . . . .	3
1.2.2. Método de análisis en el dominio del tiempo . . . . .	3
1.2.3. Método análisis en el dominio de la frecuencia . . . . .	4
1.2.4. Método de análisis híbrido . . . . .	4
1.2.5. Técnicas de acercamiento rápido al estado estacionario periódico . . . . .	5
1.3. Procesamiento en paralelo . . . . .	6
1.4. Justificación . . . . .	8
1.5. Objetivos . . . . .	8
1.6. Metodología . . . . .	9

1.7. Descripción de los Capítulos . . . . .	10
<b>2. Metodologías para la Obtención del Estado Transitorio y Estado Estacionario Periódico en el Dominio del Tiempo</b>	<b>11</b>
2.1. Introducción . . . . .	11
2.2. Ecuaciones Diferenciales Ordinarias . . . . .	12
2.3. Métodos de integración numérica . . . . .	13
2.3.1. Regla Trapezoidal . . . . .	15
2.3.2. Método de Euler . . . . .	15
2.4. Método de Runge-Kutta . . . . .	16
2.5. Método de Fuerza Bruta (FB) . . . . .	17
2.6. Conclusiones . . . . .	19
<b>3. Procesamiento en Paralelo</b>	<b>20</b>
3.1. Introducción . . . . .	20
3.2. Procesamiento en paralelo . . . . .	21
3.3. Arquitecturas de procesamiento en paralelo . . . . .	24
3.3.1. Procesamiento con memoria distribuida . . . . .	25
3.3.2. Procesamiento con memoria compartida . . . . .	26
3.4. Paralelismo a nivel de threads . . . . .	27
3.5. Paralelización de programas . . . . .	28
3.6. Speedup y Eficiencia . . . . .	29
3.7. Ley de Amdahl . . . . .	30
3.8. Granularidad . . . . .	31
3.9. Plataformas del procesamiento en paralelo . . . . .	32
3.10. OpenMP . . . . .	33
3.10.1. Modelo de Ejecución . . . . .	34
3.11. Conclusiones . . . . .	36
<b>4. Propuesta de Análisis Para Sistemas Eléctricos Usando Procesamiento en Paralelo de Grano Fino</b>	<b>37</b>
4.1. Introducción . . . . .	37
4.2. Proceso de paralelización del método de integración . . . . .	39
4.3. Descripción del código en C/C++ utilizando OpenMP . . . . .	41
4.4. Conclusiones . . . . .	43



<b>5. Modelado de elementos no lineales y variantes en el tiempo</b>	<b>44</b>
5.1. Introducción . . . . .	44
5.2. Representación monofásica de los elementos en un SEP . . . . .	45
5.2.1. Línea de transmisión . . . . .	46
5.2.2. Banco de capacitores . . . . .	46
5.2.3. Rama magnetizante . . . . .	47
5.2.4. Generador . . . . .	48
5.2.5. Transformador . . . . .	49
5.2.6. Horno de arco eléctrico . . . . .	50
5.2.7. Dispositivos FACTS . . . . .	52
5.2.8. Compensador estático de VARs (STATCOM) . . . . .	57
5.2.9. Modelo de cargas . . . . .	62
5.3. Conclusiones . . . . .	65
<b>6. Casos de estudio</b>	<b>66</b>
6.1. Introducción . . . . .	66
6.2. Metodología de solución . . . . .	67
6.3. Caso de estudio 1. Sistema 3 nodos . . . . .	69
6.4. Caso de estudio 2. Sistema IEEE-14 nodos modificado . . . . .	75
6.5. Caso de estudio 3. Sistema IEEE-30 nodos modificado . . . . .	83
6.6. Caso de estudio 4. Sistema IEEE-57 nodos modificado . . . . .	86
6.7. Caso de estudio 6. Sistema IEEE-118 nodos modificado . . . . .	90
6.8. Conclusiones . . . . .	94
<b>7. Conclusiones generales y recomendaciones para trabajos futuros</b>	<b>95</b>
7.1. Conclusiones generales . . . . .	95
7.2. Recomendaciones para trabajos futuros . . . . .	96
<b>Bibliografía</b>	<b>98</b>
<b>A. Datos de los Sistemas de Prueba</b>	<b>105</b>
A.1. Sistema de 3 nodos . . . . .	105
A.2. Sistema IEEE-14 nodos . . . . .	106
A.3. Sistema IEEE-30 nodos . . . . .	108
A.4. Sistema IEEE-57 nodos . . . . .	110
<b>B. Cantidades en Por Unidad de Frecuencia Angular</b>	<b>113</b>

<b>C. Rama magnetizante en ATP</b>	<b>116</b>
------------------------------------	------------

# Índice de figuras

2.1. Método de la regla trapezoidal . . . . .	15
2.2. Método de Euler . . . . .	16
2.3. Método de Runge-Kutta de 4to orden . . . . .	17
2.4. Método de fuerza bruta (FB) . . . . .	18
3.1. Procesamiento en paralelo . . . . .	22
3.2. Arquitectura de la taxonomía de Flynn . . . . .	25
3.3. Formas de organización de memoria en MIMD . . . . .	25
3.4. Proceso con memoria distribuida . . . . .	26
3.5. Procesamiento con memoria compartida. a) Vista abstracta b) Memoria compartida con módulos de memoria . . . . .	27
3.6. Comportamiento del tiempo de ejecución de un programa usando múltiples elementos de proceso . . . . .	30
3.7. Modelo OpenMP fork-join . . . . .	35
4.1. Representación de un sistema lineal . . . . .	39
4.2. Cálculo de primer elemento de $\dot{X}$ . . . . .	40
4.3. Esquema de procesamiento en paralelo para la evaluación de $\dot{X} = A \cdot X + B \cdot U$ . . . . .	40
5.1. Modelo de la línea de transmisión . . . . .	46
5.2. Banco de capacitores . . . . .	46
5.3. Rama magnetizante . . . . .	47
5.4. Modelo del generador . . . . .	48
5.5. Modelo del transformador monofásico incluyendo el fenómeno de saturación . . . . .	49
5.6. Modelo del Horno del Arco Eléctrico . . . . .	50
5.7. Modelo del RCT . . . . .	54
5.8. Modelo del CSCT . . . . .	55
5.9. Mallas formadas en el modelo del CSCT para su análisis . . . . .	55

## ÍNDICE DE FIGURAS

---

5.10. Función de conmutación . . . . .	57
5.11. Diagrama de Flujo de la función de conmutación . . . . .	58
5.12. Modelo del STATCOM . . . . .	59
5.13. Convertidor trifásico de seis pulsos . . . . .	60
5.14. Circuito equivalente para la fase A . . . . .	61
5.15. Función continua de conmutación $S_s$ . . . . .	62
5.16. Modelo de carga resistiva . . . . .	63
5.17. Modelo de carga inductiva . . . . .	63
5.18. Modelo de carga resistiva-inductiva-capacitiva . . . . .	64
6.1. CPU E54045 Intel (R) Xeon (R) . . . . .	68
6.2. Red de prueba de 3 nodos . . . . .	69
6.3. Gráfica del <i>speedup</i> . Caso de estudio 3 nodos monofásico . . . . .	70
6.4. Forma de onda de corriente de la línea 1 hasta alcanzar su EEP . . . . .	71
6.5. Contenido armónico para diferentes criterios de convergencia . . . . .	73
6.6. Forma de onda de corriente (a) en la línea 1 y su contenido armónico (b) . . . . .	73
6.7. Representación del sistema de 3 nodos en ATP . . . . .	74
6.8. Formas de onda de corriente en la línea 1 obtenidas con C/C++ y ATP. . . . .	74
6.9. Formas de onda de voltaje en el bus 2 obtenidas con C/C++ y ATP . . . . .	75
6.10. Sistema de 14 nodos . . . . .	75
6.12. Forma de onda en EEP de la corriente en la línea 20 caso monofásico . . . . .	77
6.11. Gráfica del <i>speedup</i> . Caso de estudio IEEE-14 nodos monofásico . . . . .	77
6.13. Forma de onda de voltaje en el nodo 14 . . . . .	78
6.14. Gráfica del <i>speedup</i> . Caso de estudio IEEE-14 nodos trifásico . . . . .	79
6.15. Forma de onda en EEP de la corriente en la línea 20 caso trifásico . . . . .	80
6.16. Forma de onda de voltaje en el nodo 14 caso trifásico . . . . .	80
6.17. Gráfica del <i>speedup</i> . Caso de estudio IEEE-14 nodos trifásico con un STATCOM . . . . .	82
6.18. Forma de onda en EEP de la corriente en la línea 20 caso trifásico con STATCOM . . . . .	82
6.19. Forma de onda de voltaje en el nodo 14 con un STATCOM caso trifásico . . . . .	83
6.20. Gráfica del <i>speedup</i> . Caso de estudio IEEE-30 nodos monofásico . . . . .	84
6.21. Gráfica del <i>speedup</i> . Caso de estudio IEEE-30 nodos trifásico . . . . .	85
6.22. Forma de onda de corriente en EEP y contenido armónico en la línea 29 . . . . .	86
6.23. Forma de onda de voltaje en el nodo 22 caso trifásico . . . . .	86
6.24. Gráfica del <i>speedup</i> . Caso de estudio IEEE-57 nodos monofásico . . . . .	88
6.25. Gráfica del <i>speedup</i> . Caso de estudio IEEE-57 nodos trifásico . . . . .	89
6.26. Forma de onda de corriente en EEP y contenido armónico en la línea 78 . . . . .	90

## ÍNDICE DE FIGURAS

---

6.27. Forma de onda de voltaje en el nodo 57 caso trifásico . . . . .	90
6.28. Gráfica del <i>speedup</i> . Caso de estudio IEEE-118 nodos monofásico . . . . .	91
6.29. Gráfica del <i>speedup</i> . Caso de estudio IEEE-118 nodos trifásico . . . . .	93
6.30. Forma de onda de corriente en EEP y contenido armónico en la línea 40 . . . . .	93
6.31. Forma de onda de voltaje en el nodo 31 para el caso trifásico . . . . .	94
C.1. Curva de saturación utilizada para la rama magnetizante en ATP . . . . .	117
C.2. Respuesta obtenida en ATP y con el método polinomial . . . . .	118

# Índice de tablas

4.1. Fragmento del código en C/C++ para análisis de SEP usando procesamiento en paralelo . . . . .	42
6.1. Proceso de convergencia. Caso de estudio 3 nodos monofásico . . . . .	70
6.2. Valores de <i>speedup</i> y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio 3 nodos monofásico. . . . .	70
6.3. Proceso de convergencia. Caso de estudio IEEE-14 nodos monofásico . . . . .	76
6.4. Valores de <i>speedup</i> y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-14 monofásico . . . . .	76
6.5. Proceso de convergencia. Caso de estudio IEEE-14 nodos trifásico . . . . .	79
6.6. Valores de <i>speedup</i> y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-14 trifásico . . . . .	79
6.7. Proceso de convergencia. Caso de estudio IEEE-14 nodos trifásico con un STATCOM . . . . .	81
6.8. Valores de <i>speedup</i> y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-14 trifásico con un STATCOM . . . . .	81
6.9. Proceso de convergencia. Caso de estudio IEEE-30 nodos monofásico . . . . .	84
6.10. Valores de <i>speedup</i> y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-30 monofásico . . . . .	84
6.11. Proceso de convergencia. Caso de estudio IEEE-30 nodos trifásico . . . . .	85
6.12. Valores de <i>speedup</i> y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-30 trifásico . . . . .	85
6.13. Proceso de convergencia. Caso de estudio IEEE-57 nodos monofásico . . . . .	87
6.14. Valores de <i>speedup</i> y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-57 monofásico . . . . .	87
6.15. Proceso de convergencia. Caso de estudio IEEE-57 nodos trifásico . . . . .	89

## ÍNDICE DE TABLAS

---

6.16. Valores de <i>speedup</i> y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-57 trifásico . . . . .	89
6.17. Proceso de convergencia. Caso de estudio IEEE-118 nodos monofásico . . . . .	91
6.18. Valores de <i>speedup</i> y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-118 monofásico . . . . .	91
6.19. Proceso de convergencia. Caso de estudio IEEE-118 nodos trifásico . . . . .	92
6.20. Valores de <i>speedup</i> y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-14 trifásico . . . . .	92
A.1. Número de nodos y componentes del sistema eléctrico de 3 nodos. . . . .	105
A.2. Datos de las líneas de transmisión del sistema eléctrico de 3 nodos. . . . .	105
A.3. Datos de los bancos de capacitores del sistema eléctrico de 3 nodos. . . . .	105
A.4. Datos de los generadores del sistema eléctrico de 3 nodos. . . . .	106
A.5. Datos de las ramas magnetizantes del sistema eléctrico de 3 nodos. . . . .	106
A.6. Número de nodos y componentes del sistema eléctrico IEEE-14 nodos . . . . .	106
A.7. Datos de las líneas de transmisión del sistema eléctrico IEEE-14 nodos . . . . .	106
A.8. Datos de los bancos de capacitores del sistema eléctrico IEEE-14 nodos . . . . .	107
A.9. Datos de los generadores del sistema eléctrico IEEE-14 nodos . . . . .	107
A.10. Datos de las ramas magnetizantes del sistema eléctrico IEEE-14 nodos . . . . .	107
A.11. Datos de los STATCOM's del sistema eléctrico IEEE-14 nodos . . . . .	107
A.12. Número de nodos y componentes del sistema eléctrico IEEE-30 nodos . . . . .	108
A.13. Datos de las líneas de transmisión del sistema eléctrico IEEE-30 nodos . . . . .	108
A.14. Datos de los bancos de capacitores del sistema eléctrico IEEE-30 nodos . . . . .	108
A.15. Datos de los generadores del sistema eléctrico IEEE-30 nodos . . . . .	109
A.16. Datos de los hornos de arco eléctrico del sistema IEEE-30 nodos . . . . .	109
A.17. Número de nodos y componentes del sistema eléctrico IEEE-57 nodos . . . . .	110
A.18. Datos de las líneas de transmisión del sistema eléctrico IEEE-57 nodos . . . . .	110
A.19. Datos de los bancos de capacitores del sistema eléctrico IEEE-57 nodos . . . . .	111
A.20. Datos de los generadores del sistema eléctrico IEEE-57 nodos . . . . .	112
A.21. Datos de las ramas magnetizantes del sistema eléctrico IEEE-57 nodos . . . . .	112
C.1. Valores utilizados en el modelo de la rama magnetizante en ATP . . . . .	117

# Lista de Símbolos y Abreviaturas

Amp	Amperes
$v$	Voltaje
$i$	Corriente
$\lambda$	Flujo magnético
$R$	Resistencia
$L$	Inductancia
$C$	Capacitancia
$\phi$	Ángulo de desfaseamiento
$J(t)$	Jacobiano
$\Delta$	Incremento
$\Phi$	Matriz de identificación
$t$	Tiempo
$h$	Paso de integración
$\Sigma$	Sumatoria
$r$	Radio del horno eléctrico
$\eta$	Agudeza de la rodilla de saturación
$T$	Periodo
$\frac{d}{dt}$	Derivada con respecto al tiempo
$e$	Exponencial
$\infty$	Infinito
C/C++	Lenguaje de programación C y C más más
$\rho$	Plano de Poincaré
PVM	<i>Parallel Virtual Machine</i>
MPI	<i>Message Passing Interface</i>
API	<i>Application Programming Interface</i>
EMTP	<i>ElectroMagnetic Transients Program</i>
ATP	<i>Alternative Transients Program</i>



$D$	Derivada
$S$	Función de conmutación
$tri$	Forma Triangular
SEP	Sistema Eléctrico de Potencia
EEP	Estado Estacionario Periódico
FB	Fuerza Bruta
DN	Diferenciación Numérica
AD	Aproximación Directa
ME	Matriz Exponencial
RK	Runge Kutta
EDO	Ecuaciones Diferenciales Ordinarias
LVK	Ley de voltajes de Kirchhoff
LCK	Ley de corrientes de Kirchhoff
PWM	Modulación de ancho de pulso
$\dot{x}$	Vector de estado $x$
$x$	Variable de estado
$v_s$	Voltaje en la fuente
$v_c$	Voltaje en el capacitor
$x_0$	Vector de estados en el tiempo cero
$M$	Valor del voltaje pico
FFT	Transformada rápida de Fourier
RTC	Reactor controlado por tiristores
CSCT	Compensador serie controlado por tiristores
STATCOM	Compensador estatico de VARs
VARs	Volts-Amper reactivos
$\alpha$	Ángulo de disparo de los tiristores
$f$	Frecuencia
$p.u.$	Por unidad
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
$p$	Pendiente ponderada
R	Espacio vectorial de dimensión $n$
seg	Segundos

# Lista de Publicaciones

1. *Analysis of Electrical Networks Using Fine-Grained Techniques of Parallel Processing Based on OpenMP*, Omar Antonio Rico-Hernández y Antonio Ramos Paz, Artículo aceptado para su presentación la *8th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2011)*, celebrada en la ciudad de Mérida, Yucatán, México, del 26 al 28 de Octubre del 2011. IEEE Catalog Number: CFP11827-ART, ISBN: 978-1-4577-1013-1, 978-1-4577-1013-1/11/\$26.00 ©2011 IEEE
2. *Using the OpenMP Platform for Parallel Processing of Single and Three Phase Electrical Networks*, Omar Antonio Rico-Hernández y Antonio Ramos Paz, Artículo aceptado para su presentación en la *XIII Reunión de Otoño de Potencia, Electrónica y Computación, ROPEC'2011 INTERNACIONAL*, celebrada en la Ciudad de Morelia, Michoacán, México, del 9 al 11 de Noviembre del 2011. ISBN: 978-607-95476-3-9.

# Capítulo 1

## Introducción

### 1.1. Introducción

Hoy en día el análisis de grandes redes eléctricas se ha convertido en un problema en el cual se tiene que optimizar recursos con el fin de realizar estudios de una manera rápida y precisa, debido a que se han incorporado modelos más detallados de los diversos componentes del sistema eléctrico, así como la incorporación de modelos de nuevos dispositivos. Debido a estos cambios se ha incrementado la complejidad del análisis de los Sistemas Eléctricos de Potencia (SEP), los cuales tienen como base de su estudio la aplicación de las leyes de voltajes y corrientes de Kirchhoff, donde la solución del sistema se obtiene resolviendo un conjunto de Ecuaciones Diferenciales Ordinarias (EDO) las cuales modelan la dinámica de la red eléctrica. La idea de aplicar técnicas de procesamiento en paralelo en el análisis de un SEP no es algo nuevo, ya que anteriormente se han realizado diferentes trabajos incorporando estas técnicas, donde además se han utilizado diferentes plataformas de procesamiento en paralelo tales como, *Paralle Virtual Machine* (PVM) [Quinn, 2004], *Message Passing Interface* (MPI) [Quinn, 2004] y *threads* [Nichols *et al.*, 1996]. Sin embargo, el objetivo principal de hacer uso del procesamiento en paralelo es disminuir el tiempo de cómputo requerido para las simulaciones del comportamiento de los SEP.

En los últimos años se han establecido diferentes formas de realizar procesamiento en paralelo en lo que se conoce como multicomputadoras, que se refiere a tener varios equipos de cómputo con las mismas características o diferentes, interconectados a través de una red de comunicación que permita la transferencia de datos entre ellos con el objeto de realizar una o varias tareas en conjunto; y también lo que se conoce como procesamiento en paralelo multiprocesador donde computadoras con varios elementos de proceso tales como procesadores, núcleos, etc., pueden estar resolviendo de manera simultánea un problema. Estas técnicas requieren que

el programador plantee el problema a resolver e identificar las zonas susceptibles a ser paralizadas estableciendo los requerimientos de su programación, además de identificar la plataforma de procesamiento en paralelo que más le convenga.

## 1.2. Simulación de Circuitos Eléctricos

La simulación computacional de los SEP ha ayudado a reducir el tiempo y costo económico en su diseño, planeación y operación, gracias a que el desempeño futuro de la red puede ser predicho y corregido en la etapa de diseño a través simulaciones. Sin embargo, en muchos de los casos el *software* disponible para la simulación de fenómenos transitorios. La principal dificultad en la simulación de los SEP es calcular las cantidades de estado estable para posteriormente utilizarse estas cantidades en estudios tales como calidad de la energía, flujos de potencia monofásicos, trifásicos, etc. [Kundert *et al.*, 1990].

Los programas de simulación de circuitos comúnmente utilizados tales como SPICE (*Simulation Program with Integrated Circuit Emphasis*) [Kundert *et al.*, 1990], EMTP (*Electro Magnetic Transients Program*) [Liu y Meyer, 1987]. y ASTAP (*Advanced Statistical Analysis Program*) [Liu y Meyer, 1987], dependen de algoritmos de integración numérica que son adecuados para el análisis transitorio de circuitos no lineales, pero a menudo estos algoritmos no pueden procesar eficiente y acertadamente los valores de estado estable que son de interés para un diseñador de SEP, o no pueden simular eficientemente SEP que contengan parámetros distribuidos [Kundert *et al.*, 1990]. En estos programas de simulación los cálculos son desarrollados paso a paso para una serie de puntos en el tiempo con un tamaño de paso variable seleccionado por el programa dentro de límites específicos, una ecuación es establecida para cada nodo, donde la suma de las corriente se establece igual a cero, y para cada rama una ecuación da la relación lineal entre la corriente y los voltajes en cada extremo, inicio y fin. De esta manera el número de ecuaciones es igual al número de voltajes y corrientes desconocidas.

El programa EMTP permite modelar matemáticamente sistemas eléctricos, mecánicos y de control, monofásicos y polifásicos. EMTP fue desarrollado como contraparte del *software Transient Network Analyzer* (TNA) en los años finales de la década de 1960 por Hermann W. Dommel. Años más tarde él cedería los derechos de autor sobre el programa a la *Bonneville Power Administration* (BPA) de los Estados Unidos [Liu y Meyer, 1987].

El programa *Alternative Transients Program* (ATP) surge del año 1984 cuando los Drs. W. Scott Meyer y Tsu-huei Liu no aprobaron la comercialización del EMTP debido a intereses económicos por parte de DCG (*Development Coordination Group* de la BPA) y EPRI (*Electric Power Research Institute*) [Liu y Meyer, 1987]. Los Drs. Meyer y Liu empezaron el desarrollo

del ATP como una alternativa no comercializada del EMTP, pero basado en una copia de éste colocada en el dominio público. Aunque el programa puede ser adquirido sin costo, ATP no es totalmente del dominio público y se requiere de una licencia antes recibir material adicional por el interesado. Los requerimientos para usar el ATP son honestidad en su manejo y el compromiso de no participación en la comercialización de EMTP o de otros programas de simulación de transitorios. En términos generales, el programa es el mismo y suele mencionarse como EMTP, ATP o bien EMTP-ATP [Liu y Meyer, 1987].

### **1.2.1. Formación de las ecuaciones que modela la dinámica del circuito**

El comportamiento de la red eléctrica construida de resistores, capacitores, inductores y fuentes de corriente y voltaje, puede ser descrito por un conjunto de EDO que modelan la dinámica de los voltajes nodales, corrientes y cargas. Este sistema de ecuaciones es usado para representar la dinámica del sistema, y servirá para analizar su estabilidad y comportamiento futuro. La coherencia de estas soluciones es observada mediante la graficación de las variables del sistema a lo largo del eje del tiempo, a partir de condiciones iniciales arbitrarias que por lo regular es un vector de ceros. El sistema de ecuaciones puede ser construido de la descripción de la red usando análisis nodal o de mallas, que involucran la aplicación de la ley de corrientes y voltajes de *Kirchhoff* [Dorf, 1993].

Otra alternativa para el análisis de redes eléctricas de gran escala consiste en el uso de técnicas *Diakópticas* [Kron, 1963], a través de estas técnicas la red se divide en varias subredes, cada una de las cuales es analizada individualmente y la solución global se obtiene analizando las conexiones entre las diferentes subredes que forman a la red. Una ventaja de esta técnica es la posibilidad de ser paralelizada por el uso de un gran conjunto de matrices y vectores usados en el proceso de generación de las subredes.

Una vez modelado el SEP por medio de un conjunto de EDO, existe una gran variedad de métodos propuestos para calcular su comportamiento transitorio y en estado estable, estos métodos pueden ser clasificados en tres categorías principales: métodos en el dominio del tiempo, métodos en el dominio de la frecuencia y métodos híbridos [Kundert *et al.*, 1990].

### **1.2.2. Método de análisis en el dominio del tiempo**

Los métodos de análisis en el dominio del tiempo emplean métodos de integración numérica para resolver un conjunto EDO a lo largo del tiempo, con pequeños incrementos de tiempo  $\Delta t$  tan pequeños como se requiera para reproducir la dinámica del sistema. La ventaja de los métodos en el dominio del tiempo es que en lugar de resolver un conjunto de EDO, resuelven un

conjunto de ecuaciones algebraicas en cada paso de integración, por esta razón la solución no es exacta pero es muy útil cuando solo se desea conocer el valor numérico de la solución con fines prácticos donde se permite una tener una tolerancia para el error. Los problemas de esta forma que buscan una solución para un conjunto de ecuaciones diferenciales ordinarias que satisfaga restricciones de dos o más puntos en el tiempo, son llamados problemas de valores en la frontera [Kundert *et al.*, 1990].

### **1.2.3. Método análisis en el dominio de la frecuencia**

El método en el dominio de la frecuencia es la forma más común para calcular la solución de estado estable de forma directa, calculando directamente los coeficientes de la representación de la serie trigonométrica. Si el circuito es lineal y las fuentes de entrada son senos y cosenos, el cálculo directo con los coeficientes es equivalente al análisis fasorial [Kundert *et al.*, 1990]. En general la ventaja de trabajar directamente con los coeficientes es que las operaciones dinámicas lineales como diferenciación e integración, se vuelven simples operaciones algebraicas que pueden ser manipuladas con mucha más facilidad [Kundert *et al.*, 1990]. Esto implica que para analizar circuitos lineales invariantes en el tiempo, los coeficientes de la respuesta son fácilmente calculados.

Calcular los coeficientes de la respuesta de circuitos no lineales es más difícil que para sistemas lineales, además para una no linealidad arbitraria los coeficientes de la respuesta pueden no ser calculados directamente. Los métodos en el dominio de la frecuencia aplicados a circuitos no lineales donde el cálculo es obtenido directamente de los coeficientes de las series trigonométricas son referidos como métodos de balance armónico [Kundert *et al.*, 1990].

### **1.2.4. Método de análisis híbrido**

Los sistemas híbridos combinan las características principales de los métodos en el dominio de la frecuencia y los métodos en el dominio del tiempo. En [Usaola, 1990] se presenta un análisis híbrido en el cual la parte lineal de la red se analiza en el dominio de la frecuencia en tanto que la parte no lineal se trata en el dominio del tiempo. En [Semlyen y Medina, 1995] todas las partes no lineales incluidas en la red y parte de las cargas que son dependientes de la frecuencia, son representadas en el dominio de la frecuencia, mientras los componentes no lineales variantes en el tiempo, principalmente cargas son representados en el dominio del tiempo.

### 1.2.5. Técnicas de acercamiento rápido al estado estacionario periódico

Uno de los objetivos más importantes en el análisis de las redes eléctricas que incluyen componentes no lineales y variantes en el tiempo es la determinación del estado estacionario periódico (EEP). En [Chua y Ushida, 1981] los métodos de acercamiento rápido al EEP se clasifican en cuatro categorías: métodos de Fuerza Bruta (FB), métodos de perturbación, métodos de balance armónico y métodos de disparo:

**Métodos de fuerza bruta (FB)** Este método aproxima la solución por integración numérica del conjunto de EDO que modelan la dinámica del sistema, iniciando de una condición inicial elegida arbitrariamente hasta alcanzar el EEP, comúnmente la condición inicial es un vector de ceros trivial. Este método es bastante general y es bastante cuestionado ya que en caso de que el sistema sea pobremente amortiguado el esfuerzo computacional necesario para obtener el EEP será bastante considerable, además si la frecuencia base es demasiado grande le será difícil al método detectar cuando se haya alcanzado el EEP [Chua y Ushida, 1981].

**Método de perturbación** Este método aproxima la solución por iteración con la condición inicial a menudo tomada de un proceso previo de integración numérica para ser la solución de una ecuación linealizada. Desafortunadamente este método trabaja solo para circuitos donde las no linealidades son a menudo extremadamente pequeñas. Para circuitos que no dependan de no linealidades en una forma esencial, este método se vuelve altamente inadecuado, siendo solo el factor que la iteración a menudo no converge [Chua y Ushida, 1981].

**Métodos de balance armónico** Este método obtiene resultados por medio de una aproximación a la solución de una serie trigonométrica finita y donde el balance del conjunto de términos tendrá componentes de frecuencia idénticos. Aunque hay mucha teoría interesante acerca de estos métodos, este es a menudo costoso en tiempo, debido a que los diferentes componentes de frecuencia son estimados por análisis de Fourier multidimensional [Chua y Ushida, 1981].

**Métodos de disparo** Es una de las principales técnicas utilizadas para la determinación del EEP de circuitos eléctricos mediante extrapolaciones. El objetivo de estos métodos es encontrar una condición inicial del vector  $x_0$  tal que cuando se integre el sistema de EDO sobre un periodo de tiempo  $T$ , a partir de la condición inicial  $x_0$  se obtenga  $x(T) = x_0$ . Uno de los primeros trabajos tendientes a encontrar el vector  $x_0$  fue presentado en [Aprilie y Trick, 1972], en donde se resuelve este problema utilizando un método Newton-Raphson. En [Semlyen y Medina, 1995] se proponen tres métodos Newton de extrapola-

ción al ciclo límite basado en el concepto del plano de Poincaré. Los métodos propuestos en este trabajo son: Diferenciación Numérica (DN), Aproximación Directa (AD) y Matriz Exponencial (ME). En trabajos más recientes se aplican las técnicas de acercamiento rápido de las variables al ciclo límite en el análisis de redes trifásicas y monofásicas de gran escala con componentes no lineales y variantes en el tiempo como en el caso del trabajo [García y Acha, 2004] donde se presenta la aplicación de estas técnicas en el análisis de redes eléctricas trifásicas para estudios de armónicos de calidad de la energía.

### 1.3. Procesamiento en paralelo

Actualmente en el área de procesamiento computacional, en algunos laboratorios de investigación, se obtienen diodos de tunelamiento en nanoestructuras semiconductoras de  $3nm^2$ , y la posibilidad de integración en este tipo de dispositivos en los próximos años aún se encuentra en fase de desarrollo, cuyos frutos desde el punto de vista comercial, probablemente no se vean en los próximos diez años. El aumento del desempeño de los procesadores en los últimos 20 años, estuvo basado en la reducción de los componentes y la integración de un número mayor de dispositivos en un volumen cada vez más reducido. Pero esto tiene un límite, ya que los átomos de silicio, galio y arsénico tienen un tamaño definido en su forma cristalina (aproximadamente de 0.5 nm) y ese límite, cada año, viene siendo alcanzado cada vez que se van desarrollando nuevas técnicas para la integración de microcircuitos en una área cada vez más reducida [Quinn, 2004].

Frente a este panorama desolador observado hace más de diez años atrás y frente a la creciente demanda de contar con sistemas de cómputo para la realización de intensos cálculos numéricos que permitan modelar y predecir eventos en áreas como la meteorología, la vulcanología, geología, diseño de dispositivos, fenómenos no lineales, así como el análisis de los SEP, surgieron condiciones para el procesamiento en paralelo usando los procesadores que actualmente se encuentran en el mercado. La primera de ellas fue de las empresas fabricantes de supercomputadoras como la *nCUBE*, *Cray* y *Fujitsu*. El sistema consistía en tener  $N$  procesadores haciendo cálculos independientes y coordinados, compartiendo una memoria física de varios Gigabytes de memoria RAM, la capacidad de procesamiento dependía del número de procesadores.

Dentro de este contexto el primer protocolo de comunicación libre fue llamado PVM, que aparece a finales de los 80's como un estándar de desarrollo para sistemas de memoria distribuida. Esto significa que, estaciones de trabajo de diferente fabricante pueden comunicarse a través de una red de trabajo compartida, y no solo los procesadores en un ambiente paralelo, sino también la memoria. A partir de 1992 se trabaja en un protocolo mejorado llamado MPI, que es el actual protocolo estándar de procesamiento en paralelo con multicomputadoras.



También aparecieron las tarjetas madre con dos, cuatro y más procesadores, lo que ha hecho que las nuevas computadoras se vuelvan más y más poderosas introduciendo nuevos conceptos como *Multithreading* visto como un camino para desempeñar muchas tareas de un programa con buena eficiencia y velocidad, y se discute la conveniencia de escribir programas divididos en múltiples tareas, ahora aparece el modelo de *threads* que toma procesos y los divide en múltiples tareas, donde los procesos padres se segmentan en procesos hijos para desempeñar subtareas. En el modelo de *Threads*, múltiples tareas deben existir como un flujo individual de control dentro del mismo proceso, donde el tamaño de las tareas (en términos de instrucciones) es llamado *granularidad* y existe la posibilidad de elegir tareas de diferentes tamaños [Quinn 2004].

Recientemente surge un nuevo API (*Application Programming Interface*) el *OpenMP* (*Open Multi – Processing*) que es un estándar para programación en paralelo en memoria compartida multiprocesadores. OpenMP no es un nuevo lenguaje, sino que extiende los lenguajes existentes tales como *Fortran* y *C/C++* con un conjunto de directivas. En contraste a previos API tales como *Threads* y *MPI*, *OpenMP* da una aproximación incremental para la paralelización de programas secuenciales, ya que el programador puede añadir directivas para paralelizar ciclos y sentencias en el programa secuencial. Esto abrió la posibilidad de realizar procesamiento en paralelo a un mayor número de usuarios alrededor del mundo, especialmente en países como el nuestro con pocos recursos para invertir en sistemas de cómputo sofisticados [Rauber y Runger, 2010].

El procesamiento en paralelo aplicado en el análisis de los SEP es algo que se ha desarrollado por varios años, por ejemplo en [Stavrakakis *et al.*, 1990] se presenta un algoritmo en paralelo para la detección de fallas en manejadores de motores de corriente directa. En 1993 en [Werler y Glavitsch, 1993] se presenta un análisis de transitorios electromagnéticos mediante el uso del procesamiento en paralelo. En 1994 en [Mariños *et al.*, 1994] se hace una aplicación del procesamiento en paralelo al análisis de armónicos en sistemas eléctricos de potencia. En el 2002 se presenta en [García *et al.*, 2001] la aplicación de técnicas de procesamiento en paralelo a las técnicas Newton de acercamiento rápido al ciclo límite de las variables de estado, utilizando la plataforma de *Multithreading*.

En artículos más recientes como en [García y Acha, 2004] se presentan las técnicas de acercamiento rápido al ciclo límite de las variables de estado al análisis del sistema de prueba IEEE-118 nodos trifásico modificado, donde se utilizaron técnicas de procesamiento en paralelo basadas en *Multithreading* con el objetivo de incrementar la eficiencia computacional en el proceso de solución. En el 2006 se presenta en [Medina *et al.*, 2006] un análisis de la red monofásica IEEE-118 nodos con componentes no lineales y variantes en el tiempo, donde se aplican técnicas de acercamiento rápido al ciclo límite y técnicas de procesamiento en paralelo basadas en

PVM y *Multithreading*. En el 2005 en [Medina y Ramos, 2005] se hace una aplicación de las técnicas de procesamiento en paralelo basadas en dos plataformas de procesamiento, PVM y *Multithreading*, en el análisis de la respuesta a la frecuencia de redes eléctricas.

## 1.4. Justificación

En base a lo que se ha expuesto en la revisión del estado del arte, donde se han revisado diversos trabajos de investigación en relación a la simulación de sistemas eléctricos de potencia, generación del conjunto de EDO que modelan la dinámica de un SEP, técnicas para la obtención del estado estacionario periódico, procesamiento en paralelo y plataformas de procesamiento en paralelo, en los cuales se conjuntan varias de ellas, se ha llegado a la conclusión de que hasta ahora no se ha realizado algún trabajo donde se haga la paralelización de los métodos de integración numérica con una granularidad muy fina. Específicamente, en base a los trabajos que se han realizado en el área del procesamiento en paralelo aplicados a los SEP, en esta tesis se utiliza una plataforma de procesamiento en paralelo relativamente nueva pero que ha llegado a consolidarse, como lo es *OpenMP*, debido a los grandes avances tecnológicos donde las nuevas generaciones de computadoras cuentan con un mayor número de núcleos disponibles para el procesamiento en paralelo y la posibilidad de realizar una aplicación estándar que pueda ser portable a equipos de cómputo con características de *hardware* diferentes, utilizando siempre el máximo número de núcleos que se tengan disponibles para realizar el procesamiento en paralelo, esto con la finalidad de reducir el tiempo de análisis de grandes redes que cuentan con elementos de electrónica de potencia en sus modelos, los cuales representan una gran fuente de contenido armónico inyectándose a la red y aumentando el esfuerzo computacional para la obtención del EEP.

## 1.5. Objetivos

- Incorporar técnicas de procesamiento en paralelo con granularidad fina basadas en la plataforma OpenMP para el análisis de redes eléctricas monofásicas y trifásicas de gran escala que cuenten con elementos lineales, no lineales y variantes en el tiempo, mediante el método de FB.
- Realizar un estudio detallado del método de integración de *Runge – Kutta* de cuarto orden, utilizado para la integración numérica del conjunto de EDO, con la finalidad de realizar su programación paralela con granularidad fina.

- Obtener la dinámica del comportamiento transitorio y el tiempo necesario para obtener el EEP al utilizar el procesamiento en paralelo con un número diferente de elementos de proceso, además de realizar un análisis del contenido armónico presente en la red debido a la incorporación de elementos no lineales.
- Incorporar modelos de dispositivos FACTS para ser incluidos en los diferentes casos de estudio y mostrar el contenido armónico en la red al tenerlos activos.
- Realizar una comparación de los tiempos necesarios para obtener el EEP con un número diferente de elementos de procesamiento en términos del speedup así como observar el comportamiento al incrementar las dimensiones del problema.

## 1.6. Metodología

Se hará una representación en espacio de estado de la red eléctrica, con la finalidad de que la solución del conjunto de EDO se reduzca a la evaluación repetida de un conjunto de ecuaciones algebraicas, lo cual es la parte principal de los métodos de integración convencionales, además de especificar casos especiales de evaluación para cuando se presenten elementos no lineales en la red.

Con el fin de presentar esta metodología se hará la implementación del método de integración de *Runge – Kutta* de 4to orden en el lenguaje de programación *C/C++* de acuerdo a la lógica de programación adoptada para maximizar las funciones de multiplicación matricial, estas funciones nos permitirán obtener el máximo provecho al utilizar el procesamiento en paralelo.

Ya que se tenga la programación de las funciones de multiplicación dentro del método de integración de *Runge – Kutta* de 4to orden en forma secuencial, se le incorporarán técnicas de procesamiento en paralelo basadas en la plataforma *OpenMP* y se probará la herramienta desarrollada en la solución de diversos casos de estudio que cuenten con diferentes características de tamaño y complejidad, para hacer una evaluación de la eficiencia ante diferentes casos de estudio, utilizando diferente número de unidades de procesamiento. La eficiencia se medirá en términos del speedup obtenido.

Adicionalmente se hace una comparación de los resultados obtenidos con la aplicación desarrollada y el software de simulación ATP en los casos más pequeños con el fin de validar resultados.

Finalmente en base a los resultados obtenidos en los diferentes casos de estudio, se presentarán conclusiones y recomendaciones para trabajos futuros.

## 1.7. Descripción de los Capítulos

En el Capítulo 1 se justifica esta tesis, indicando la importancia de utilizar las nuevas herramientas de procesamiento en paralelo para el análisis de redes eléctricas en el dominio del tiempo.

En el Capítulo 2 se hace un análisis de los métodos utilizados para la obtención del estado transitorio y del estado estacionario periódico en los sistemas eléctricos de potencia.

En el Capítulo 3 se describe lo que es el procesamiento en paralelo basado en Multithreading, así como su arquitectura y las plataformas que existen para implantarlo en el análisis de redes eléctricas de gran escala.

En el Capítulo 4 se hace una descripción detallada del procedimiento seguido para lograr la paralelización a un nivel de grano fino del método de integración de Runge-Kutta utilizando la plataforma OpenMP de procesamiento en paralelo.

En el Capítulo 5 se se presentan los modelos matemáticos para cada uno de los elementos utilizados en los diferentes casos de estudio, presentando una breve descripción de la deducción matemática en los elementos más complejos.

En el Capítulo 6 se presentan el análisis monofásico y trifásico de cinco casos de estudio y se da una breve descripción de los resultados obtenidos en términos del *speedup*, así como de su contenido armónico. En los casos más completos se presentan las formas de onda en EEP y su respectivo contenido armónico y para los casos más compactos se realiza una validación con el programa de simulación ATP.

En el Capítulo 7 se presentan las conclusiones generales obtenidas en la presente tesis y se hacen recomendaciones para trabajos futuros.

## Capítulo 2

# Metodologías para la Obtención del Estado Transitorio y Estado Estacionario Periódico en el Dominio del Tiempo

### 2.1. Introducción

Un transitorio eléctrico es la manifestación externa de un cambio repentino en las condiciones del sistema, el cual se puede dar por la operación de conmutación de un interruptor o por una falla en el sistema. El periodo transitorio usualmente es muy corto. La fracción de tiempo de operación que emplean los circuitos en las condiciones transitorias son insignificantes comparada con el tiempo en estado estable. Sin embargo estos periodos son extremadamente importantes, ya que en este periodo los componentes del sistema son sujetos a un gran estrés debido a excesivas corrientes o altos voltajes. Esto puede deshabilitar equipos, cerrar una planta o dejar sin energía eléctrica a una ciudad entera dependiendo del circuito afectado. Por esta razón, una apreciación clara de los eventos que toman lugar durante los periodos transitorios es esencial para un completo entendimiento del comportamiento de los sistemas eléctricos.

Aunque muchos programas de simulación se enfocan en el análisis transitorio, también es de interés típico para los diseñadores el comportamiento en estado estable de los circuitos eléctricos. Esto es porque ciertos aspectos del desempeño son fácilmente caracterizados y verificados en estado estable. Cantidades como distorsión armónica, potencia, frecuencia, ruido y características de transferencia como ganancia e impedancia, son cantidades que pueden ser medidas con mucha más facilidad cuando el sistema se encuentra en estado estable. Una ecuación diferencial puede no tener una solución de estado estable o puede tener cualquier número de soluciones y dependerá de la condición inicial [Kundert *et al.*, 1990]. En particular para cada solución de

estado estable, tiene que corresponder una región de atracción, para que si la condición inicial está contenida en la región asociada, entonces la solución se aproxima la solución de estado dada.

Hay diferentes tipos de comportamiento de estado estable que son de interés. El primero es el estado estable de Corriente Directa (CD). Aquí la solución es un punto de equilibrio del circuito y no varía con el tiempo. Circuitos lineales asintóticamente estables alimentados por fuentes senoidales, eventualmente exhiben una solución de estado estable senoidal, que es caracterizada como puramente senoidal excepto para alguna posible compensación de CD. Si la respuesta de estado estable de un circuito consiste únicamente de una combinación de una compensación de CD y un número infinito de senoides, se dice que el circuito está en estado estable periódico. El periodo de la solución es usualmente igual que el periodo de la entrada, si este existe, aunque ocasionalmente el periodo de los dos resultará del mismo periodo común. Si un circuito no lineal es alimentado con varias fuentes periódicas con frecuencias no relacionadas, el circuito típicamente tendrá una respuesta de estado estacionario periódica. Una respuesta periódica consiste de una combinación de senoides como la suma o diferencia de frecuencias de un conjunto finito de frecuencias fundamentales y sus armónicas. La frecuencia fundamental es usualmente igual que la señal de entrada, aunque algunas veces son múltiplos pares. También puede resultar una frecuencia fundamental de una auto oscilación en lugar de un estímulo. El estado estable cuasiperiódico puede incluir al estado estable periódico como un caso especial [Kundert *et al.*, 1990].

## 2.2. Ecuaciones Diferenciales Ordinarias

El tratamiento de cualquier problema de análisis transitorio o de estado estable propiamente comienza con el establecimiento de la ecuación diferencial o conjunto de ecuaciones diferenciales que describen el comportamiento del sistema. Por ejemplo, se pueden aplicar leyes de voltajes y corrientes de Kirchhoff para sistemas eléctricos o leyes de Newton para sistemas mecánicos.

La aparición de las ecuaciones diferenciales está ligada a la del cálculo diferencial y se asocia con personajes de gran renombre como Newton y Leibniz, e incluso Newton en el año de 1671 en sus trabajos ya consideraba varios ejemplos de ecuaciones diferenciales de primer orden, pero los estudios más relevantes de Newton en relación a las EDO fueron en su artículo *Principia Mathematica* publicado en el año 1687. En este artículo Newton introdujo el modelo gravitatorio y planteó las ecuaciones diferenciales que han servido de base para todos los estudios de mecánica celeste.

Tras varios siglos de estudio de las ecuaciones diferenciales han resultado ser una fuente

inagotable de problemas en el campo de las matemáticas y por otro lado una herramienta indispensable para modelar los más variados fenómenos en la naturaleza, por lo que hoy en día los modelos matemáticos de fenómenos de cualquier disciplina en la investigación que son desarrollados se resuelven por medio de teorías bien fundamentadas y sofisticados recursos matemáticos, además del desarrollo de manipuladores algebraicos como MATLAB [MathWorks, 2009], MAPLE [Carrillo y Centeno, 2010], Mathematica [Wolfram, 2011], etc., que ayudan a modelar fenómenos muy complejos.

Las ecuaciones diferenciales se suelen clasificar en ecuaciones diferenciales ordinarias y ecuaciones diferenciales parciales (EDP). En este contexto las incógnitas o variables de las EDO dependen de una sola variable independiente y en las EDP la variables de penden de múltiples variables.

En general, las ecuaciones diferenciales tienen un número infinito de soluciones y es necesario establecer restricciones en la solución hasta que solo quede una [Zill y Cullen, 2005]. Si las restricciones son ubicadas en el mismo punto en el tiempo, la combinación de las ecuaciones diferenciales y las ecuaciones de restricción son llamadas un problema de valor inicial debido a que las restricciones son ubicadas normalmente en el comienzo del intervalo de interés y la ecuación diferencial integrada con un incremento  $t$ . Es posible poner las restricciones al final del intervalo de interés. Tal problema es referido como problema de valor final y es tratado idénticamente al problema de valor inicial, excepto que la ecuación diferencial es integrada con un decremento  $t$ . También es posible ubicar la restricción en un punto dentro del intervalo de interés y dividir el problema en dos problemas independientes, un problema de valor inicial y otro de valor final [Zill y Cullen, 2005]. La descripción general para el conjunto de EDO tiene la forma general,

$$\dot{x} = f(x, t) \quad (2.1)$$

Donde  $\dot{x}$  es el vector de estados de  $n$  variables dependientes y  $t$  es la variable independiente (tiempo). El objetivo es resolver  $\dot{x}$  como una función de  $t$ , con valores iniciales de  $x$  y  $t$ .

### 2.3. Métodos de integración numérica

Los métodos de integración numérica son utilizados para integrar y obtener la solución aproximada de funciones dadas, mediante una serie de puntos o de forma analítica, ya que aun en el caso donde sea posible la integración analítica, la integración numérica puede ahorrar tiempo y esfuerzo si sólo se desea conocer el valor numérico y evitar entrar manipulación de las ecuaciones diferenciales, sin embargo salvo en algunos problemas muy concretos en los que es posible



la integración analítica de forma simple.

Los métodos de integración numérica son obtenidos por integración de un polinomio de interpolación. Por lo que distintas fórmulas de interpolación darán por resultado distintos métodos de integración numérica [Nakamura, 1992] de los cuales los métodos más conocidos y empleados son, el método de Euler, la Regla Trapezoidal y los métodos de Runge-Kutta, donde la metodología generalizada para resolver un conjunto de EDO está dada por(2.1).

$$\text{Valornuevo} = \text{Valor anterior} + \text{Tamaño del paso} \times \text{Pendiente} \quad (2.2)$$

La diferencia entre los métodos convencionales de integración recae en la manera que se calcula la pendiente en cada uno de ellos. En [Hornbeck, 1975] estos métodos numéricos se clasifican como:

**Métodos explícitos** La solución en estos métodos depende exclusivamente del paso previo. Entre los principales métodos explícitos pueden mencionarse: El método de Euler y los métodos de Runge-Kutta.

**Métodos implícitos** Estos métodos se denominan A-estables (Absolutamente estables) debido a que la convergencia no depende directamente de la elección del paso de integración  $h$ , por tanto estos métodos son muy utilizados en el análisis de sistemas rígidos (stiff systems). El método de la Regla trapezoidal (Euler modificado) es una técnica implícita, donde es necesario utilizar un método de solución para encontrar una aproximación a  $f(t_{n+1}, x_{n+1})$ . Esta es una característica de todos los métodos implícitos. Una buena aproximación a  $f(t_{n+1}, x_{n+1})$  puede obtenerse por medio del método de Euler.

**Métodos predictor-corrector** Los también llamados métodos semi-implícitos se basan en dos procesos:

- Predictor: es donde se determina un valor inicial  $x_{n+1}^p$  (valor predicho) por medio de la utilización de un método explícito.
- Corrector: en este paso se obtiene  $x_{n+1}$  por medio de la aplicación de un método implícito, el cual utiliza el valor predicho por el predictor para evaluar  $f(x_{n+1}, t_{n+1})$ , de tal forma que no es necesario resolver de manera numérica para obtener  $x_{n+1}$ .

Estos métodos requieren mayor esfuerzo computacional ya que cada paso de integración se convierte en un proceso iterativo.



### 2.3.1. Regla Trapezoidal

La regla del trapecio o regla trapezoidal es una de las fórmulas cerradas de Newton-Cotes. La regla Trapezoidal estima dos pendientes en el intervalo, una al inicio y otra al final. Estas dos pendientes son promediadas con el fin de obtener una mejor aproximación, su expresión matemática es:

$$x_{n+1} = x_n + \frac{h}{2} \cdot [f(x_{n+1}, t_{n+1}) + f(x_n, t_n)] \quad (2.3)$$

Donde  $f(x_n, t_n)$  es la pendiente estimada al inicio del intervalo, y  $f(x_{n+1}, t_{n+1})$  es la pendiente al final y se puede aplicar a cualquier número de intervalos con una separación uniforme  $h$  como se muestra en el diagrama de flujo de la Figura 2.1.

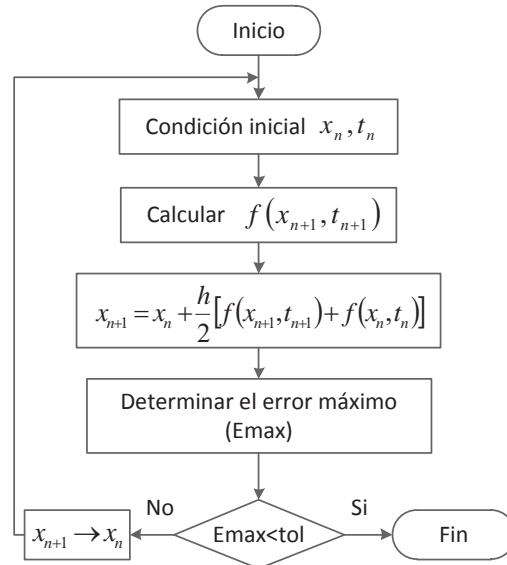


Figura 2.1: Método de la regla trapezoidal

### 2.3.2. Método de Euler

El método de Euler es el más simple de los métodos numéricos, este método utiliza la pendiente al inicio del intervalo como una aproximación a la pendiente promedio sobre todo el intervalo. La primera derivada proporciona una estimación directa de la pendiente en  $x_n$ , donde  $f(x_n, t_n)$  es la ecuación diferencial evaluada en  $x_n$  y  $t_n$ . Sustituyendo esta estimación en la ecuación (2.2), se tiene:

$$x_{n+1} = x_n + h \cdot f(x_n, t_n) \quad (2.4)$$

Esté método es adecuado para una programación rápida debido a su sencillez. De hecho, una gran parte de los métodos numéricos para las ecuaciones diferenciales parabólicas e hiperbólicas (que son mucho más complicadas que las EDO) se basan en los métodos de Euler y no en los métodos de Runge-Kutta. Su diagrama de flujo se muestra en la Figura 2.2.

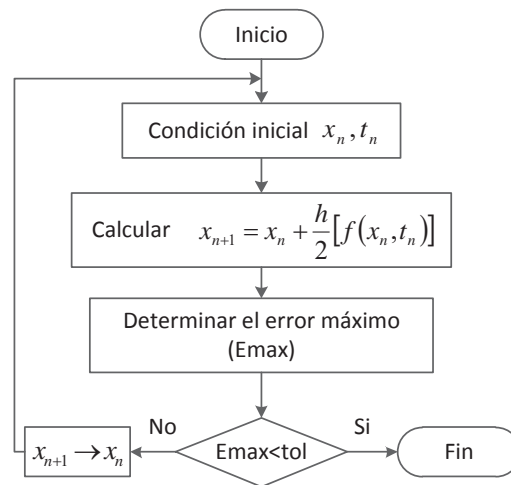


Figura 2.2: Método de Euler

## 2.4. Método de Runge-Kutta

Los métodos de Runge-Kutta se derivan a partir de la serie de Taylor. La forma general de la ecuación usada para formular el método de Runge-Kutta es:

$$y_{n+1} = y_n + \Delta y_n \quad (2.5)$$

$$\Delta y_n = P \cdot (t_n, y_n) \cdot h \quad (2.6)$$

donde  $\Delta y_n$  es la función incremental que puede interpretarse como la pendiente respectiva del intervalo, y en general:

$$P = a_1 \cdot k_1 + a_2 \cdot k_2 + \dots + a_n \cdot k_n \quad (2.7)$$

Por lo que se puede representar la fórmula del método de Runge-Kutta de 4to orden como:

$$\Delta y_n = \frac{\Delta t}{6} \cdot (K_1 + 2 \cdot K_2 + 2 \cdot K_3 + K_4)$$

Para representar mejor el método se presenta un diagrama de flujo en Figura 2.3.

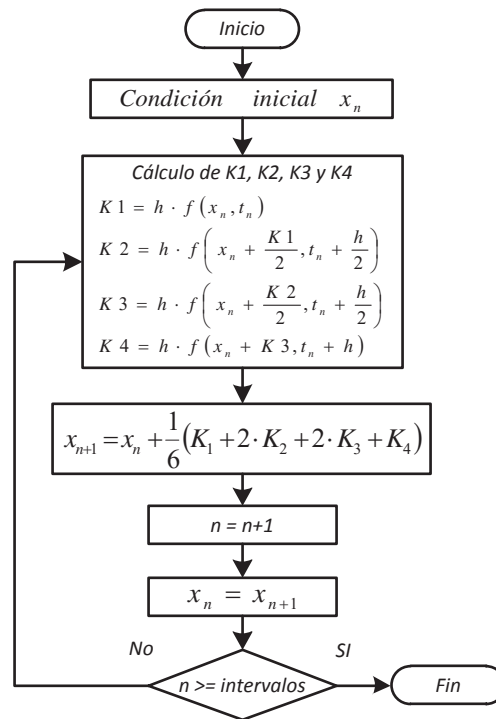


Figura 2.3: Método de Runge-Kutta de 4to orden

De la Figura 2.3 que para cada punto contenido en el intervalo de estudio se tiene que realizar el cálculo de cuatro constantes  $K$ s. El cálculo de cada una de ellas implica un proceso de evaluación de una función matricial de la forma  $AX + BU$ . Por esta razón es importante concentrar esfuerzos sobre una forma eficiente para el cálculo de estas  $K$ s.

## 2.5. Método de Fuerza Bruta (FB)

El método de FB es el método más simple para determinar el estado estable de una red eléctrica aplicando de forma repetida alguno de los métodos de integración al conjunto de EDO que modelan a la red. El proceso comienza con la elección de un vector de condiciones iniciales, entonces éste realizar la integración sobre un periodo de frecuencia. El método de integración más comúnmente utilizado es el método de Runge-Kutta de cuarto orden, con un paso de integración que depende de las necesidades del analista. Al término del periodo de integración se determina el error máximo entre el vector de condiciones iniciales (o vector de estados iniciales) y el vector solución de las variables de estado obtenido tras el periodo de integración. Si el error máximo entre el vector de condiciones iniciales y finales es mayor a un criterio de convergencia

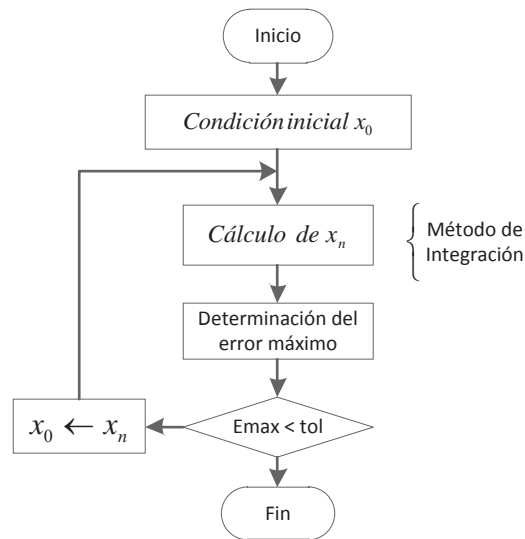


Figura 2.4: Método de fuerza bruta (FB)

establecido, que generalmente es de  $1 \times 10^{-10}$  p.u., entonces se realiza la sustitución del vector de condiciones iniciales por el vector solución, este proceso se repite hasta que el error máximo calculado sea menor al criterio de convergencia establecido. El número de veces que se tenga que aplicar este procedimiento depende de las características de la red. La Figura 2.4 muestra el diagrama de flujo del método de FB [Parker y Chua, 1989].

Este método de FB presenta varias ventajas y desventajas respecto a los métodos Newton. Como ventajas podemos decir que:

- Su programación es fácil
- El esfuerzo computacional es reducido al no realizar inversiones matriciales.
- Da una visión completa del comportamiento dinámico del sistema.
- Si la condición inicial se encuentra cercana al ciclo límite, el método siempre converge.

Así mismo debemos decir que el método de FB presenta las siguientes desventajas [Halpin, 2003]:

- La convergencia puede ser lenta o incluso podría nunca llegar cuando se trata de sistemas rígidos (*Stiff*), en los cuales el amortiguamiento es muy pobre.
- En el caso de los sistemas donde la frecuencia de oscilación es muy pequeña, el método de FB podría no detectar cuando ya se encuentre en el estado estable.

## **2.6. Conclusiones**

En este capítulo se ha presentado el concepto de lo que son las ecuaciones diferenciales y una generalización de los métodos que existen para resolver un sistemas de ecuaciones diferenciales ordinarias no lineales. Además se dio una explicación general de la metodología para el cálculo del estado estacionario periódico de redes eléctricas. Además se hizo una descripción de los métodos de integración numérica y se presentaron los diagramas de flujo de cada uno de ellos para facilitar la comprensión y la elección de alguno que cumpla las necesidades de cálculo en el presente trabajo.

# Capítulo 3

## Procesamiento en Paralelo

### 3.1. Introducción

Dentro del campo de la enseñanza y la investigación se tiene la necesidad de analizar sistemas cada vez más complejos, donde la obtención de resultados no es la única prioridad, sino que también el tiempo en que estos se obtienen. En la actualidad existen modelos experimentales en áreas como la genética que ha modelado el genoma humano, la aeronáutica que realiza modelados de aeronaves militares y comerciales así como en el área automotriz donde se realizan pruebas de colisión y por supuesto en el área de la ingeniería eléctrica donde se han modelado múltiples elementos en los SEP [Dongarra, 2003]. Las pruebas que se realizan en estas áreas tienen grandes costos económicos y en ocasiones restricciones físicas y éticas por lo que se recurre a modelos matemáticos. En estos modelos hay cálculos que pueden tardar horas, días e incluso semanas en realizarse y debemos pensar si podemos darnos el lujo de esperar ese tiempo. Por supuesto podríamos esperar a que las Unidades Centrales de Proceso (CPU por sus siglas en inglés) comerciales sean más veloces, lo que sería, que en alrededor de 5 años los CPUs serán tan solo unas cantas veces más rápidos que los actuales, ya que se está llegando a límites de velocidad físicamente posibles, donde para seguir el sendero de la obtención de mejores desempeños y ahorros económicos los fabricantes incorporan varios procesadores o núcleos en un solo CPU.

En países en vías de desarrollo como el nuestro donde las universidades no cuentan con recursos necesarios para la adquisición de equipo de cómputo sofisticado con características especiales para la realización de intensos cálculos numéricos, se presentan alternativas para la implementación del procesamiento, las cuales permiten utilizar el equipo de cómputo existente en las instituciones bajo alguna de las plataformas que existen para el procesamiento en paralelo.

Ya que en el área de los sistemas eléctricos de potencia se han hecho varios trabajos don-

de se ha aplicado el procesamiento en paralelo para la rápida solución de problemas que han sido susceptibles de ser paralelizados, y donde los resultados obtenidos en estos trabajos han sido buenos, se ha motivado a los investigadores para seguir realizando trabajos que empleen el procesamiento en paralelo con el fin de obtener resultados de forma más rápida.

Además las innovaciones en la estructura del *hardware*, como *Hyper – Threading* o procesadores multinúcleo, hacen disponibles los recursos de procesamiento en paralelo para computadoras de escritorio. Sin embargo, estas innovaciones requieren del uso de diferentes técnicas de procesamiento en paralelo. En pocos años, muchos productos de *software* estándar estarán basados en conceptos de procesamiento en paralelo para hacer un uso eficiente los recursos del *hardware* multinúcleo. Por lo tanto, la necesidad de la programación en paralelo será extendida a todas las áreas del desarrollo de *software*.

## 3.2. Procesamiento en paralelo

El procesamiento paralelo se define como el uso de múltiples elementos de proceso de forma simultánea con el fin de reducir el tiempo necesario para el cálculo de un problema computacional [Nichols *et al.*, 1996], véase Figura 3.1. El procesamiento en paralelo hoy en día es considerado como un camino estándar de ingenieros y científicos para resolver problemas en áreas tan diversas como la evolución galáctica, predicción del clima, diseño de aeronaves, dinámica molecular, entre otras. Para entender mejor el concepto, debemos entender que el cálculo paralelo requiere de un sistema de procesamiento multiprocesador que soporte procesamiento en paralelo, donde hay dos categorías importantes de procesamiento en paralelo: multicomputadoras y multiprocesadores [Quinn, 2004].

Como su nombre lo indica, un sistema multicomputadoras es un sistema de procesamiento en paralelo construido de múltiples computadoras y una red de interconexión, donde los procesos en las diferentes computadoras interactúan mediante paso de mensajes unos con otros.

En contraste un sistema multiprocesadores es un sistema integrado más ligero, en el que todos los CPU comparten acceso a una sola memoria global. Esta memoria compartida soporta comunicación y sincronización entre procesos, además el lenguaje de programación en paralelo permite indicar explícitamente como diferentes partes del proceso pueden ser ejecutadas al mismo tiempo por diferentes procesadores [Nichols *et al.*, 1996].

El primer paso en la programación en paralelo es el diseño de un algoritmo o programa para un problema de aplicación dado. El diseño comienza con la descomposición de los cálculos de una aplicación en varias partes, llamadas tareas, que pueden ser procesadas en los núcleos o procesadores del *hardware*. La descomposición de tareas puede ser complicada y laboriosa

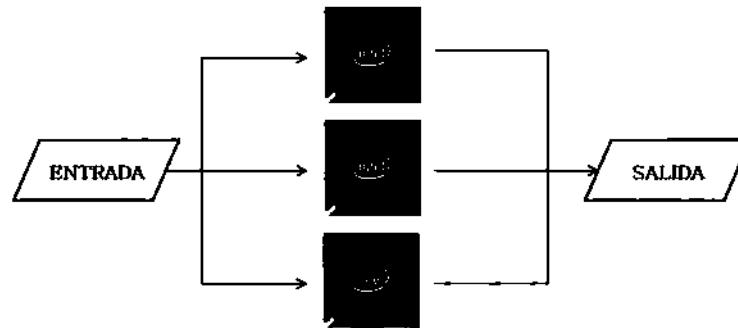


Figura 3.1: Procesamiento en paralelo

ya que usualmente hay diferentes posibilidades de descomposición para el mismo algoritmo de aplicación. El tamaño de las tareas (en términos de instrucciones) es llamado granularidad y existe la posibilidad de elegir tareas de diferentes tamaños. Definir apropiadamente las tareas de una aplicación es uno de los principales trabajos intelectuales en el desarrollo de un programa paralelo y es difícil de automatizar, y pudiendo definir como paralelismo potencial a la propiedad inherente de un algoritmo de aplicación e influye en como una aplicación puede ser dividida en tareas.

Las tareas de una aplicación son códigos en un lenguaje de programación o ambiente paralelo y son asignados a procesos o *threads* que son asignados a unidades físicas de proceso para su ejecución. La asignación de tareas a procesos o *threads* es llamada programación paralela y fija el orden en que las tareas son ejecutadas. La programación paralela puede ser hecha a mano en el código fuente o por el ambiente de programación, al momento de la compilación o dinámicamente en la ejecución [Nichols *et al.*, 1996].

La asignación de procesos o *threads* en las unidades físicas, procesadores o núcleos, es llamado asignación (*mapping*) y es usualmente hecha por el sistema de ejecución pero puede ser influenciado por el programador. La sincronización de las tareas en un algoritmo de aplicación puede ser implícita o explícita resultando en dependencia de datos o control de tareas. La dependencia de datos o control pueden requerir de un orden de ejecución específico de las tareas, es decir, si una tarea necesita datos producidos por otra tarea, entonces la ejecución de la primera tarea puede iniciar después de que la otra tarea genere ese dato y proporcione la información. Por lo tanto, la dependencia de tareas es una limitante para la programación paralela. Además, los programas paralelos necesitan sincronización y coordinación de *threads* y procesos para ejecutarse correctamente. Los métodos de sincronización y coordinación son intercambiados entre procesos o *threads*, esto depende de la organización del *hardware* [Quinn, 2004].

Una amplia clasificación de la organización de memoria distingue entre máquinas de memoria compartida y máquinas de memoria distribuida. A menudo el término *thread* es relacionado



con memoria compartida y el termino proceso es relacionado con memoria distribuida. Para máquinas con memoria compartida, una memoria global almacena los datos de una aplicación y todos los procesadores o núcleos del *hardware* pueden acceder a ella. El intercambio de información entre *threads* es hecho por variables compartidas escritas por un *thread* y leído por otro *thread*. El comportamiento correcto del programa entero tiene que ser logrado por sincronización entre *threads* de modo que se coordine el acceso a datos compartidos. Un *thread* lee un dato, pero no antes de que la operación de escritura por otro *thread* haya finalizado su almacenamiento del elemento dato. Dependiendo del lenguaje de programación o entorno, la sincronización es hecha el por sistema de ejecución o por el programador. Para máquinas de memoria compartida, existe una memoria privada para cada procesador, la cual puede ser accesada solamente por ese procesador y no es necesaria la sincronización para el acceso a memoria. El intercambio de información se hace por el envío de datos de un procesador a otro, a través de una red de interconexión por operaciones de comunicación explícitas [Rauber y Runger, 2010].

Las operaciones específicas *barrier*, a menudo son otras formas de condición que están disponibles para memoria compartida y memoria distribuida. Todos los procesos o *threads* tienen que esperar en un punto de sincronización *barrier* hasta que todos los demás procesos o *threads* hayan alcanzado ese punto. Solo después de que todos los procesos o *threads* han ejecutado el código antes del *barrier*, entonces ellos pueden continuar trabajando con el código subsecuente después del *barrier* [Nichols *et al.*, 1996].

Un aspecto importante del procesamiento en paralelo es el tiempo de ejecución en paralelo que es el lapso entre el inicio y final de la aplicación, el cual consiste en el tiempo utilizado para el cálculo en los procesadores o núcleos y el tiempo para el intercambio de datos o sincronización. El tiempo de ejecución paralelo tiene que ser más pequeño que el tiempo de ejecución secuencial en un solo procesador, de tal modo que bien vale el esfuerzo diseñar un programa paralelo. Este tiempo es influenciado por la distribución del trabajo a procesadores o núcleos, el tiempo para el intercambio de información o sincronización y tiempos inactivos en que un procesador no puede hacer nada útil pero espera a que pase algún evento. En general un tiempo pequeño de ejecución resulta cuando la carga de trabajo es asignada uniformemente a los procesadores o núcleos, lo que se conoce como carga balanceada, y cuando los recursos para el intercambio de información, sincronización y tiempos inactivos son pequeños. La búsqueda de una programación específica y una estrategia de segmentación que lleven a un buen balance en la carga de procesamiento y poco consumo de recursos, es a menudo difícil por las muchas interacciones. Por ejemplo, reducir los recursos para el intercambio de información lleva a desbalances en la carga de procesamiento, mientras un buen balance puede requerir más recursos para intercambio de información y sincronización. Para una evaluación cuantitativa del tiempo

de ejecución de programas paralelos se usan medidas de costo tales como *speedup* y eficiencia, que comparan el tiempo de ejecución paralelo resultante con el tiempo de ejecución secuencial en un procesador [Foster, 1994].

### 3.3. Arquitecturas de procesamiento en paralelo

En aproximadamente tres décadas entre principios de los 1960's y mediados de los 1990's, científicos e ingenieros exploraron una amplia variedad de arquitecturas del procesamiento en paralelo. En los 1960's muchas compañías tomaron ventaja de las nuevas facilidades de fabricación en el desarrollo de procesadores especiales para el procesamiento en paralelo, mientras otros confiaron en la misma propuesta basada en estaciones de trabajo y computadoras personales. Hoy en día muchos expertos debaten si los sistemas de procesamiento en paralelo dominantes contendrán unas pocas docenas de procesadores de alto desempeño o miles de procesadores con características comunes.

Todos los sistemas con múltiples procesadores deben proveer una manera para que los procesadores interactúen. En algunos sistemas, los procesadores usan la red de interconexión para acceder a memoria compartida, y en otros se utiliza para mandar mensajes entre procesadores cuando la memoria se encuentra distribuida en los diferentes equipos. La taxonomía de *Flynn* [Rauber y Runger, 2010] es el régimen de clasificación para el procesamiento en paralelo. El objetivo de esta clasificación está en la multiplicidad del *hardware* usado para manipular las instrucciones y el flujo de datos, esto es el modo en que se van ejecutando las tareas programadas. El *hardware* de una computadora puede soportar un solo flujo de instrucción o múltiples flujos de instrucciones, manipulando un solo flujo de datos o múltiples flujos de datos, véase Figura 3.2 [Rauber y Runger, 2010], por lo tanto la clasificación de *Flynn* resulta en cuatro categorías:

- SISD (*Single-Instruction, Single-Data*)
- SIMD (*Single-Instruction, Multiple-Data*)
- MISD (*Multiple-Instruction, Single-Data*)
- MIMD (*Multiple-Instruction, Multiple-Data*)

Casi toda la propuesta general del procesamiento en paralelo está basada en el modelo MIMD, véase Figura 3.3 [Rauber y Runger, 2010]. Una clasificación más de procesamiento se puede hacer de acuerdo a su organización de memoria. Se pueden distinguir dos aspectos: la

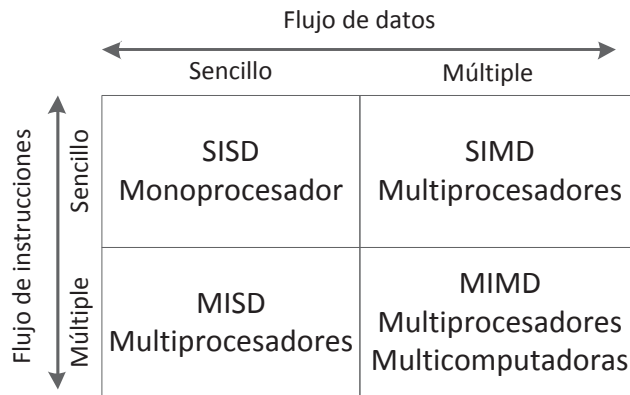


Figura 3.2: Arquitectura de la taxonomía de Flynn

organización física de memoria y el punto de vista del organizador de la memoria. Para la organización física, se pueden tener computadoras con una memoria física compartida (también llamada multiprocesador) y computadoras con memoria física distribuida (también llamada multicomputadoras). Pero también existen muchas organizaciones híbridas, por ejemplo dada una memoria física compartida virtual en la superficie de una memoria física distribuida.

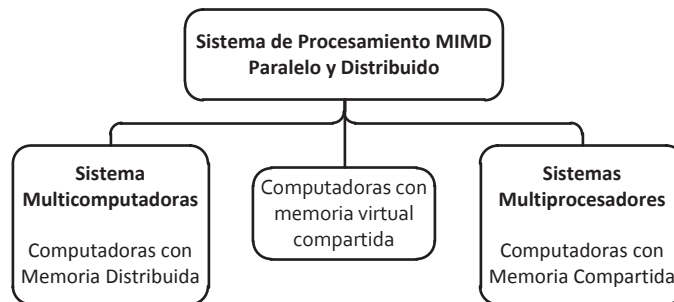


Figura 3.3: Formas de organización de memoria en MIMD

### 3.3.1. Procesamiento con memoria distribuida

El procesamiento con memoria física distribuida es también llamado máquinas de memoria distribuida DMM (Distributed Memory Machine), véase Figura 3.4, la cual consiste de un número de elementos de proceso (llamados nodos) y una red de interconexión que conecta los nodos y soporta la transferencia de datos entre ellos. Un nodo es una unidad independiente, consta de procesos, memoria local y algunas veces elementos de periferia.

Los datos del programa son almacenados en la memoria local de uno o varios nodos. Toda

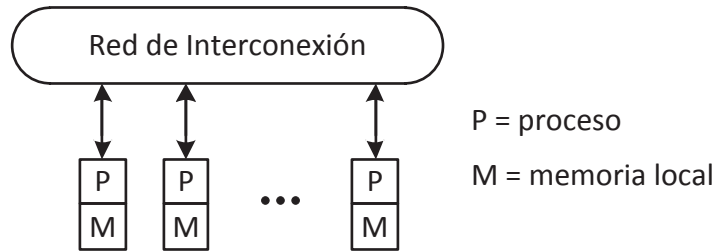


Figura 3.4: Proceso con memoria distribuida

la memoria local es privada y solo los procesadores locales pueden acceder a ella directamente. Cuando un procesador necesita información de memoria local de otro nodo para realizar el procesamiento local, se tiene que realizar el *message – passing* a través de una red de interconexión. Por lo tanto, las máquinas de memoria distribuida están fuertemente conectadas con el modelo de programación *message – passing*, que está basado en la comunicación entre procesos secuenciales cooperantes [Rauber y Runger, 2010]. La ejecución de operaciones de comunicación puede ser desacoplada de las operaciones de procesamiento por medio de la inclusión de un DMA (*direct memory access*) a los nodos de control. Esto permite transferir datos desde o hacia la memoria local sin la participación del procesador y permitiendo comunicación sincronizada. Técnicamente, los DMA son bastante fáciles de ensamblar, por lo tanto, las computadoras de escritorio pueden ser usadas como nodos [Rauber y Runger, 2010].

### 3.3.2. Procesamiento con memoria compartida

Las computadoras con memoria física compartida también son llamadas máquinas de memoria compartida (SMM *Shared Memory Machines*) [Rauber y Runger, 2010]. La Figura 3.5 muestra el esquema de un SMM. La memoria compartida también es llamada memoria global. Las SMM consisten de un número de procesadores o núcleos, una memoria física compartida (memoria global) y una red de interconexión para conectar a los procesadores con la memoria. La memoria compartida puede ser implementada por un conjunto de módulos de memoria. La información puede ser intercambiada entre procesadores vía memoria global para lectura y escritura de variables compartidas. Los núcleos de un procesador multinúcleo son un ejemplo para un SMM. Físicamente la memoria global consiste de módulos de memoria separados proporcionando un espacio de dirección común en el que pueden acceder todos los procesadores [Rauber y Runger, 2010].

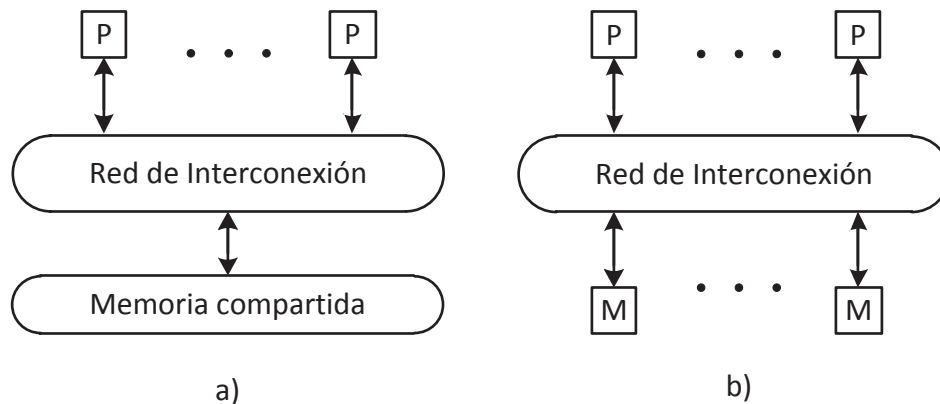


Figura 3.5: Procesamiento con memoria compartida. a) Vista abstracta b) Memoria compartida con módulos de memoria

### 3.4. Paralelismo a nivel de threads

La organización de la arquitectura en un microprocesador puede requerir el uso de programas paralelos explícitos para hacer más eficiente el uso de los recursos proporcionados. Esto es llamado paralelismo a nivel de *threads*, por lo tanto los flujos de control necesario son a menudo llamado threads. La organización de la arquitectura correspondiente es también llamada *chip multiprocessing* (CMP) [Nichols *et al.*, 1996]. Un ejemplo de CMP es la colocación de múltiples núcleos de ejecución independientes con todos los recursos de ejecución en un solo procesador. El proceso resultante es llamado procesamiento multinúcleo. Un enfoque alternativo es el uso de *multithreading* para ejecutar múltiples *threads* cuando sea necesario para el *hardware* [Nichols *et al.*, 1996].

La idea de *multithreading* simultáneo (SMT) es usar varios *threads* y programar instrucciones ejecutables de diferentes *threads* en el mismo ciclo, si es necesario, por lo tanto se utiliza la unidad funcional de un procesador de manera más eficiente. Esto conduce a una ejecución simultánea de varios *threads*, lo que da el nombre a esta técnica. En cada ciclo, las instrucciones de varios *threads* compiten por la unidad funcional de un procesador.

En el procesamiento multinúcleo se integran múltiples núcleos de ejecución en un solo microprocesador, donde para la operación del sistema, cada núcleo de ejecución representa un proceso lógico independiente con recursos de ejecución separados como unidades funcionales o ejecución segmentada. Cada núcleo tiene que ser controlado separadamente, y el sistema operativo puede asignar diferentes programas de aplicación a los diferentes núcleos para obtener una ejecución en paralelo. Las aplicaciones de fondo tales como chequeo de virus, compresión de imágenes y codificación pueden correr en paralelo a programas de aplicación del usuario.

Gracias al uso de técnicas de procesamiento en paralelo, es posible ejecutar programas que contengan aplicaciones de procesamiento intenso tales como juegos, visión por computadora o simulaciones científicas, sobre un conjunto de núcleos de forma paralela; por lo tanto, el tiempo de ejecución se verá reducido en comparación a una ejecución con un solo núcleo. En el futuro, los usuarios de programas de aplicaciones estándar como son los juegos de computadora, probablemente esperaran un uso más eficiente de los núcleos de ejecución de un microprocesador, y para lograr esto, los programadores tendrán que usar indudablemente técnicas de procesamiento en paralelo [Nichols *et al.*, 1996].

El uso de múltiples núcleos en un solo microprocesador también permite a programas estándar tales como: los procesadores de textos, aplicaciones de oficina o juegos de computadora, dar función adicional que sea procesada en un segundo plano sobre un núcleo del procesador de forma independiente, de tal manera que el usuario no notará ningún retardo o retardo mínimo en la aplicación principal.

### 3.5. Paralelización de programas

La paralelización de un algoritmo o programa es típicamente desarrollada en base al modelo del programa usado. Independientemente del modelo de programación utilizado, se pueden identificar los pasos típicos para realizar la paralelización. Se pueden describir tres pasos, asumimos que el cálculo a ser paralelizado es dado en la forma de un programa secuencial o algoritmo. Para transformar el cálculo secuencial a un programa paralelo, su control y dependencia de datos tienen que ser tomadas en consideración para asegurar que el programa en paralelo produzca el mismo resultado como el programa secuencial para todos los posibles valores de entrada. El objetivo principal es reducir el tiempo de ejecución del programa tanto como sea posible usando múltiples procesadores o núcleos. La transformación en un programa secuencial a un programa en paralelo es también referida como paralelización. Para desempeñar esta transformación en un camino sistemático, en [Quinn, 2004] se divide en varios pasos:

**Descomposición del cálculo** El cálculo del algoritmo secuencial es descompuesto en tareas, y las dependencias entre las tareas son determinadas. Las tareas son las pequeñas unidades del paralelismo [Quinn, 2004].

**Asignación de tareas a procesos o *threads*** Un proceso o *thread* representa un flujo de control ejecutado por un procesador o núcleo. Un proceso o *thread* puede ejecutar diferentes tareas, una después de la otra. El número de procesos o *threads* no necesita ser igual al número de procesadores o núcleos físicos, pero a menudo se usa el mismo número. El

objetivo principal de este paso es asignar las tareas de tal modo que resulte un buen balance en la carga para cada proceso o *thread*, donde deben tener alrededor del mismo número de cálculos a realizar. La asignación de tareas a procesos o *threads* es llamada *scheduling*, el cual es un método para determinar la eficiencia en la ejecución de un conjunto de tareas de una duración determinada sobre un conjunto de unidades de proceso [Quinn, 2004].

**Asignación de procesos o *threads* a procesadores o núcleos físicos** En el caso simple, cada proceso o *thread* es asignado a un procesador o núcleo de manera separada. Si se dispone de menos núcleos que *threads*, deben asignarse múltiples *threads* a un solo núcleo. Esta asignación puede ser hecha por el sistema operativo, pero también puede ser soportado por declaraciones en el programa [Quinn, 2004].

### 3.6. Speedup y Eficiencia

Cuando se implementa un programa en paralelo, se hace esperando que esté se ejecute más rápido que su contraparte secuencial. El *speedup* es la relación entre el tiempo de ejecución secuencial y el tiempo de ejecución en paralelo, y está dado por [Chandra *et al.*, 2001].

$$Speedup = \frac{T_1}{T_m} \quad (3.1)$$

- $T_1$  Tiempo de ejecución con un solo procesador.
- $T_m$  Tiempo de ejecución con  $m$  procesadores.
- $m$  Número de procesadores.

El *speedup* es inversamente proporcional al tiempo de ejecución paralelo y tiene como base el tiempo de ejecución secuencial. De 3.1 resulta claro que al reducir el tiempo de procesamiento en paralelo se incrementa el *speedup* pero la realidad es que a medida que se incrementa el número de procesadores en paralelo, también el tiempo de comunicación entre ellos será mayor y la ganancia en términos del *speedup* comenzará a decrecer como lo predice la ley de *Amdahl*.

La eficiencia de un programa paralelo y es el grado de utilización de los procesador, midiendo los tiempo de efectivos de ejecución y los tiempos muertos donde los procesadores no realizan ningún cálculo. Por lo tanto la eficiencia es la división del *speedup* entre el número total de procesadores utilizados [Chandra *et al.*, 2001]:

$$Eficiencia = \frac{T_1}{m \times T_m} \quad (3.2)$$

Donde la eficiencia es mayor o igual a cero:  $0 \leq Eficiencia \leq 1$ , esto quiere decir que el valor de la eficiencia muestra el grado en el que se está utilizando la capacidad de procesamiento de las unidades.

### 3.7. Ley de Amdahl

Esta ley establece que si un programa tiene una sección paralelizable, y otra sección la cual debe de ser ejecutada en forma secuencial, entonces el tiempo de ejecución del programa tendrá una aproximación asintótica hacia el tiempo de ejecución secuencial, esto quiere decir que por más elementos de procesamiento que sean utilizados el tiempo de computo no decrecerá más [Lewis y Berg, 1998]. Sea  $\gamma$  la fracción de operaciones que deben realizarse en forma secuencial, donde  $0 \leq \gamma \leq 1$ , y el maximo speedup posible con un arreglo paralelo con  $m$  elementos de proceso se establece por (3.3).

$$Speedup \leq \frac{1}{\gamma + \frac{(1-\gamma)}{m}} \quad (3.3)$$

La Figura 3.6 muestra el comportamiento del tiempo total de ejecución utilizando procesamiento en paralelo.

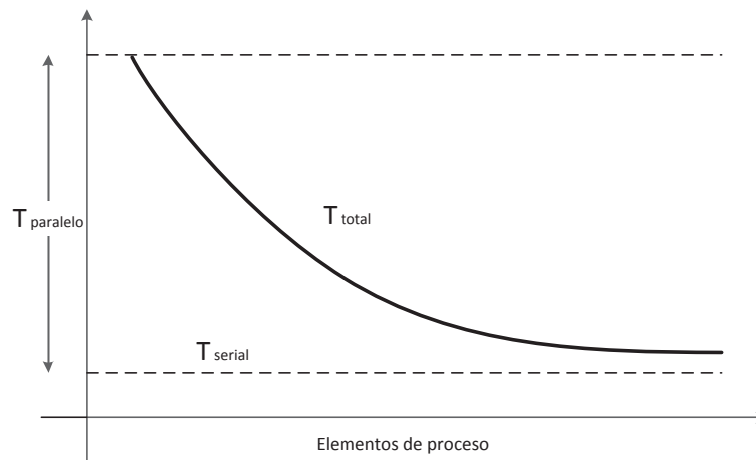


Figura 3.6: Comportamiento del tiempo de ejecución de un programa usando múltiples elementos de proceso

En la Figura 3.6 se aprecia que el uso de un mayor número de elementos de proceso reduce el tiempo total de ejecución, sin embargo, se observa que la curva tiende hacia el tiempo de ejecución secuencial, el cual no puede ser reducido debido a que no siempre todas las tareas pueden ser paralelizadas [Quinn, 2004].



### 3.8. Granularidad

En la paralelización de una aplicación, es necesario contar con un lenguaje de programación o biblioteca que brinde las herramientas necesarias para ello, comenzando con la división de tareas en términos de instrucciones para su posterior ejecución en forma paralela. Dependiendo de la herramienta con que se cuente para la división del código en múltiples tareas es como surge el término de granularidad, pudiendo existir diferentes niveles de granularidad. Por lo tanto, la granularidad es el tamaño promedio de las partes de un programa que se asignan a cada procesador o que se quieren ejecutar concurrentemente: Instrucciones, iteraciones, funciones, etc., por lo que este término se utiliza como el mínimo componente del sistema que puede ser preparado para ejecutarse de manera paralela. Por norma general, cuanto más fuertemente acoplado es un sistema, menor es la granularidad del paralelismo en la programación [Chandra *et al.*, 2001]. Dependiendo del grado de granularidad del sistema se clasifica en:

**Granularidad fina:** Por lo regular, se hace a nivel de instrucciones en lenguaje muy cercano al ensamblador. Generalmente son explotados por sistemas de *hardware* muy costosos con los nodos o procesadores fuertemente acoplados (multiprocesadores). Es la granularidad más pequeña y basa prácticamente todo su funcionamiento en propiedades del *hardware*. El *hardware* puede ser suficientemente inteligente para que el programador no tenga que hacer mucho por soportar esta granularidad. Por ejemplo, el *hardware* puede aportar reordenamiento de instrucciones. Las tareas individuales son relativamente pequeñas en términos de tiempo de ejecución. La comunicación entre los procesadores es frecuente. También se le conoce como paralelismo de datos.

**Granularidad media:** Por lo general es explotado por el programador o el compilador. Dentro de él también se encuentran diversas bibliotecas como pueden ser PVM (*Parallel Virtual Machine*) o MPI (*Message Passing Interface*). El *hardware* normalmente también se prepara para poder aprovechar este tipo de paralelismo. Por ejemplo, los procesadores pueden disponer de instrucciones especiales para ayudar en el cambio de una tarea a otra que realiza el sistema operativo. La comunicación entre los procesadores es poco frecuente y se realiza después de largos periodos de ejecución.

**Granularidad gruesa:** Es el que explota el programador mediante programas que no tienen por qué requerir la utilización de ninguna biblioteca externa, sino solamente el uso de conocimientos de programación para paralelizar un algoritmo. Se basa principalmente en cualquier tipo de medio que utilice el programador para crear un programa, que solucione un problema de manera paralela, sin tener por qué hacer uso más que de su habilidad de

programador y de un buen algoritmo. Son los más limitados al carecer de métodos específicos para comunicación entre nodos o procesadores, se dan en sistemas muy débilmente acoplados (multicomputadoras). También se le conoce como paralelismo de tareas.

### 3.9. Plataformas del procesamiento en paralelo

La posibilidad de emplear programación paralela de cálculos, depende fuertemente de la arquitectura y plataforma de ejecución que se elijan. Las principales plataformas que se pueden mencionar son: *PVM*, *MPI*, *OpenMP* y *Threads*.

**PVM** En los años 80's del siglo pasado muchas compañías comenzaron a fabricar y a vender multicomputadoras, típicamente, el entorno de programación era un lenguaje secuencial ordinario (*C/C++* ó *Fortran*) aumentada con una librería de paso de mensaje para la comunicación entre las máquinas, pero el problema que existía era que cada proveedor tenía su propio llamado de funciones. En el verano de 1989 aparece la primera versión de una librería de paso de mensaje llamada *PVM*, escrita por *Oak Ridge National Laboratory*. *PVM* facilita la ejecución de programas en paralelo a través de colecciones heterogéneas de sistemas para ser vistos como una sola máquina paralela eliminando la barrera de tener variedad en los equipos de proceso. Los miembros del equipo reescribieron el *software* dos veces para una versión 3.0 de *PVM* que se publicó en Marzo de 1993 [Quinn, 2004].

**MPI** La versión 1.0 del estándar *MPI* aparece en Mayo de 1994. *MPI* es una estandarización de las bibliotecas de paso de mensajes definiendo la sintaxis y semántica de las rutinas para los patrones de comunicación estándar. Dentro de esta plataforma los programas escritos en *Fortran*, *C/C++*, por los usuarios son decodificados en compiladores comunes y vinculados con la biblioteca *MPI*. A partir de esta plataforma, todos los proveedores de cómputo en paralelo ofrecieron una implementación *MPI* para sus máquinas, donde las implementaciones públicas están disponibles y pueden ser descargadas del Internet. Un programa *MPI* planteado en forma correcta debe ser capaz de correr en todas las implementaciones *MPI* sin cambios. En este modelo se asume que el hardware principal es una colección de procesadores, cada uno con su propia memoria local, y una red de interconexión que soporta paso de mensajes entre procesadores [Quinn, 2004].

**OpenMP** Es una implementación *multithreading*, *OpenMP* un método de paralelización por medio del cual el *thread* maestro (serie de instrucciones ejecutadas consecutivamente) se divide en un número específico de *threads* esclavos y la tarea objetivo es dividida entre ellos. Entonces los *threads* se ejecutan simultáneamente, donde en el entorno de ejecución

se asignan *threads* a diferentes procesadores. Un programa *OpenMP* siempre comienza con un solo thread de control que tiene una asociación con un contexto de ejecución o dato de entorno. Este *thread* de control inicial es referido como el *thread* maestro. El *thread* maestro y el contexto de ejecución existen durante todo el programa. Cuando el *thread* maestro encuentra un constructor paralelo, se crean nuevos *threads* de ejecución con un contexto de ejecución para cada *thread*. Además *OpenMP* proporciona dos tipos de constructores para controlar el paralelismo. El primero, da una directiva para crear múltiples *threads* de ejecución que se ejecutan al mismo tiempo y la única instancia es la directiva paralela que encierra un bloque de código y crea un conjunto de *threads* donde cada uno de ellos ejecuta este bloque de código al mismo tiempo. Y segundo, *OpenMP* proporciona constructores para dividir trabajo entre un conjunto existente de *threads* paralelos, donde una instancia de este es la directiva "for", usada para explotar el paralelismo a nivel de ciclos (*loop – level*). Este divide las iteraciones de un ciclo entre múltiples *threads* ejecutándose al mismo tiempo [Chandra *et al.*, 2001].

**Threads** El modelo de *threads* es una extensión del modelo de procesos. En este modelo de *threads*, cada proceso puede consistir de múltiple independencia del flujo de control y es llamado *thread*. La palabra thread es usada para indicar que una secuencia continua de instrucciones. Durante la ejecución de un proceso, los diferentes *threads* referentes a éste son asignados para ejecutar los recursos por un método de programación. Los *threads* son típicamente usados si la ejecución de los recursos tiene acceso a memoria física compartida, como es el caso de un procesador multinúcleo, en particular, los diferentes *threads* de un proceso pueden ser asignados a diferentes núcleos de un procesador multinúcleo [Nichols *et al.*, 1996].

### 3.10. OpenMP

OpenMP es un modelo de programación paralelo para memoria compartida y memoria distribuida multiprocesadores. Introducido por primera vez por SGI (*Silicon Graphics International Corp*) y desarrollado en colaboración con otros proveedores de cómputo. *OpenMP* se ha vuelto rápidamente un factor estándar para la paralelización de aplicaciones. Hoy hay una organización *OpenMP* independiente con muchos de los mejores fabricantes de equipo de cómputo en el área, incluyendo a *Compaq*, *Hewlett – Packard*, *Intel*, *IBM*, *KuckAssociates (KAI)*, *SGI*, *Sun* and *U.S. Department of Energy ASCI Program*. Los esfuerzos de *OpenMP* también han sido probados por más de 15 proveedores y desarrolladores de aplicaciones, reflejando un amplio soporte industrial para los estándares de OpenMP [Chandra *et al.*, 2001].

Desafortunadamente, la principal información disponible acerca de *OpenMP* es la *OpenMP specification* (disponible en el sitio *web* de *OpenMP* [www.openmp.org](http://www.openmp.org)). Aunque ésto es apropiado como especificaciones formales, no es un formato muy detallado para programadores que deseen usar *OpenMP* en aplicaciones paralelas más completas.

La interfaz de programación *OpenMP* es un estándar ya consolidado, donde las API *OpenMP* definen un conjunto de directivas que aumentan el estándar de *C/C++* y *Fortran77/90* [Sato, 2002]. En contraste a previos API, *OpenMP* facilita el incremento de la aproximación a la paralelización de un programa secuencial. El programador puede añadir directivas de paralelización a ciclos o declaraciones en el programa. Esto también es de interés para el grupo de la comunidad de cómputo, debido a que muchos grupos recientes son construidos a partir de nodos de memoria compartida. Una API *OpenMP* se puede usar para explotar el paralelismo en un nodo mientras un *message – passing* API es usado entre nodos [Sato, 2002].

### 3.10.1. Modelo de Ejecución

Las API *OpenMP* utilizan el modelo *fork – join* de ejecución paralela, véase Figura 3.7, en éste modelo la ejecución del programa comienza con el *master thread* en forma secuencial hasta el punto A donde se encuentra un constructor paralelo y se crean múltiples *thread* para ejecutar en forma paralela a la función “*foot*” hasta el punto D donde termina la región paralela y solo el *master thread* continua con la ejecución secuencial de las tareas. El modelo *fork – join* puede ser usado de manera eficiente para resolver una gran variedad de problemas, es algo que se adaptó para aplicaciones de grandes arreglos. *OpenMP* pretende soportar programas que se ejecuten correctamente como programas paralelos (múltiples *threads* de ejecución y una completa biblioteca de soporte *OpenMP*), o como programas secuenciales (ignorando directivas y una simple biblioteca de apoyo *OpenMP*). Sin embargo, es posible y permitido desarrollar un programa que se ejecute correctamente como un programa paralelo pero no como un programa secuencial, o que genere diferentes resultados cuando se ejecuta como un programa paralelo [Sato, 2002].

Un programa *OpenMP* comienza su ejecución con un solo *thread*, llamado *thread* inicial o *master thread*. El *thread* inicial se ejecuta en forma secuencial como si estuviera encerrado dentro de una región paralela inactiva implícita abarcando a todo el programa [Sato, 2002]. Sin embargo, cuando cualquier *thread* encuentra un constructor *parallel*, el *thread* crea un conjunto de *threads* de él mismo que se añaden al proceso de solución, y éste se vuelve el *master thread* del nuevo conjunto. Todos los miembros del nuevo conjunto ejecutan el código dentro del constructor *parallel*. En el constructor *parallel* existe un *barrier* implícito al final de este, y sólo el *master thread* continuara ejecutando el código del usuario más allá del constructor *parallel*.

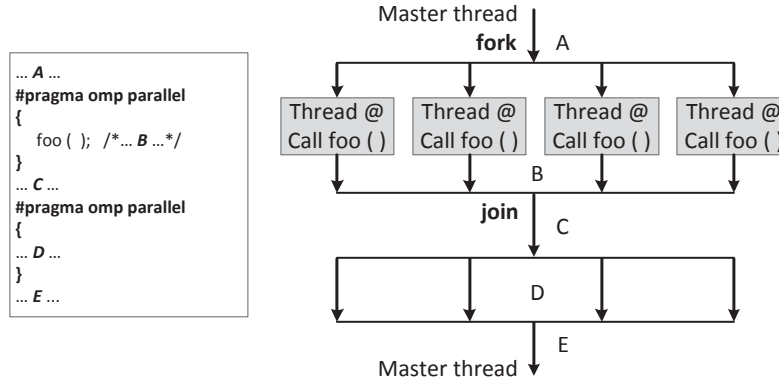


Figura 3.7: Modelo OpenMP fork-join

Cualquier número de constructores *parallel* se pueden especificar en un solo programa. Esto quiere decir que se pueden anidar arbitrariamente regiones paralelas dentro de cualquier otra. Si el paralelismo anidado está deshabilitado o no es soportado por las implementaciones *OpenMP*, entonces el nuevo conjunto creado por el *thread* que se encuentra con el constructor *parallel* dentro de una región paralela consistirá solamente del *thread* encontrado. Sin embargo, si se soporta el paralelismo anidado y está habilitado, entonces el nuevo conjunto de *threads* puede consistir en más de uno [Sato, 2002].

Cuando un conjunto de *threads* encuentra un constructor *work – sharing*, el trabajo dentro del éste se divide entre el número total de *threads* del conjunto y se ejecutan de manera cooperativa, en lugar de comenzar la ejecución con cada *thread* por separado. Existe un *barrier* opcional al final del constructor *work – sharing*, de tal modo que la ejecución del código por cada *thread* se reanuda al termino del constructor *work – sharing* [Chandra *et al.*, 2001].

En *OpenMP* están disponibles constructores y bibliotecas de sincronización para coordinar *threads* y datos en constructores *parallel* y *work – sharing*. Adicionalmente se encuentran disponibles rutinas de bibliotecas y variables de entorno para controlar y verificar las rutinas de entorno de los programas *OpenMP*.

*OpenMP* no garantiza que la entrada y salida al mismo archivo estén sincronizadas cuando se ejecuta en paralelo. En este caso el programador es responsable de sincronizar los estados de entrada y salida usando constructores de sincronización y rutinas de bibliotecas disponibles. Para el caso donde cada *thread* accede a diferentes archivos, la sincronización hecha por el programador no es necesaria [openmp.org].

### **3.11. Conclusiones**

De acuerdo a lo presentado en el presente capítulo sobre el procesamiento en paralelo, se ha podido elegir la plataforma OpenMP para realizar la paralelización de los métodos convencionales de integración, debido a su propuesta innovadora de procesamiento en paralelo que utiliza de manera eficiente la característica multinúcleo de los equipos de cómputo en un esquema de memoria compartida, lo que permite que el programa sea portable a distintos equipos con diferente número de procesadores y núcleos, sin la necesidad de realizar adaptaciones, e incluso en el caso de tener solo un procesador disponible, el programa paralelo se ejecutará sin ningún problema en forma secuencial.

## Capítulo 4

# Propuesta de Análisis Para Sistemas Eléctricos Usando Procesamiento en Paralelo de Grano Fino

### 4.1. Introducción

Como ya se ha mencionado en esta tesis, en el estudio de las ecuaciones diferenciales que modelan la dinámica de los sistemas eléctricos de potencia la herramienta numérica ha jugado siempre un papel importante, debido a que la mayor parte de las ecuaciones que aparecen en los problemas prácticos no se pueden resolver exactamente y por tanto hay que recurrir a algún tipo de aproximación para tener una idea del comportamiento de la red eléctrica a partir del conjunto de ecuaciones diferenciales que la modelan. Sin embargo, durante el siglo pasado y buena parte de este, la importancia de los medios de cálculo fue una barrera para la utilización de muchos métodos numéricos. Esta situación comienza a cambiar a partir de los años cuarentas y cincuentas con la aparición de los primeros ordenadores electrónicos y se ha acelerado especialmente durante los últimos veinte años con la disponibilidad de ordenadores personales que poseen una elevada velocidad de cálculo además de una gran capacidad de almacenamiento de datos. Estas posibilidades han tenido doble efecto en la matemática numérica: por una parte ha sido posible abordar la resolución efectiva de problemas que debido a su complejidad habían quedado anteriormente fuera de toda consideración, y por otra parte, ha sido necesario desarrollar software nuevo para hacer frente a estos problemas con la máxima eficiencia posible.

La forma más común para representar la dinámica de un SEP se hace a través de un conjunto de EDO en el modelo de espacios de estado de  $n$  variables de estado y  $v$  entradas, donde la notación matricial está dada por (4.1).

$$\dot{X} = A \cdot X + B \cdot U \quad (4.1)$$

Donde:

$A$  es una matriz cuadrada de  $n \times n$ .

$B$  es una matriz de entradas de  $n \times v$ .

$X$  es el vector de estados  $n \times 1$ .

$U$  es el vector de señales de entrada  $n \times 1$ .

La solución en el dominio de tiempo de este tipo de sistemas de EDO se puede obtener por medio de métodos de fuerza bruta o métodos de acercamiento rápido [Aprilie y Trick, 1972]. Este tipo de métodos en el dominio del tiempo hacen uso de la aplicación sucesiva de métodos de integración como la regla trapezoidal, el método de Euler, métodos de Runge-Kutta o cualquier otro, lo cual evita resolver un conjunto de EDO en forma directa, es decir, la idea de utilizar alguno de los métodos de integración es ver al conjunto de EDO que modelan al sistema como un sistema lineal para pequeños intervalos de tiempo, esto hace que ahora se resuelva un conjunto de ecuaciones algebraicas y no de ecuaciones diferenciales, y la elección de cualquiera de ellos dependerá de las necesidades del investigador y características del sistema que se esté analizando.

Tradicionalmente el proceso se realiza de manera secuencial, donde la complejidad de la operación es  $O(n^2)$  [Miller y Boxer, 2005], esto significa que el tiempo de cómputo requerido para el cálculo es proporcional al cuadrado del tamaño  $n$  de matriz, la cual está asociada al número de variables de estado que modelan la dinámica de la red a través del conjunto de EDO.

El análisis de sistema eléctricos de gran escala implica

- Un mayor número de nodos
- Un modelado más preciso de sus elementos
- Una mayor interconexión entre los diferentes elementos, etc.

Ante esta situación y considerando las características y ventajas tanto de las plataformas como de las arquitecturas para el procesamiento en paralelo se propone a continuación un esquema a través del cual se aplicarán técnicas de procesamiento en paralelo basadas en OpenMP en el análisis en estado estacionario de redes eléctricas de gran escala.



## 4.2. Proceso de paralelización del método de integración

La mayor parte de los métodos de análisis en estado estable o en estado transitorio de una red eléctrica modelada mediante el uso de EDO, usan de alguna manera la evaluación de la función matricial

$$\dot{f}(x,t) = A \cdot X + B \cdot U \quad (4.2)$$

En el proceso para la determinación del estado estacionario periódico estacionario se observa que es la operación más utilizada, por lo tanto en esta tesis se enfocará en el esfuerzo para realizar esta operación en forma eficiente haciendo uso de técnicas de procesamiento en paralelo de grano fino.

La Figura 4.1 muestra como se realiza el proceso de evaluación de (4.2).

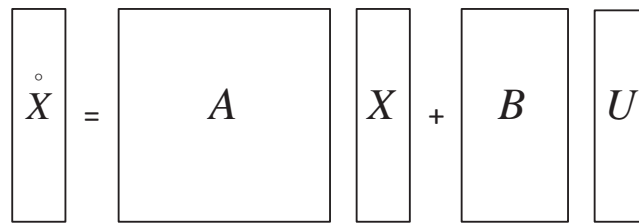

$$\begin{matrix} \circ \\ \dot{X} \end{matrix} = \begin{matrix} & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{matrix} \begin{matrix} X \\ \\ \\ \\ \end{matrix} + \begin{matrix} B \\ \\ \\ \\ \end{matrix} \begin{matrix} U \\ \\ \\ \\ \end{matrix}$$

Figura 4.1: Representación de un sistema lineal

En este esquema, la solución se puede obtener por medio de la aplicación de alguno de los métodos de integración tales como la Regla Trapezoidal, el método de Euler o el método de Runge-Kutta. Tradicionalmente estos métodos requieren la solución de (4.1) dada por (4.2).

Desde el punto de vista de operación, la solución de (4.2) implica la multiplicación de la matriz  $A$  por el vector de condiciones iniciales  $X$  combinado con el resultado del producto de  $B$  y  $U$  para un punto en el tiempo. Estas operaciones hechas en modo secuencial toman la primer fila de la matriz  $A$  y se multiplica por el vector  $X$ , al resultado se le suma el producto de la primer fila de la matriz  $B$  por el vector  $U$ , esto resulta en el primer elemento del vector solución. Este proceso se repite hasta evaluar la última fila de la matriz  $A$  y  $B$  para obtener el último elemento del vector solución.

De la Figura 4.2 se puede ver que el cálculo del primer elemento de  $\dot{X}$  es independiente del segundo, y el cálculo del segundo elemento es independiente del tercero y así sucesivamente, donde  $X_0$  es el primer elemento del vector  $X$ ,  $A_0$  es la primer fila de la matriz  $A$  y  $B_0$  es la primer fila de la matriz  $B$ . Por ésta razón, el cálculo de cada uno de los elementos de  $\dot{X}$  es independiente de cualquiera de los otros y pueden ser obtenidos en forma paralela usando un conjunto de



proceso calcula una sola fila de  $\dot{X}$ . En el caso de que el número de filas es mayor, entonces cada elemento de proceso calcula más de una fila de  $\dot{X}$ , y en el caso de presentarse un número menor de elementos de proceso es mayor al número de variable de estado, los elementos de proceso sobrantes no se utilizan, esto evita que se generen esquemas de competencia.

### 4.3. Descripción del código en C/C++ utilizando OpenMP

La multiplicación de una matriz por un vector es la operación principal que se realiza en el proceso de solución. Esta operación es segmentada para obtener la multiplicación de un vector fila por un vector columna y que representa la máxima operación que se realiza en forma independiente por cada *thread*, estas operaciones son la parte esencial en el proceso de solución paralelo.

En el código C/C++ que se presenta en la Tabla 4.1 se pueden ubicar cuatro *#pragma omp parallel* esta leyenda quiere decir que la instrucción “*for*” que sigue a continuación se ejecutará en forma paralela y que el número de iteraciones de éste se tienen que dividir en el número de elementos de proceso que se tengan disponibles. Las cuatro instrucciones de ejecución paralela que se encuentran tienen como finalidad obtener en forma paralela los elementos que conforman a cada una de las  $K$ 's, como se muestra en el siguiente fragmento del código principal:

Tabla 4.1: Fragmento del código en C/C++ para análisis de SEP usando procesamiento en paralelo

```
1: double *Runge_Kutta(double t0,double *yini)
2: {
3:     for (io=0; io<hh; io++)
4:     {
5:         t01 = t00 + h; t02 = t00 + h/2.0;
6:         yn2 = inicializa2(y0);
7:         #pragma omp parallel for private(i)
8:         for(i=0;i<ve;i++)
9:             k_1[i] = evalua(t00,yn2,i);
10:        for(i=0;i<ve;i++)
11:            aux00[i] = y0[i] + h*k_1[i]/2.0;
12:        yn2 = inicializa2(aux00);
13:        #pragma omp parallel for private(i)
14:        for(i=0;i<ve;i++)
15:            k_2[i] = evalua(t02,yn2,i);
16:        for(i=0;i<ve;i++)
17:            aux00[i] = y0[i] + h*k_2[i]/2.0;
18:        yn2 = inicializa2(aux00);
19:        #pragma omp parallel for private(i)
20:        for(i=0;i<ve;i++)
21:            k_3[i] = evalua(t02,yn2,i);
22:        for(i=0;i<ve;i++)
23:            aux00[i] = y0[i] + h*k_3[i];
24:        yn2 = inicializa2(aux00);
25:        #pragma omp parallel for private(i)
26:        for(i=0;i<ve;i++)
27:            k_4[i] = evalua(t01,yn2,i);
28:        for (i=0; i<ve; i++)
29:            y0[i] = y0[i] + h*(k_1[i]+2.0*k_2[i]+2.0*k_3[i]+k_4[i])/6.0;
30:        t00 = t00+h;
31:    }
32: }
```

Con el fin de hacer eficiente el reparto de tareas entre los diferentes elementos de proceso se creó una función llamada “*evalua*” la cual hace la división de las tareas para asignarse a cada thread. Con esta función “*evalua*” se asegura que la tarea que realizará cada thread es la multiplicación de un vector de matriz  $A$  por el vector de estados  $X$  y sumará el resultado de la multiplicación de la matriz  $B$  por el vector  $U$ . El *barrier* se encuentra implícito al término de cada uno de los ciclos “*for*”.

En el caso de que no se especifique de la división de tareas como en este caso con la función “*evalua*” el reparto de trabajo entre los procesadores se hará automático y la máxima tarea que desempeñaría cada *thread* sería la multiplicación de elemento por elemento, lo cual no es una buena opción ya que se incrementa el tiempo de comunicación y sincronización entre los procesadores

#### 4.4. Conclusiones

La paralelización de operaciones sencillas y que son la base del método de integración de Runge-Kutta de cuarto orden fue esencial para realizar una división de las tareas a un nivel de grano fino, además de que del procesamiento en paralelo se aplicó en las zonas principales de los métodos de integración numérica, la aplicación se hizo en solo cuatro secciones del algoritmo que son las que aportan la mayor carga computacional. Las cuatro secciones paralelizadas representan a las cuatro pendientes que son necesarias para obtener la pendiente ponderada final, equivalente a resolver cuatro sistemas de ecuaciones algebraicas para obtener una aproximación del sistema de EDO en un periodo de integración. Esta metodología para realizar el procesamiento en paralelo permite obtener mejores desempeños a medida que se incrementan las dimensiones del problema, pero también hay que señalar que para la solución de sistemas donde el número de ecuaciones diferenciales es pequeño la ventaja de realizar la paralelización es mínima.

# Capítulo 5

## Modelado de elementos no lineales y variantes en el tiempo

### 5.1. Introducción

Existen diferentes marcos de referencia por medio de los cuales se puede realizar el análisis del comportamiento de una red eléctrica. Una de estas alternativas es el marco de referencia en el dominio del tiempo. En este marco de referencia, una red eléctrica es modelada mediante un conjunto de EDO. Una vez que la red es representada por medio de un conjunto de EDO's se aplica algún método numérico de solución con el objeto de conocer el comportamiento transitorio y en estado periódico estacionario de la red [Dommel, 1969].

En los sistemas eléctricos prácticos en donde el número de nodos y elementos que los conforman puede ser muy elevado, una tarea ardua es la formulación del conjunto de EDO's que describan su comportamiento dinámico. En [Ramos, 2007] se describe la técnica que permite la generación automática del conjunto de EDO que modelan una red cualquiera en base a la topología de la misma utilizando programación orientada a objetos, lo cual es la base para la obtención del modelo en espacio de estado necesario para realizar la paralelización a un nivel de grano fino del método de Runge-Kutta de cuarto orden.

Las técnicas usadas en esta tesis para la formación automática del conjunto de ecuaciones diferenciales ordinarias que modelan a la red eléctrica es la propuesta por [Ramos, 2007] la cual se resume a través del siguiente algoritmo:

1. Se lee un archivo de datos que contiene la topología de la red eléctrica así como los parámetros de la misma.
2. Se realiza el proceso de validación de los datos para evitar parámetros incorrectos, apari-

ción de islas eléctricas, elementos inexistentes, etc.

3. Se formula una ley de corrientes de Kirchhoff en cada uno de los nodos del sistema.
4. A partir de cada elemento leído desde el archivo de datos se va generando una lista simplemente enlazada de los objetos que están asociados a cada elemento de la red, los cuales pueden ser generadores, líneas de transmisión, bancos de capacitores, dispositivos FACTS, etc. Cada objeto tiene un procedimiento el cual está asociado a su representación en espacio de estado. Esta representación queda en función de variables de conexión con otros elementos, y los parámetros propios del elemento, las evaluaciones de la expresión entera se hace a través de cada elemento que es común a todos los objetos y cada elemento de la red conoce las variables de estado asociadas a él.
5. Finalmente el sistema puede verse en forma general como  $X = A \cdot X + B \cdot U$ .

El presente Capítulo tiene como objetivo realizar una presentación simple del modelo matemático que presenta a cada uno de los elementos utilizados en la formación de los sistemas eléctricos no-lineales analizados. La generación de conjunto de EDO se realiza a partir de un archivo de datos que contiene la topología de la red así como los parámetros de los diversos dispositivos que forman la red. Una vez que ha sido construido el conjunto de EDO se obtiene el comportamiento transitorio y el EEP de la red eléctrica. En base a un análisis nodal y de mallas se asocian las variables de estado de los diferentes modelos de los elementos y se representa al sistema en espacio de estado. Una vez terminado este proceso el SEP es visto desde el punto de vista computacional como un conjunto de matrices y vectores solamente. Una vez que se tiene una representación en espacio de estado del SEP, se obtiene la solución en EEP de la red eléctrica por medio del método de FB utilizando el método de integración de Runge-Kutta de cuarto orden. Adicionalmente, con el propósito de incrementar la eficiencia computacional en la obtención de la solución en EEP de la red eléctrica, son utilizadas técnicas de procesamiento en paralelo basadas en OpenMP. Cuando se ha obtenido el EEP se procede a realizar un estudio de la distorsión armónica en las formas de onda de las diferentes variables de estado asociadas con el modelado del sistema eléctrico. Para fines didácticos y de análisis el sistema desarrollado genera archivos de salida en los cuales se almacena el comportamiento en el tiempo de las diferentes variables de estado y el conjunto de EDO que modelan la dinámica de la red.

## 5.2. Representación monofásica de los elementos en un SEP

En la presente tesis se hace uso de diversos elementos para formar al SEP, los cuales son modelados en forma individual e interactúan con los demás elementos de acuerdo a la topología de la

red. A continuación se muestra el modelo matemático utilizado para representar a cada uno de los elementos del SEP contemplado en este trabajo.

### 5.2.1. Línea de transmisión

El modelo de la línea de transmisión que es utilizado en este trabajo está formado por una resistencia  $R$  y una inductancia  $L$ , como se muestra en la Figura 5.1.



Figura 5.1: Modelo de la línea de transmisión

La ecuación diferencial obtenida al aplicar LVK al circuito de la Figura 5.1 es:

$$\frac{di}{dt} = \frac{V_A - V_B}{L} - \frac{i(t) \cdot R}{L} \quad (5.1)$$

### 5.2.2. Banco de capacitores

De la Figura 5.2 la expresión  $\sum i_{salida}$  representa a las corrientes que salen del nodo al cual está conectado el banco de capacitores, y la expresión  $\sum i_{entrada}$  representa a las corrientes que entra al nodo. La corriente en el capacitor  $i_C$  no se incluye en ninguna de las dos expresiones anteriores. La corriente  $i_C$  se considera que está saliendo del nodo.

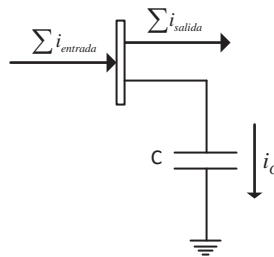


Figura 5.2: Banco de capacitores

Por medio de la aplicación de las LCK al circuito de la Figura 5.2 se tiene:

$$\sum i_{entrada} = \sum i_{salida} + i_C \quad (5.2)$$



Por lo tanto el voltaje en el nodo al cual está conectado el banco de capacitores es:

$$\frac{dV_C}{dt} = \frac{\sum i_{entrada} - \sum i_{salida}}{C} \quad (5.3)$$

### 5.2.3. Rama magnetizante

En la Figura 5.3 se muestra a una rama magnetizante conectada al nodo A.

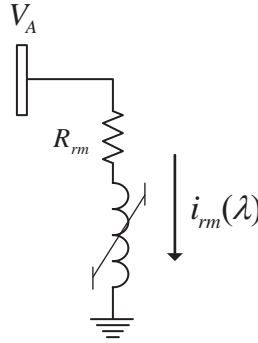


Figura 5.3: Rama magnetizante

Aplicando LVK al circuito de la Figura 5.3 se tienen que:

$$V_A = V_{rm} + V_L \quad (5.4)$$

donde:

$$V_{rm} = R_{rm} \cdot i_{rm}(\lambda) \quad (5.5)$$

$$V_L = L \cdot \frac{d\lambda}{dt} \quad (5.6)$$

sustituyendo (5.5) y (5.6) en (5.4) y despejando  $\frac{d\lambda}{dt}$  se tiene:

$$\frac{d\lambda}{dt} = \frac{V_A - R_{rm} \cdot i(\lambda)}{L} \quad (5.7)$$

En esta tesis el efecto de saturación en transformadores de potencia se representa mediante dos aproximaciones basadas en un polinomio de grado  $\eta$ , en la primera, la dinámica no-lineal de la rama magnetizante que se representa en [Dick y Watson, 1981] es:

$$i(\lambda) = \lambda^\eta \quad (5.8)$$

donde:

$\eta$  es un número impar, debido a que (5.8) tiene simetría impar.

En la segunda, el efecto de saturación en transformadores de potencia y ramas magnetizantes puede ser representado mediante la propuesta realizada en [Dick y Watson, 1981], en donde el efecto de saturación se presenta mediante un polinomio obtenido en forma experimental de la forma:

$$i(\lambda) = a\phi + b\lambda^\eta \quad (5.9)$$

donde:

$a$  es un coeficiente que corresponde a la pendiente en la región lineal de la curva de saturación.

$\eta$  es la medida de agudeza de la rodilla del transformador de potencia. El valor de  $\eta$  es impar debido a que la Ecuación (5.9) tiene simetría impar.

Los coeficientes  $a$ ,  $b$  y  $\eta$  se obtienen a partir de los datos experimentales tales como,  $\lambda_{nom}$ ,  $\lambda_j$ ,  $i_{nom}$  y  $i_j$ . Estos valores en el punto de operación  $j$ , que corresponden a la región lineal, son sustituidos en (5.9). Una vez que se sustituyen se determinan los valores de  $b$  a partir de diversos valores impares para  $\eta$ . Finalmente se selecciona el valor de  $b$  y  $\eta$  que mejor aproximen al punto de operación nominal.

### 5.2.4. Generador

En la Figura 5.4 se muestra el modelo monofásico del generador que es utilizado, el cual es modelado a través de una fuente senoidal constante de voltaje conectada en serie con una inductancia.

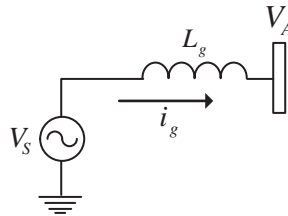


Figura 5.4: Modelo del generador

Aplicando las LVK y despejando  $\frac{di_g}{dt}$  se tiene:

$$\frac{di_g}{dt} = \frac{V_S - V_A}{L_g} \quad (5.10)$$

El voltaje en terminales del generador está dado por:

$$V_S = M \cdot \text{Sen}(\omega t + \phi) \quad (5.11)$$

donde:

$M$  es el valor del voltaje pico.

$\phi$  es el ángulo de fase del voltaje expresado en radianes.

$\omega$  es la frecuencia en radianes/segundo.

### 5.2.5. Transformador

En la Figura 5.5 se muestra el modelo monofásico del transformador saturable, el cual es utilizado en el presente trabajo.

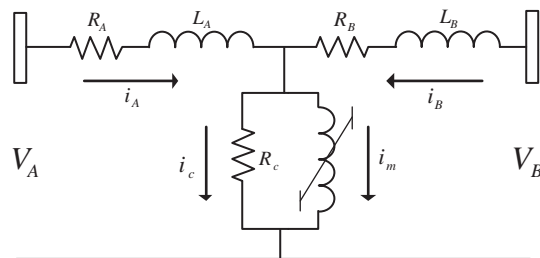


Figura 5.5: Modelo del transformador monofásico incluyendo el fenómeno de saturación

Donde:

$V_A$  es el voltaje terminal primario

$V_B$  voltaje terminal secundario

$R_A$  es la resistencia del devanado primario

$R_B$  resistencia del devanado secundario

$R_c$  es la resistencia del núcleo

$L_A$  es la inductancia de dispersión del primario

$L_B$  inductancia de dispersión del secundario

Por medio de la aplicación de LVK en la malla 1 del circuito mostrado en la Figura 5.5 se tiene:

$$\frac{di_A}{dt} = \frac{V_A - i_A R_A - v_{Rc}}{L_A} \quad (5.12)$$

Ahora aplicando LVK en la malla 2 del circuito mostrado en la Figura 5.5 se tiene:

$$\frac{di_B}{dt} = \frac{v_{Rc} - i_B R_B - V_B}{L_B} \quad (5.13)$$

donde:

$$\frac{d\lambda_m}{dt} = v_{Rc} = R_c \cdot (i_A - i_B - i_m(\lambda)) \quad (5.14)$$

Por lo que el efecto de saturación es representado por la Ecuación (5.8) o por la Ecuación (5.9).

### 5.2.6. Horno de arco eléctrico

En la Figura 5.6 se muestra la conexión de un horno de arco eléctrico al nodo A. En la presente tesis se utiliza el modelo del horno de arco eléctrico propuesto en [Acha *et al.*, 1990]. Este modelo está basado en el principio de conservación de la energía y es representado por la Ecuación (5.15).

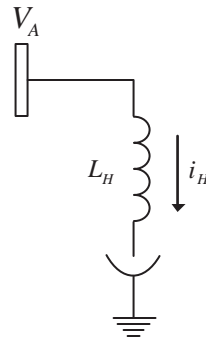


Figura 5.6: Modelo del Horno del Arco Eléctrico

$$P_1 + P_2 = P_3 \quad (5.15)$$

donde:

$P_1$  representa la potencia transmitida en forma de calor al ambiente externo.

$P_2$  representa la potencia que incrementa la energía interna en el arco y la cual por tanto afecta su radio.

$P_3$  representa la potencia total desarrollada en el arco y convertida en calor.

En la Ecuación 5.15 se considera que el efecto de enfriamiento es únicamente una función del radio  $r$  del arco. Por lo tanto:

$$P_1 = k_1 r^n \quad (5.16)$$

De hecho, también es función de la temperatura del arco, esta dependencia se considera menos insignificante y por lo tanto es ignorada con el fin de mantener el modelo simplificado. Por lo tanto, únicamente el radio  $r$  aparece como variable de estado y  $k_1$  es una constante de enfriamiento. Si el ambiente alrededor del arco es caliente el enfriamiento del arco no depende de su radio por completo, así que en este caso  $n = 0$ . Si este no es el caso y el arco es grande, entonces el área de enfriamiento es principalmente su superficie lateral, así que  $n = 1$ . Si el arco es pequeño, entonces el área de enfriamiento es proporcional a el área de sección transversal en los electrodos, así que  $n = 2$  [Acha *et al.*, 1990].

El término  $P_2$  es proporcional a la derivada de la energía dentro del arco, la cual es proporcional a  $r^2$ .

$$P_2 = k_2 r \frac{dr}{dt} \quad (5.17)$$

Finalmente:

$$P_3 = v \cdot i = \frac{k_3}{r^m} i^2 \quad (5.18)$$

En esta expresión la resistividad de la columna del arco se considera que es inversamente proporcional a  $r^m$ , donde  $m = 0.2$ , para reflejar el hecho de que el arco puede ser más caliente en su interior si tiene un radio grande.

Sustituyendo (5.16), (5.17) y (5.18) en (5.15), se obtiene la ecuación diferencial del arco:

$$k_1 r^n + k_2 r \frac{dr}{dt} = k_3 r^{-(m+2)} i^2 \quad (5.19)$$

El voltaje en el arco está dado por:

$$v = k_3 r^{-(m+2)} i \quad (5.20)$$

Ahora considerando el circuito de la Figura 5.6. Por medio de la aplicación de LVK se tiene:

$$V_A = V_{LH} + V_{RH} \quad (5.21)$$

donde:

$$V_{LH} = L_H \frac{di_H}{dt} \quad (5.22)$$

$$V_{RH} = k_3 r^{-(m+2)} i_H \quad (5.23)$$

Sustituyendo (5.22) y (5.23) en (5.21) y despejando el término  $\frac{di_H}{dt}$  se tiene que:

$$\frac{di_H}{dt} = \frac{-k_3 r^{-(m+2)} i_H + V_A}{L_H} \quad (5.24)$$

y la derivada del radio del arco está dada por:

$$\frac{dr}{dt} = \frac{k_3}{k_2} r^{-(m+3)} i_H^2 - \frac{k_1}{k_2} r^{(n-1)} \quad (5.25)$$

### 5.2.7. Dispositivos FACTS

Los dispositivos FACTS (*Flexible AC Transmission Systems*) surgen de la aplicación de la electrónica de potencia en los sistemas de transmisión [Hingorani y Gyugyi, 2000]. La principal propuesta de esta tecnología consiste en controlar y regular las variables eléctricas en sistemas de potencia. Esto se logra usando convertidores como una conexión controlable entre terminales del sistema de potencia. La representación del convertidor resultante puede ser útil para una variedad de configuraciones. Básicamente la familia de dispositivos FACTS basados en *Voltage Source Converters* (VSC) constan de un compensador serie (SSSC), un compensador *shunt* (STATCOM) y un compensador serie/*shunt* (UPFC).

Para entender y controlar la interacción entre los dispositivos FACTS basados en VSC y los sistemas utilizados, existe la necesidad de aproximar modelos para obtener resultados rápidos y precisos. Existen varios modelos de los dispositivos FACTS basados en VSCs ideales [Uzunovic *et al.*, 1997], [Nabavi y Iravani, 1996], [Tavakoli y Hamill, 2005] y también se pueden encontrar en [Uzunovic *et al.*, 1998] y [Cañizares, 2000]. La aproximación de modelos de FACTS es utilizada frecuentemente debido a su simplicidad. Sin embargo, estos modelos tienen limitaciones evidentes para evaluar la distorsión armónica introducida al sistema de potencia por los VSCs, así como el proceso de *conmutación* en los VSCs.

Cualquier aplicación basada en convertidores FACTS actúa como una fuente de corrientes armónicas inyectadas en el sistema y también interactuando con distorsiones armónicas presentes en el sistema. La representación de la interacción armónica entre estos dispositivos y la red, así como su impacto en el sistema, pueden ser logrados a partir de un modelo matemático adecuado del VSC, el cual puede dar una percepción clara acerca del comportamiento del dispositivo FACTS, así como la posibilidad de determinar la distorsión armónica producida por

estos dispositivos y para asegurar su impacto adverso en la calidad de la energía. Sin embargo, es difícil analizar la dinámica de sistemas VSC, ya que ellos incorporan dinámicas de tiempo continuo y eventos en tiempo discreto. Estas dificultades analíticas están comúnmente en todos los componentes FACTS ya que sus operaciones están basadas en un proceso de conmutación. Los dispositivos de conmutación empleados en estos FACTS están basados en VSC con interruptores semiconductores. El modelado de los dispositivos de conmutación se puede realizar con diferentes niveles de detalle. Un modelo muy detallado puede ser necesario y justificado cuando estamos interesados en estudiar cualquier fenómeno asociado con el proceso de conmutación, pero típicamente el costo computacional es muy alto debido a que el conjunto de EDO que modelan al sistema eléctrico se vuelve rígido, esto quiere decir que las derivadas del conjunto de EDO son muy grandes, requiriendo un método de solución para un conjunto de EDO rígido, ya que se introducen pequeñas constantes de tiempo requiriendo valores muy pequeños para el paso de integración que se está utilizando, y probablemente causando la divergencia del método de solución para el conjunto de EDO's; típicamente el método de Newton-Raphson.

En [Hingorani y Gyugyi, 2000] se clasifican los controladores FACTS en 4 categorías:

- Controladores serie
- Controladores en derivación.
- Controladores combinados serie-serie.
- Controladores combinados serie-derivación.

A continuación se presenta la deducción de la formulación en variables de estado de algunos dispositivos FACTS tales como el Reactor Controlado por Tiristores (RCT), el Compensador Serie Controlado por Tiristores (CSCT) y el Compensador Estático de VARs (STATCOM).

#### 5.2.7.1. Reactor Controlado por Tiristores (RCT)

Considere el modelo del RCT mostrado en la Figura 5.7, el cual está compuesto por un arreglo de tiristores conectados en antiparalelo, que a su vez están conectados en serie a una rama resistiva inductiva [Miller, 1982].

Aplicando LVK en la malla formada tenemos:

$$V_A = V_{R_{RCT}} + V_{L_{RCT}} \quad (5.26)$$

donde:

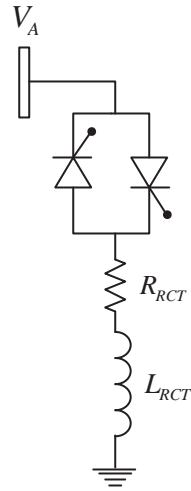


Figura 5.7: Modelo del RCT

$$V_{R_{RCT}} = i_{RCT} \cdot R_{RCT} \quad (5.27)$$

$$V_{L_{RCT}} = L_{RCT} \cdot \frac{di_{RCT}}{dt} \quad (5.28)$$

sustituyendo (5.27) y (5.28) en (5.26) se tiene:

$$V_A = i_{RCT} \cdot R_{RCT} + L_{RCT} \cdot \frac{di_{RCT}}{dt} \quad (5.29)$$

despejando para  $\frac{di_{RCT}}{dt}$  se tiene:

$$\frac{di_{RCT}}{dt} = \frac{V_A - i_{RCT}R_{RCT}}{L_{RCT}} \quad (5.30)$$

Dado que la corriente en el RCT está en función de la conmutación de los tiristores, se tiene:

$$\frac{di_{RCT}}{dt} = S \cdot \left( \frac{V_A - i_{RCT}R_{RCT}}{L_{RCT}} \right) \quad (5.31)$$

donde:

$S$  es una función de conmutación que vale  $S(t) = 0$  ó  $S(t) = 1$  dependiendo del ángulo de disparo de los tiristores.



### 5.2.7.2. Compensador serie controlado por tiristores (CSCT)

El modelo en el dominio del tiempo del compensador serie controlado por tiristores CSCT utilizado en el presente trabajo está reportado en [Hak *et al.*, 1997] y [Medina *et al.*, 2003].

Considere el arreglo propuesto para el CSCT, tal como se muestra en la Figura 5.8. El modelo está formado por una resistencia conectada en serie con una inductancia, que representan a la línea de transmisión, en serie con este elemento se tiene un arreglo en paralelo formado por una rama con un capacitor y una segunda rama constituida por un arreglo serie de una resistencia, una inductancia y dos tiristores conectados en antiparalelo.

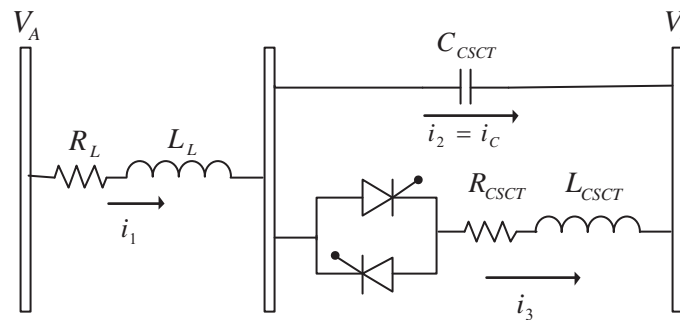


Figura 5.8: Modelo del CSCT

Considere las mallas mostradas en la Figura 5.9.

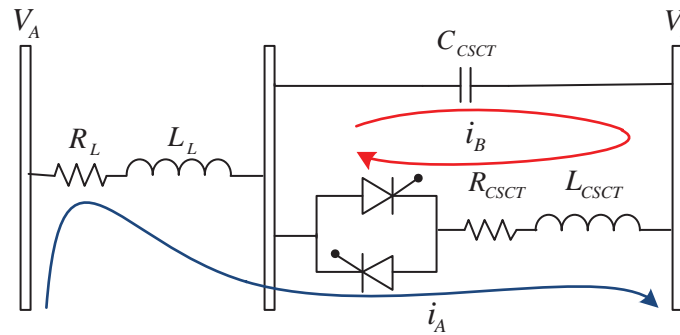


Figura 5.9: Mallas formadas en el modelo del CSCT para su análisis

Por medio de la aplicación de LVK se tiene:

$$V_A - V_R - V_L - V_{CSCT} - V_B = 0 \quad (5.32)$$

donde:

$V_A$  Voltaje en el nodo A.

$V_B$  Voltaje en el nodo B.

$V_{CSCT}$  Voltaje en el capacitor C.

$$V_R = i_1 R_L \quad (5.33)$$

$$V_L = L_L \frac{di_1}{dt} \quad (5.34)$$

sustituyendo (5.32) y (5.33) en (5.32) se tiene:

$$V_A - V_B - i_1 R_L - L_L \frac{di_1}{dt} - V_{CSCT} = 0 \quad (5.35)$$

despejando  $\frac{di_1}{dt}$

$$\frac{di_1}{dt} = \frac{V_A - V_B - i_1 R_L - V_{CSCT}}{L_L} \quad (5.36)$$

Por medio de la aplicación de LVK en la malla B, se tiene:

$$S \cdot V_{CSCT} = i_3 R_{CSCT} + L_{CSCT} \frac{di_3}{dt} \quad (5.37)$$

despejando  $\frac{di_3}{dt}$  se tiene:

$$\frac{di_3}{dt} = \frac{-i_3 R_T + S \cdot V_{CSCT}}{L_T} \quad (5.38)$$

finalmente:

$$\frac{dV_{CSCT}}{dt} = \frac{i_C}{C} \quad (5.39)$$

donde:

$$i_C = i_1 - i_3 \quad (5.40)$$

por lo que:

$$\frac{dV_{CSCT}}{dt} = \frac{i_1 - S \cdot i_3}{C} \quad (5.41)$$

### 5.2.7.3. Función de conmutación en dispositivos FACTS

El funcionamiento de los dispositivos FACTS se basa en los periodos de conducción y no conducción de un par de Tiristores conectados en antiparalelo, los cuales a su vez son conectados a

elementos lineales tales como resistencias, inductores y capacitores, con el objeto de modificar sus características de operación. Estos periodos de conducción se representan en este trabajo mediante el uso de una función de conmutación  $S(t)$ . Esta función puede tomar los valores de:

$S(t) = 1$  cuando los tiristores están en conducción.

$S(t) = 0$  cuando los tiristores no conducen.

La Figura 5.10 muestra la función de conmutación de la corriente en el tiristor. El inicio del periodo  $t_0$  es el cruce por cero de la corriente. El instante  $t_1$  y  $t_3$  son los inicios de conducción de la corriente del tiristor determinados por medio de los ángulos de disparo  $\sigma_1$  y  $\sigma_2$ , respectivamente.

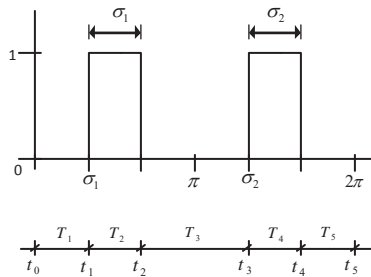


Figura 5.10: Función de conmutación

Los instantes  $t_2$  y  $t_4$  son los finales de conducción de la corriente del tiristor, conmutados naturalmente para cada medio periodo. El final del periodo  $t_5$  es el próximo cruce por cero de la corriente de línea. De la Figura 5.10 se aprecia que la salida de esta función es un número discreto 0 ó 1, el cual se multiplica por la o las variables que dependan del estado de encendido o apagado de los tiristores. Esta función básicamente lo que hace es monitorear la corriente en los tiristores además del voltaje en las terminales del dispositivo FACTS, para establecer el valor de salida de dicha corriente. Dos datos adicionales que requiere esta función son: el valor del ángulo de disparo de los tiristores y el tiempo. La Figura 5.11 ilustra el diagrama de flujo utilizado en la programación de la función de conmutación, donde  $V$  e  $I$  son el voltaje y corriente respectivamente en el tiristor.

### 5.2.8. Compensador estático de VARs (STATCOM)

Para efectos prácticos el compensador estático de VAR's (SVC) basado en tiristores ha hecho redundante al compensador síncrono rotatorio, excepto donde el incremento del nivel de corto circuito se requiere a lo largo del sistema con soporte de potencia reactiva de acción rápida [Miller, 1982]. Sin embargo, la tecnología en electrónica de potencia continúa en desarrollo,

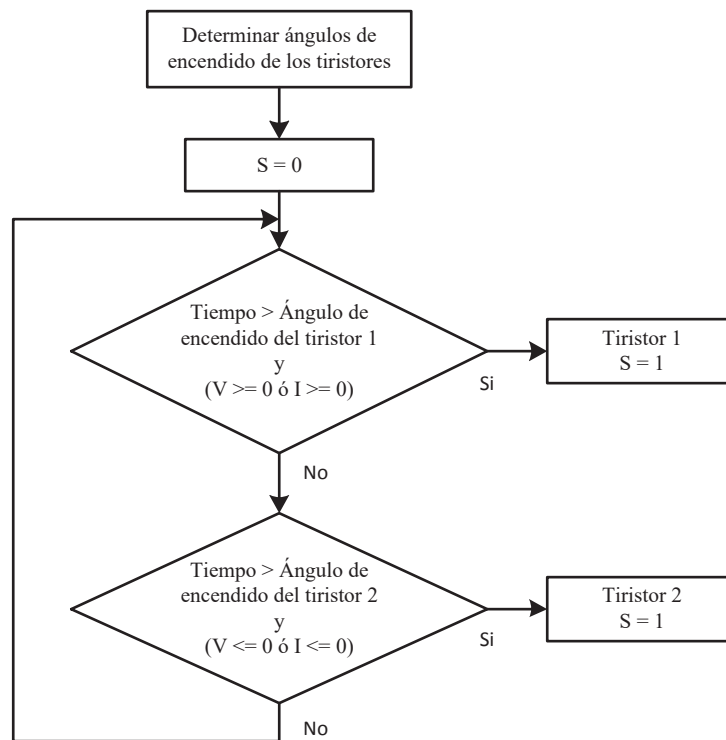


Figura 5.11: Diagrama de Flujo de la función de conmutación

por lo que es inminente el reemplazo del SVC por una nueva familia de compensadores estáticos basados en el uso de convertidores de fuente de voltaje (VSC). Estos son conocidos como STATCOMs (compensadores estáticos de VARs) y proveen todas las funciones que el SVC puede dar pero en alta velocidad; éste es más compacto y requiere solo una fracción del terreno requerido para la instalación de un SVC. El STATCOM es esencialmente un VSC interconectado al sistema de CA a través de un transformador conectado en derivación. El VSC es la base de la construcción de la nueva generación de controladores de electrónica de potencia que han emergido de las iniciativas FACTS y *Custom Power*.

Por lo tanto, el compensador estático de VARs (STATCOM) es un condensador síncrono de estado sólido conectado en derivación (*shunt*) con el sistema de CA. La corriente de salida es ajustada para controlar tanto el voltaje nodal como al potencia reactiva inyectada en el bus al cual está conectado.

En la Figura 5.12 se muestra el modelo de STATCOM conectado al nodo A, en esta tesis se utiliza el modelo propuesto en [Segundo y Medina, 2009].

Aplicando LVK se tiene que:

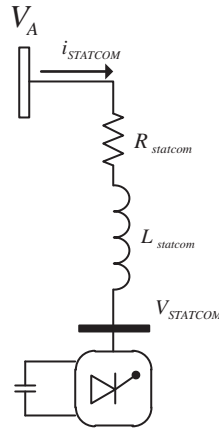


Figura 5.12: Modelo del STATCOM

$$V_A = V_{R_{statcom}} + V_{L_{statcom}} + V_{STATCOM} \quad (5.42)$$

Resolviendo para la corriente que circula por la malla que se forma, despejando  $\frac{di_{STATCOM}}{dt}$  se tiene que:

$$\frac{di_{STATCOM}}{dt} = \frac{V_A - i_{STATCOM}(t) \cdot R_{statcom} - V_{STATCOM}}{L_{statcom}} \quad (5.43)$$

donde:

$V_{STATCOM}$  es el voltaje en terminales del STATCOM y se obtiene a través de la funciones de conmutación propuestas en [Segundo y Medina, 2009].

### 5.2.8.1. Convertidor de fuente de voltaje de seis pulsos

En la Figura 5.13 se muestra la representación de un convertidor de seis pulsos. La función de conmutación bidireccional se identifica por  $S_i$  y  $S'_i$  para cada fase ( $i = a, b, c$ ) que puede estar en modo encendido o apagado,  $r$  es la resistencia en estado de encendido,  $S_i$  es 1 ó 0 correspondiente al estado de encendido o apagado de los switches respectivamente. Además,  $S_i$  y  $S'_i$  son complementarias lo que significa que  $S_i + S'_i = 1$ .

para describir el modelo matemático de este convertidor consideremos la fase A el lado de CD mostrado en la Figura 5.14.

La ecuación que describe el voltaje  $v_{ta}$  se puede escribir como:

$$v_{ta} = v_{aH} + v_{Hn} \quad (5.44)$$

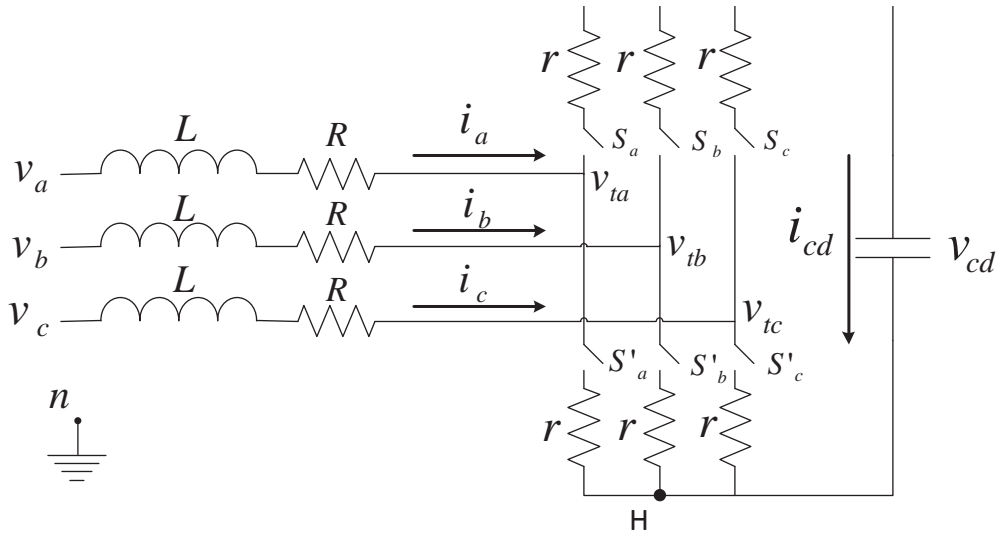


Figura 5.13: Convertidor trifásico de seis pulsos

Cuando  $S_a$  vale 1 entonces tenemos:

$$v_{aH} = (r \cdot i_a + v_{cd}) \cdot S_a \quad (5.45)$$

Por otro lado, cuando tenemos que  $S_a$  está apagada, entonces tenemos que:

$$v_{aH} = r \cdot i_a \cdot S'_a \quad (5.46)$$

ya que  $S_a$  y  $S'_a$  son complementarias, tenemos:

$$v_{aH} = r \cdot i_a + v_{cd} \cdot S_a \quad (5.47)$$

Para las dos fases restantes se encuentran ecuaciones similares:

$$v_{bH} = r \cdot i_b + v_{cd} \cdot S_b \quad (5.48)$$

$$v_{cH} = r \cdot i_c + v_{cd} \cdot S_c \quad (5.49)$$

ya que  $v_{ta} + v_{tb} + v_{tc} = 0$  e  $i_a + i_b + i_c = 0$  obtenemos:

$$v_{Hn} = \frac{v_{cd}}{3} (S_a + S_b + S_c) \quad (5.50)$$

Finalmente, el voltaje en terminales del STATCOM puede ser expresado como:

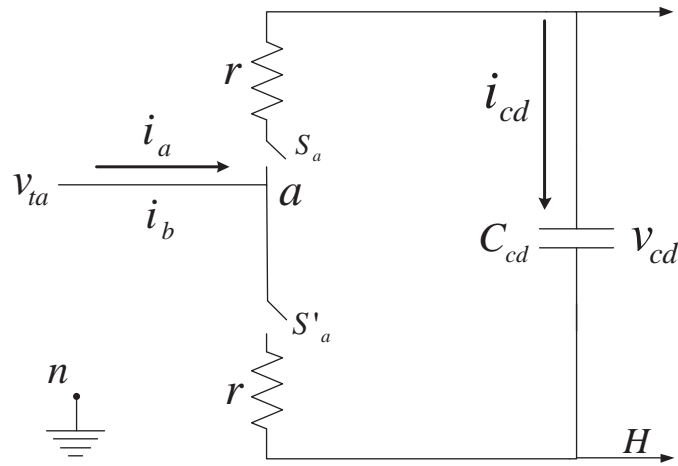


Figura 5.14: Circuito equivalente para la fase A

$$\begin{aligned}
 v_{ta} &= r \cdot i_a + v_{cd} \cdot S_a - \frac{v_{cd}}{3} (S_a + S_b + S_c) \\
 v_{tb} &= r \cdot i_b + v_{cd} \cdot S_b - \frac{v_{cd}}{3} (S_a + S_b + S_c) \\
 v_{tc} &= r \cdot i_c + v_{cd} \cdot S_c - \frac{v_{cd}}{3} (S_a + S_b + S_c)
 \end{aligned} \tag{5.51}$$

Considerando un solo brazo del convertidor como el que se muestra en la Figura 5.14. La señal deseada (señal de referencia) es  $v_s$ , y el voltaje de CD a través del capacitor es  $v_{cd}$ . La señal de referencia es definida como:

$$v_s = E_s \cdot \cos(\omega \cdot t + \alpha) \tag{5.52}$$

Donde  $E_s$  es la magnitud de  $v_s$  y  $\alpha$  es su ángulo de fase. La propuesta en [Segundo y Medina, 2009] es transformar la señal continua  $v_s$ , en una serie de pulsos teniendo una amplitud fija de  $v_{cd}$ . Para lograr este resultado, se dibuja una forma de onda triangular (*tri*) con una magnitud  $v_{cd}$ . Ahora la conducción toma lugar mientras  $v_s$  permanezca sobre la forma triangular (*tri*) y cesa cada vez que esta se encuentra por debajo, por lo tanto tenemos:

$$\begin{aligned}
 v_{sa} &= E_s \cdot \cos(\omega \cdot t + \alpha) \\
 v_{sb} &= E_s \cdot \cos\left(\omega \cdot t + \alpha - \frac{2\pi}{3}\right) \\
 v_{sc} &= E_s \cdot \cos\left(\omega \cdot t + \alpha + \frac{2\pi}{3}\right)
 \end{aligned} \tag{5.53}$$

### 5.2.8.2. Modelo de la tangente hiperbólica para la función de conmutación

La señal de conmutación  $S_s$  mostrada en la Figura 5.15 satisface la propuesta de evitar discontinuidades. La Figura 5.15(a) muestra a  $S_s$  contra el tiempo y la Figura 5.15(b) muestra a  $S_s$  contra

$$u = v_s - tri.$$

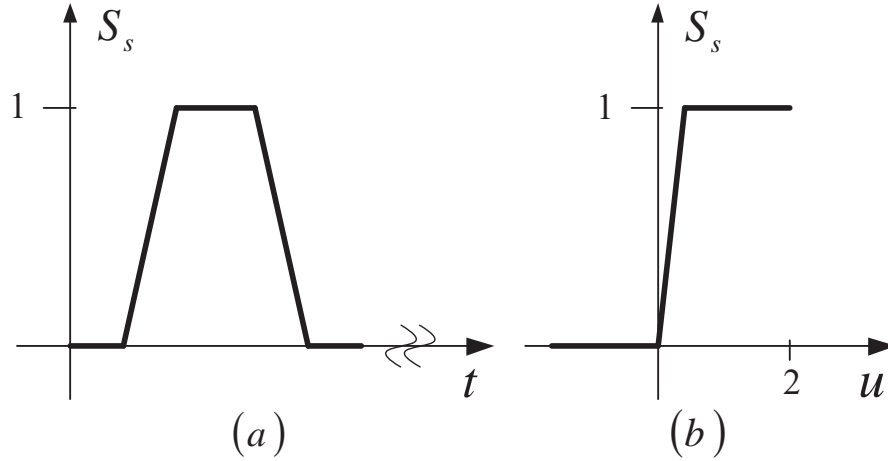


Figura 5.15: Función continua de conmutación  $S_s$

La función de conmutación  $S_s$  mostrada en la Figura 5.15 es continua: sin embargo, esta no es diferenciable en las esquinas. Para resolver este problema la función puede ser representada por una función más suave. Una función continua y diferenciable que satisface estas condiciones es la función tangente hiperbólica:

$$S_{si} = \frac{\tanh(\alpha(v_{si} - tri)) + 1}{2} \quad (5.54)$$

$$tri = \frac{2}{\pi} \operatorname{asen}(\operatorname{sen}(2 \cdot \pi \cdot m_f \cdot f \cdot t + \pi)) \quad (5.55)$$

donde:

$v_{si}$  es la señal de control,  $i = a, b, c$ .

$tri$  es una forma de onda triangular.

$\alpha$  es la pendiente máxima de  $S_{si}$ .

### 5.2.9. Modelo de cargas

Ahora se muestra el modelo de los diferentes tipos de cargas que pueden ser conectadas a los diversos nodos del SEP. Los modelos de cargas utilizados son: resistiva, inductiva y resistiva-inductiva-capacitiva.



### 5.2.9.1. Carga resistiva

En la Figura 5.16 se muestra la conexión de una carga completamente resistiva al nodo A que es parte del SEP.

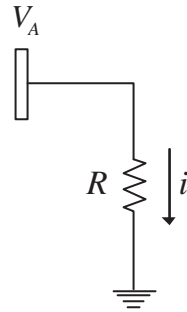


Figura 5.16: Modelo de carga resistiva

Aplicando la ley de Ohm se obtiene la corriente que circula a través de la resistencia y que fluye del nodo A hacia tierra.

$$i = \frac{V_A}{R} \quad (5.56)$$

La carga resistiva no genera una variable de estado, sin embargo este elemento genera una extracción de corriente  $i$  en el nodo donde se conecta.

### 5.2.9.2. Carga inductiva

La Figura 5.17 muestra la conexión de una carga inductiva al nodo A de valor  $L$ .

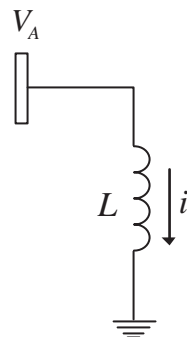


Figura 5.17: Modelo de carga inductiva

Aplicando LVK al circuito de la Figura 5.17 y despejando  $\frac{di}{dt}$ , se tiene:

$$\frac{di}{dt} = \frac{V_A}{L} \quad (5.57)$$

en contraste con la carga puramente resistiva, las cargas inductivas si generan una variable de estado adicional.

### 5.2.9.3. Carga resistiva-inductiva-capacitiva

La Figura 5.18 muestra el modelo de una conexión serie de una carga resistiva, una carga inductiva y una carga capacitiva conectadas al nodo A de un SEP.

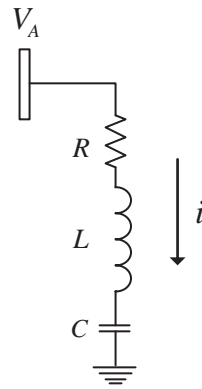


Figura 5.18: Modelo de carga resistiva-inductiva-capacitiva

Aplicando LVK se tiene que:

$$\frac{di}{dt} = \frac{-i(t) \cdot R - V_C + V_A}{L} \quad (5.58)$$

y también se tiene que:

$$\frac{dV_C}{dt} = \frac{i}{C} \quad (5.59)$$

que son las dos variables de estado que introduce una rama RLC conectada a un nodo que es parte del SEP.

### **5.3. Conclusiones**

El modelado en el dominio de tiempo que se está realizando de cada uno de los elementos que integran al SEP y que son utilizados en los diferentes casos de estudio, permite la generación automática del conjunto de ecuaciones diferenciales en forma simplificada con la finalidad de aplicar alguno de los métodos de solución del conjunto de EDO y finalmente obtener un historial de la dinámica que presenta el SEP a lo largo del tiempo. El modelado que se está realizando tiene una forma simplificada, por lo que si se requiere un estudio más detallado del comportamiento de uno o más elementos del sistema, entonces estará justificado el esfuerzo adicional requerido para desarrollar un modelado detallado de los elementos del sistema eléctrico de potencia.

# Capítulo 6

## Casos de estudio

### 6.1. Introducción

La dinámica de los sistemas eléctricos puede representarse adecuadamente mediante un conjunto de EDO. Este conjunto de ecuaciones, generalmente se presenta mediante notación vectorial-matricial también conocida como *espacio de estado*. La notación matricial, simplifica la representación matemática de los sistemas de ecuaciones y el aumento en el número de variables de estado de entrada o salida no aumenta la complejidad de las ecuaciones [Ogata, 1993].

La razón de emplear el espacio de estado en el análisis y diseño de sistemas eléctricos es consecuencia de la necesidad de manejar un número muy grande de ecuaciones diferenciales ordinarias de una manera organizada y eficiente. Intrínseco a un espacio de estado están tres tipos de variables: *variables de entrada*, *variables de estado* y *variables de salida*, todos generalmente vectores; el espacio de estado identifica la dinámica e interacción de las variables de estado [Raymod y DeCarlo, 1989].

En el presente capítulo se utilizan los conceptos de espacio de estado y variables de estado para representar la dinámica de los diferentes casos de estudio que aquí se presenta, ya que la representación en espacio de estado es necesaria para aplicar el procesamiento en paralelo de grano fino al método de integración de Runge-Kutta de cuarto orden.

Finalmente se aplica la formulación analítica de *espacio de estado* en seis casos de prueba, los cuales incorporan distintos componentes lineales, no lineales y variantes en el tiempo, donde el conjunto de EDO generados para cada caso se resuelven mediante la aplicación repetida del método de integración de Runge-Kutta de cuarto orden comúnmente conocido como método de FB. Los resultados obtenidos en las simulaciones se comparan en términos del tiempo requerido para obtener el *ciclo límite* o estado estacionario periódico de operación en forma secuencial y utilizando procesamiento en paralelo.

## 6.2. Metodología de solución

El análisis de espacio de estado es un procedimiento que se aplica a los circuitos eléctricos que incorporan elementos variantes en el tiempo [Hayt y Kemmerly, 1993]. Estos circuitos contienen un conjunto de condiciones iniciales en  $t_0$  para las variables de interés las cuales se denominan *variables de estado*, y por lo tanto a este conjunto de condiciones iniciales se les denomina *estados* [Chen, 1984].

El conocimiento de las variables de estado en  $t_0$ , determina el comportamiento del sistema eléctrico para todo  $t \geq t_0$  [Chen, 1984]. Las variables de estado que constituyen el conjunto mínimo de variables que determinan el estado de un sistema dinámico. Si se requieren  $n$  variables de estado para describir completamente el comportamiento de un sistema dado, entonces se pueden considerar a las  $n$  variables como  $n$  componentes del vector  $x$ , el cual recibe el nombre de *vector de estados*.

En esta tesis, las variables de estado seleccionadas para el análisis de los diferentes casos de estudio son las corrientes a través de cada uno de los inductores y los voltajes de los capacitores conectados en los nodos, y en el caso donde se cuente con un horno de arco eléctrico se considera como una variable de estado al radio del arco eléctrico. Cada una de estas variables pueden ser utilizadas en forma directa para expresar la energía almacenada por el inductor o el capacitor en cualquier instante de tiempo.

El conjunto de ecuaciones de estado o sistema de EDO que modelan a cada una de las redes eléctricas de prueba se resuelven mediante el método convencional de FB utilizando el método de integración de Runge-Kutta de cuarto orden y técnicas de procesamiento en paralelo. Los resultados se muestran en tablas, donde se presentan distintos valores que corresponden al máximo error calculando durante el proceso de convergencia de las variables de estado y el *speedup* que se obtiene al aplicar procesamiento en paralelo con diferente número de procesadores para cada red eléctrica. En esta tesis se especifica un error menor de  $1 \times 10^{-10}$  p.u entre valores sucesivos de  $x_n$  como criterio de convergencia, esto representa la precisión con la cual se localiza el *ciclo límite*. Se utiliza este criterio de convergencia con el objetivo asegurar que los niveles de oscilación de voltaje y corriente en valores reales se encuentren dentro de los estándares de calidad, ya que un criterio de convergencia mayor podría colocarnos en los límites de estos estándares. Los resultados obtenidos con el método de FB se comparan con el método de FB utilizando procesamiento en paralelo, en términos de eficiencia computacional y tiempo para obtener el *ciclo límite* o estado estacionario de operación. Así también, para cada una de las redes eléctricas se muestra el comportamiento de diversas variables de estado en el dominio del tiempo, de las cuales se realiza un análisis adicional para determinar su contenido armónico aplicando la Transformada Rápida de Fourier (*FFT* por sus siglas en inglés) [Press *et al.*, 2007], y este se



Figura 6.1: CPU E54045 Intel (R) Xeon (R)

muestra mediante una gráfica de barras con la magnitud de los diferentes órdenes armónicos. La computadora utilizada para las pruebas es una Intel (R) Xeon (R) CPU E54045 de dos procesadores con cuatro núcleos cada uno con una velocidad de procesamiento de 2.0 GHz, Figura 6.1.

Todas las redes eléctricas de prueba en esta tesis cuentan con uno o más generadores eléctricos de corriente alterna representados por fuentes de voltaje senoidales  $v_s$  de valor 1.0 *p.u.*, los nodos se indican mediante barras pequeñas con un número asociado, las líneas o ramas se indican mediante un resistencia  $R_L$  y una inductancia  $L_L$ , los hornos de arco eléctrico se representan mediante una inductancia y un arco considerados como cargas. Todas las redes eléctricas de prueba cuentan con capacitores fijos, los cuales están conectados en derivación a cada nodo con el fin de asignar una variable de estado que nos represente el voltaje en cada nodo de la red. Los datos de cada uno de caso de estudio se pueden encontrar en apéndice A.

### 6.3. Caso de estudio 1. Sistema 3 nodos

Con la finalidad de presentar un caso de estudio donde el número de variables de estado sea pequeño y que pueda ser fácilmente validado en el software de simulación ATP, se presenta la siguiente red monofásica de tres nodos Figura 6.2. La red está compuesta de tres líneas de transmisión, dos ramas magnetizantes y tres bancos de capacitores. Su representación en espacio de estado está asociado a 9 EDO.

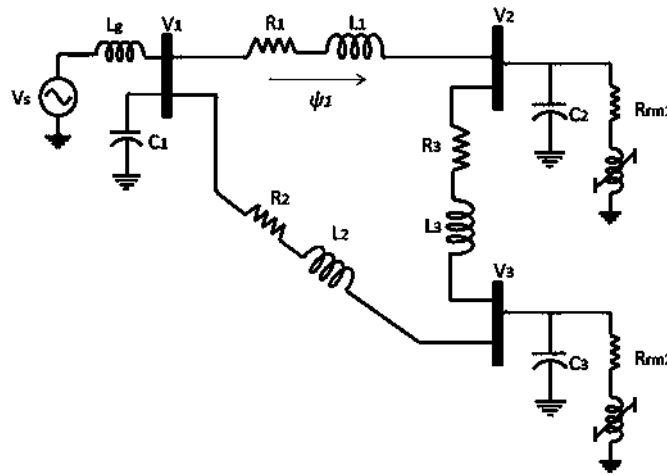


Figura 6.2: Red de prueba de 3 nodos

La Tabla 6.1 muestra el proceso de convergencia hacia el EEP de la red de prueba mostrada en la Figura 6.2. Se aprecia que el método de FB requieren 58 periodos completos de integración (ciclos) necesarios para llegar al EEP en base al criterio de convergencia de  $1 \times 10^{-10}$  especificado en esta tesis, este resultado es el mismo para el método de FB utilizando procesamiento en paralelo.

El proceso de solución en forma secuencial requiere de 0.285 segundos de proceso para llegar al EEP lo que equivale a utilizar un solo elemento de proceso. En la Figura 6.3 se aprecia el comportamiento de la eficiencia relativa o *speedup* obtenida con la aplicación de técnicas de procesamiento en paralelo basado en OpenMP con el uso de 1 a 8 núcleos de procesamiento, este se obtiene a partir de la Tabla 6.2, donde se muestran los tiempos empleados en cada caso para llegar al EEP. Para este caso de estudio se aprecia que el *speedup* disminuye a medida que se aumenta el número de núcleos de procesamiento, esto significa que el tiempo de comunicación de los procesadores supera al tiempo que se ahorra al introducir varios elementos de proceso. Por lo tanto, el aplicar técnicas de procesamiento en paralelo para resolver sistemas muy pequeños no resulta ventajoso.

Tabla 6.1: Proceso de convergencia. Caso de estudio 3 nodos monofásico

<i>Ciclos completos</i>	<i>Error máximo</i>
1	6.487361E-01
2	5.332388E-01
3	4.514516E-01
⋮	⋮
8	1.402911E-02
11	5.341426E-03
15	9.790764E-04
19	2.098005E-04
23	4.307292E-05
26	1.799213E-05
30	3.969362E-06
⋮	⋮
58	7.752640E-11

Tabla 6.2: Valores de *speedup* y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio 3 nodos monofásico.

<i>Número de núcleos</i>	<i>Tiempo de procesamiento (seg)</i>	<i>Speedup</i>
1	0.285	1
2	0.545	0.522
3	0.556	0.512
4	0.610	0.467
5	0.625	0.452
6	0.636	0.446
7	0.658	0.439
8	0.717	0.395

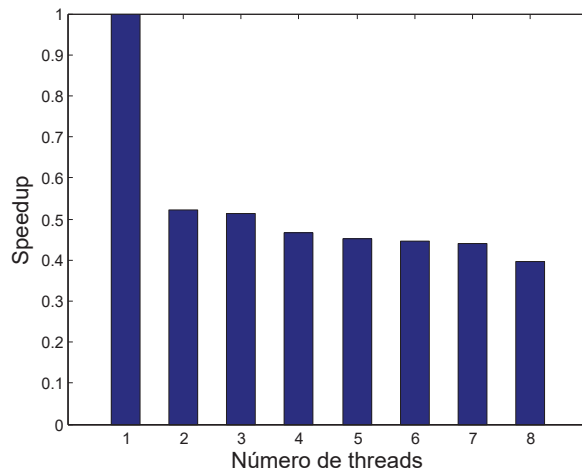


Figura 6.3: Gráfica del *speedup*. Caso de estudio 3 nodos monofásico

Para este caso de estudio se presenta la forma de onda de corriente a través de la línea 1 durante el proceso transitorio hasta alcanzar su estado EEP Figura 6.4, para el cual se presenta un análisis del contenido armónico en diez diferentes criterios de convergencia Figura 6.5.

En la Figura 6.4 se pueden observar los intervalos en los cuales se cumplen los diferentes criterios de convergencia para la obtención del EEP, terminando en un criterio de convergencia de  $1 \times 10^{-10}$ .

En este tipo de estudios es importante conocer el perfil del contenido armónico a lo largo del proceso de solución hasta llegar al EEP. Por esta se obtuvo el contenido armónico de la red



en diez puntos, en los cuales se alcanzan diferentes criterios de convergencia, véase Figura 6.4. Después de obtenerse el contenido armónico para cada punto podemos decir que se mantiene un perfil constante del contenido armónico a partir de una error de convergencia de  $1 \times 10^{-3}$ , como se muestra en la Figura 6.5 en la cual se aprecia una variación mínima en el perfil del contenido armónico durante el resto del estudio.

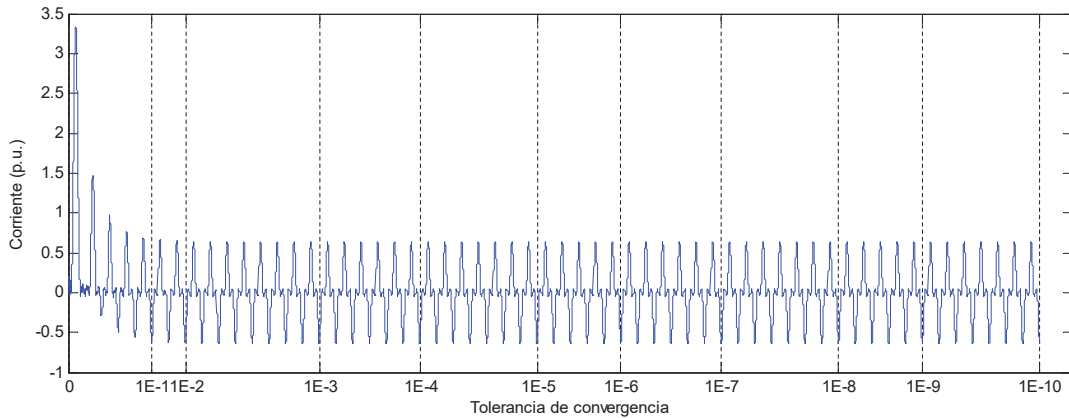
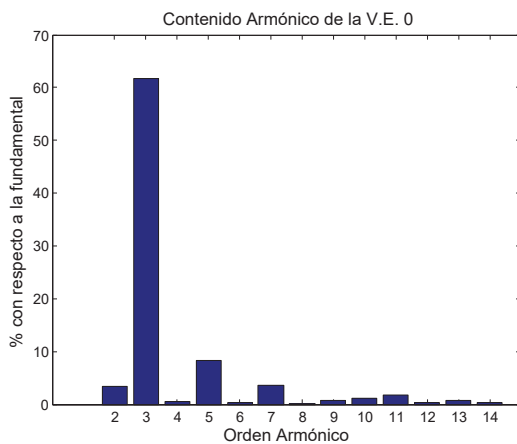
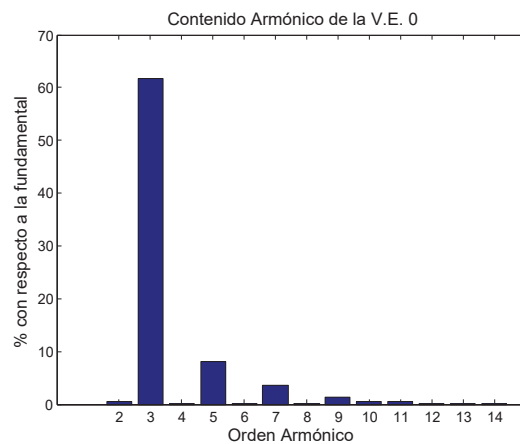


Figura 6.4: Forma de onda de corriente de la línea 1 hasta alcanzar su EEP



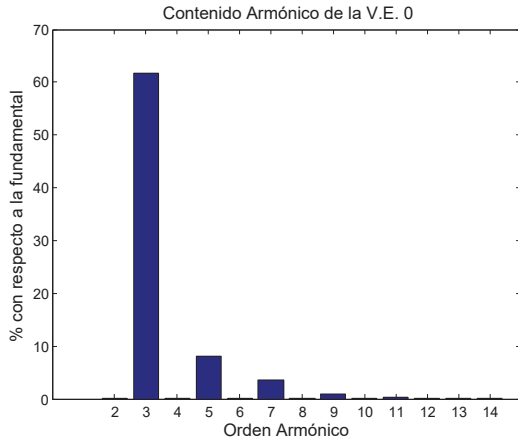
(a) Tolerancia de  $1 \times 10^{-1}$



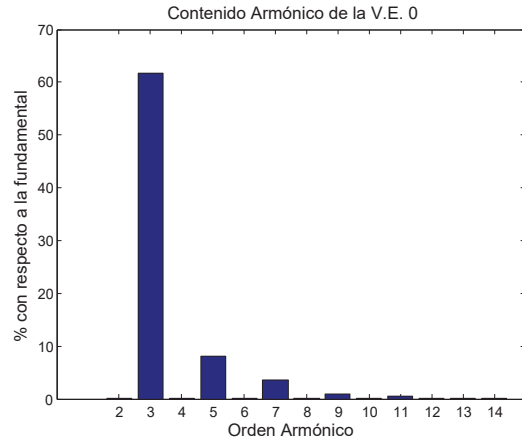
(b) Tolerancia de  $1 \times 10^{-2}$

## CAPÍTULO 6. CASOS DE ESTUDIO

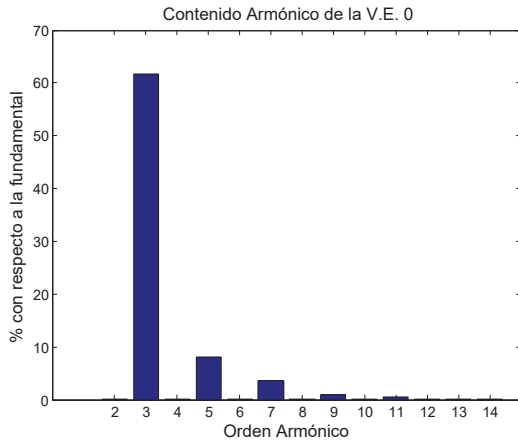
---



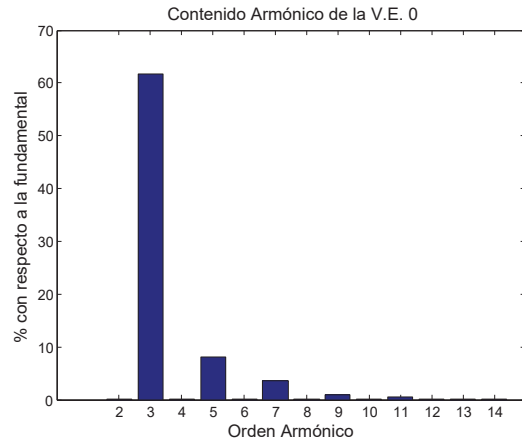
(c) Tolerancia de  $1 \times 10^{-3}$



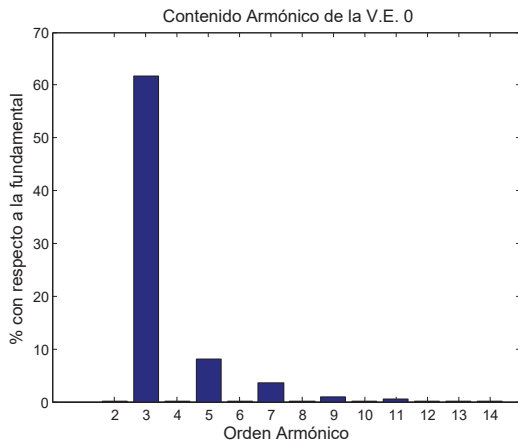
(d) Tolerancia de  $1 \times 10^{-4}$



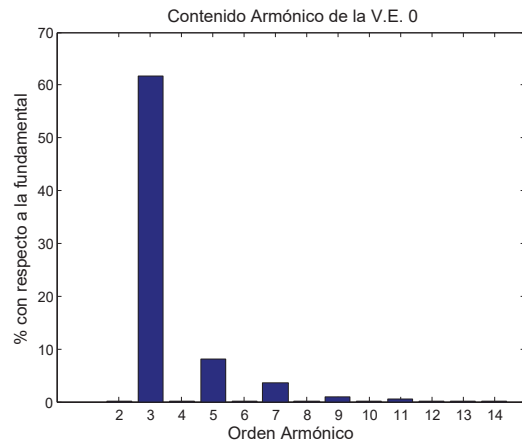
(e) Tolerancia de  $1 \times 10^{-5}$



(f) Tolerancia de  $1 \times 10^{-6}$



(g) Tolerancia de  $1 \times 10^{-7}$



(h) Tolerancia de  $1 \times 10^{-8}$

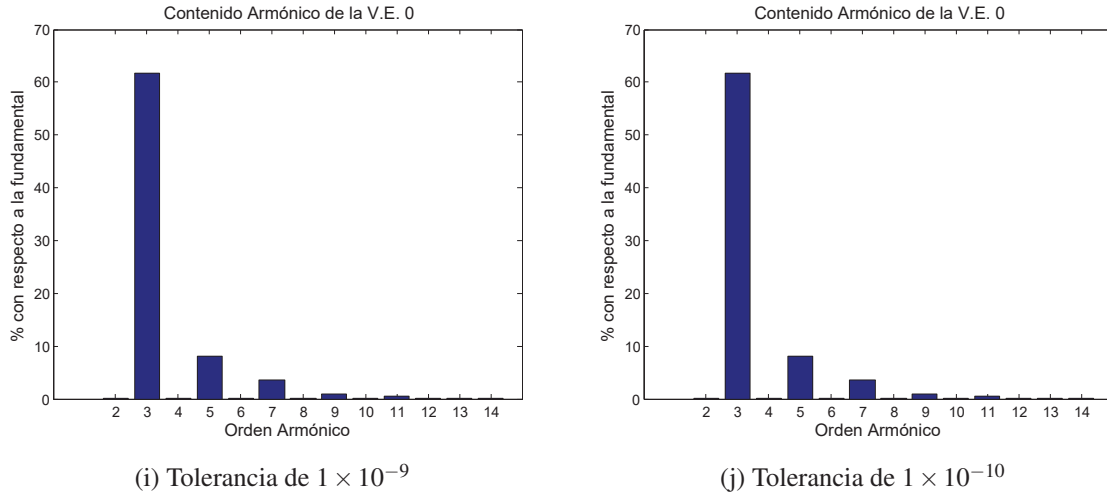


Figura 6.5: Contenido armónico para diferentes criterios de convergencia

Para este caso de estudio se muestra la forma de onda de la corriente en EEP y que fluye a través de la línea 1 Figura 6.6a y su contenido armónico Figura 6.6b, en el cual se puede apreciar que la 3ra armónica representa el 61.67% de la fundamental, la 5ta representa el 8.11% y la 7ma el 3.6%, teniendo un factor de distorsión armónica total de  $THD = 63.31\%$ .

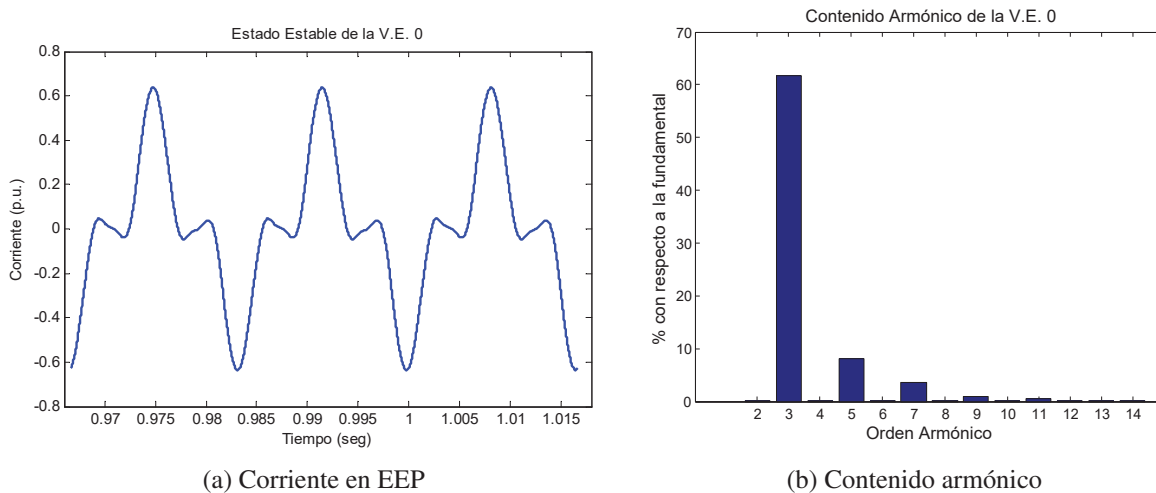


Figura 6.6: Forma de onda de corriente (a) en la línea 1 y su contenido armónico (b)

Con el fin de validar los resultados obtenidos en las simulaciones, se realizó la comparación con el software de simulación ATP Figura 6.7, en el cual se utiliza un modelo aproximado de la rama magnetizante, esto se logra por medio de un polinomio de 5 grado que representa la relación de corriente eléctrica y flujo magnético en la rama, ver Apéndice C.

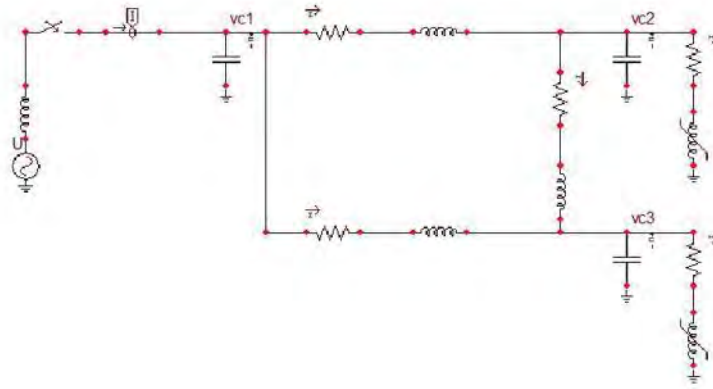


Figura 6.7: Representación del sistema de 3 nodos en ATP

En la Figura 6.8 se muestran las formas de onda de corriente en la línea 1 para el sistema de 3 nodos obtenidas con las simulaciones en C/C++ y las obtenidas en las simulaciones con ATP. En la Figura 6.9 se muestran las formas de onda de voltaje para el nodo 2 en el cual existe una rama magnetizante conectada. La diferencia entre ambas es que la forma de onda obtenida en C/C++ proviene de una función continua, en este caso de un polinomio de grado 5 el cual se presenta en el trabajo de [Dick y Watson, 1981], y la que se obtiene en ATP es una aproximación por medio de un conjunto de puntos experimentales. En el apéndice C se pueden consultar los puntos utilizados para generar la aproximación a un polinomio también de grado 5 expuesto en [Dick y Watson, 1981].

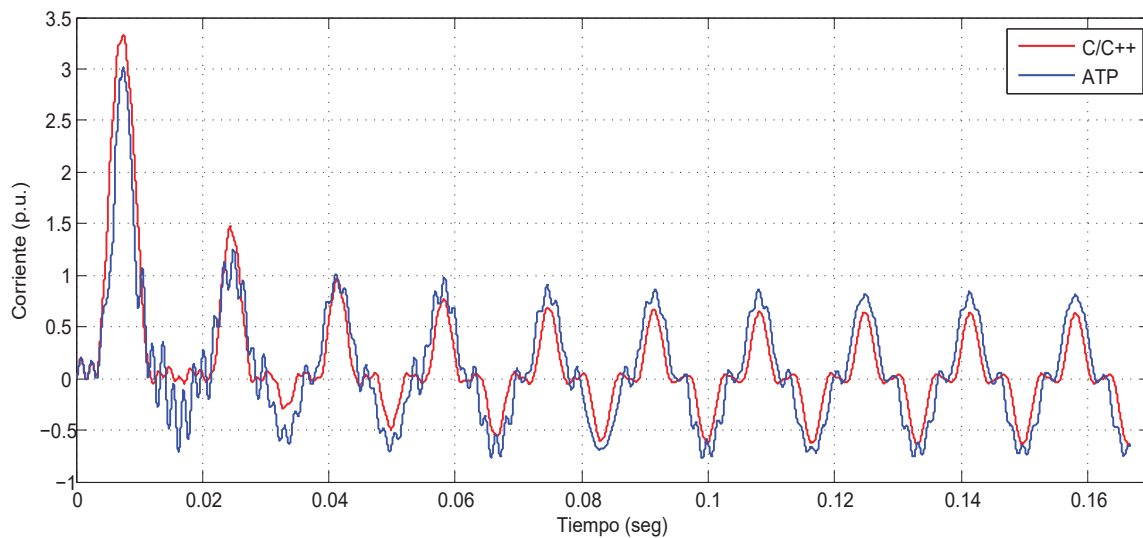


Figura 6.8: Formas de onda de corriente en la línea 1 obtenidas con C/C++ y ATP.

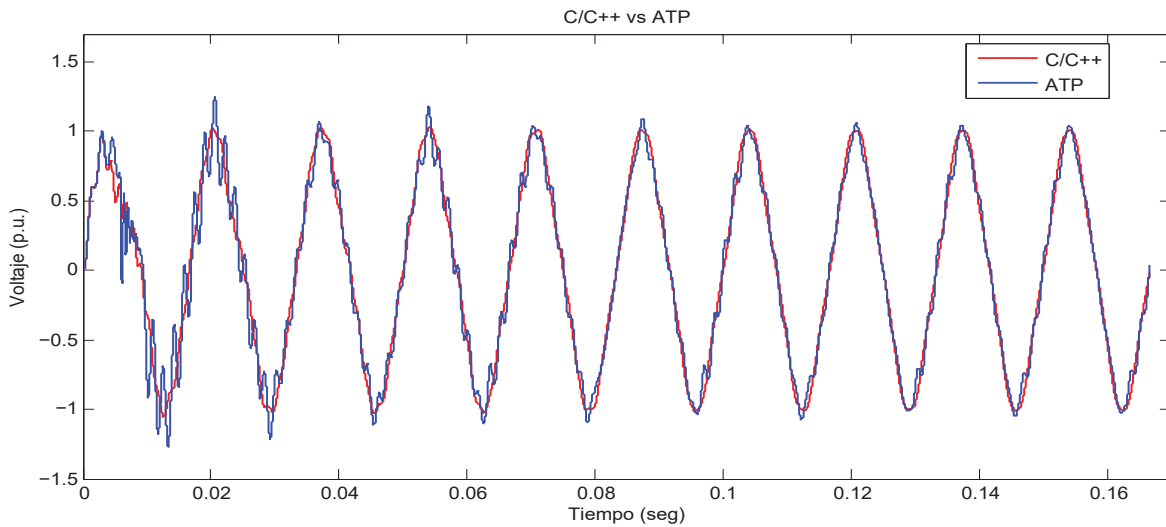


Figura 6.9: Formas de onda de voltaje en el bus 2 obtenidas con C/C++ y ATP

#### 6.4. Caso de estudio 2. Sistema IEEE-14 nodos modificado

Esta red de prueba tiene la topología del sistema IEEE-14 nodos [IEEE test case, 1993]. Sin embargo, ésta ha sido modificada añadiendo ramas magnetizantes para generar distorsión armónica como se muestra en la Figura 6.10. El análisis se realizó para el caso monofásico y caso trifásico, finalmente se añade el modelo del STATCOM propuesto por [?] en el caso trifásico.

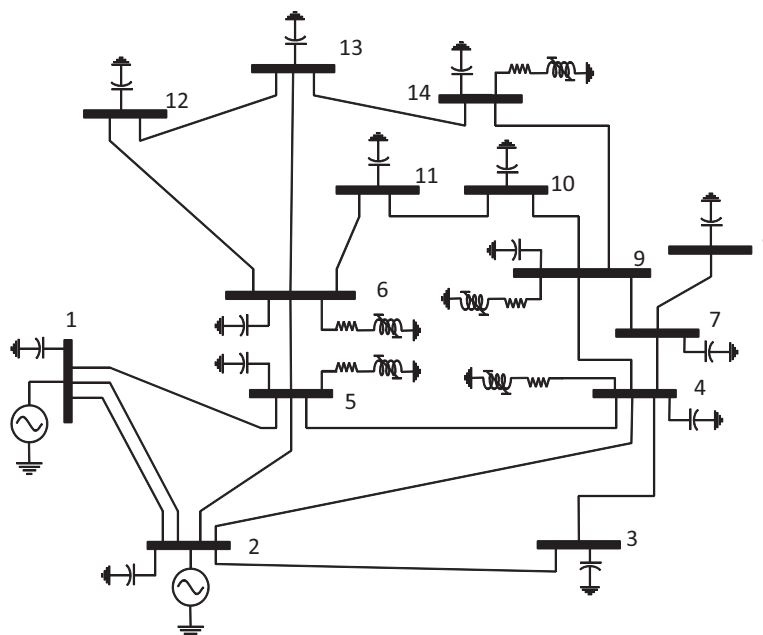


Figura 6.10: Sistema de 14 nodos

Para el caso monofásico la red contiene 20 líneas de transmisión, 14 bancos de capacitores, 5 ramas magnetizantes y 2 generadores. Su presentación en espacio de estado se modela por 41 EDO. El efecto de saturación en los transformadores de potencia está dado por la expresión (5.9). La Tabla 6.3 muestra el proceso de convergencia hacia el EEP, donde se aprecia que al método de FB le toma 68 de periodos completos de integración para llegar al EEP en base al criterio de convergencia especificado, éste resultado es el mismo para el método de FB utilizando procesamiento en paralelo.

Para esta red de prueba se requieren 1.457 segundos de procesamiento secuencial para llegar al EEP. En la Figura 6.11 se aprecia el comportamiento de la eficiencia relativa o *speedup* obtenido con la aplicación del procesamiento en paralelo basado en OpenMP con el uso de 1 a 8 núcleos de procesamiento, en la Tabla 6.4 se muestran los tiempos empleados en cada caso para llegar al EEP, y se aprecia que el máximo *speedup* que se logra es de 1.459 con 6 núcleos, esto quiere decir que con 6 núcleos el EEP se obtiene 45.9% más rápido comparado a la forma secuencial.

Tabla 6.3: Proceso de convergencia. Caso de estudio IEEE-14 nodos monofásico

<i>Ciclos completos</i>	<i>Error máximo</i>
1	7.870307E-01
2	1.787694E+00
3	5.975724E-01
⋮	⋮
16	2.136475E-03
25	7.223948E-05
28	1.655981E-05
34	2.353703E-06
38	3.904539E-07
43	9.837032E-08
51	8.310281E-09
⋮	⋮
68	7.430098E-11

Tabla 6.4: Valores de *speedup* y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-14 monofásico

<i>Número de núcleos</i>	<i>Tiempo de procesamiento (seg)</i>	<i>Speedup</i>
1	1.457	1
2	1.257	1.159
3	1.092	1.334
4	1.031	1.413
5	1.022	1.425
6	0.998	1.459
7	1.021	1.427
8	1.056	1.379

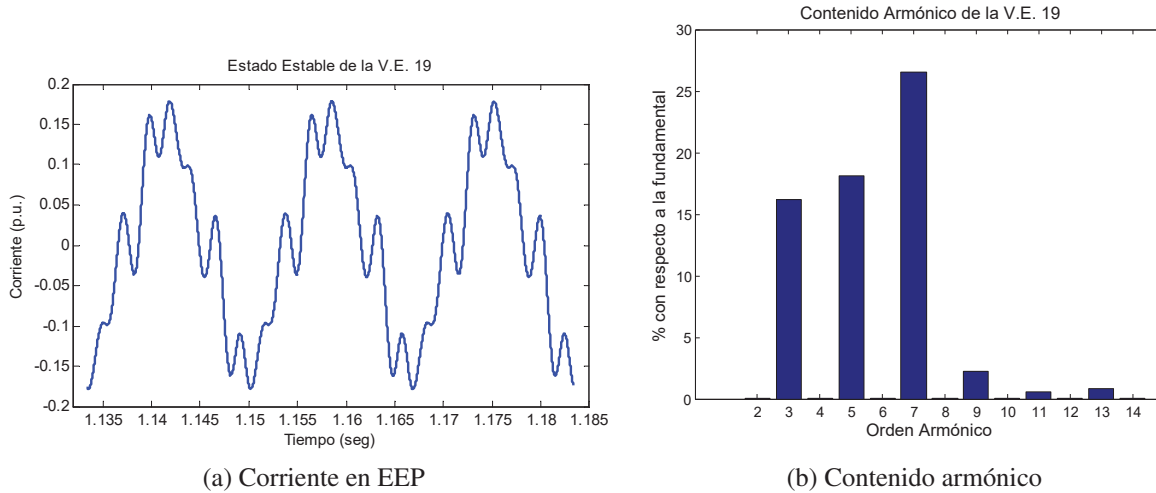


Figura 6.12: Forma de onda en EEP de la corriente en la línea 20 caso monofásico

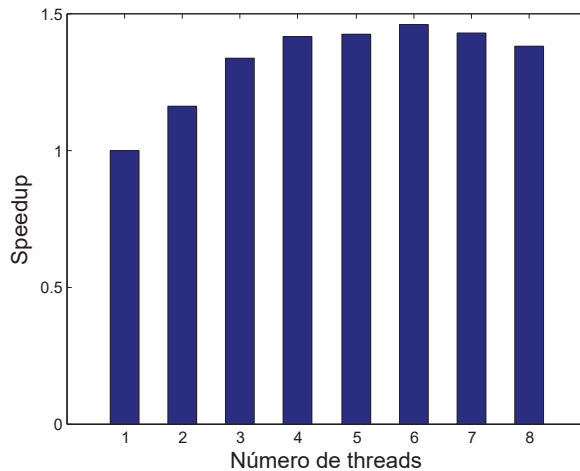


Figura 6.11: Gráfica del *speedup*. Caso de estudio IEEE-14 nodos monofásico

La Figura 6.12a muestra la forma de onda de corriente en EEP que circula a través de la línea 20 que va del nodo 13 al nodo 14 la cual es elegida por estar conectada al nodo más alejado del generador. Adicionalmente en la Figura 6.12b se muestra su contenido armónico, en el cual se puede observar que la 3ra armónica representa el 16.19%, la 5ta el 18.11% y la 9na 26.55% de la fundamental, teniendo un factor de distorsión armónica total de  $THD = 36.07\%$ .

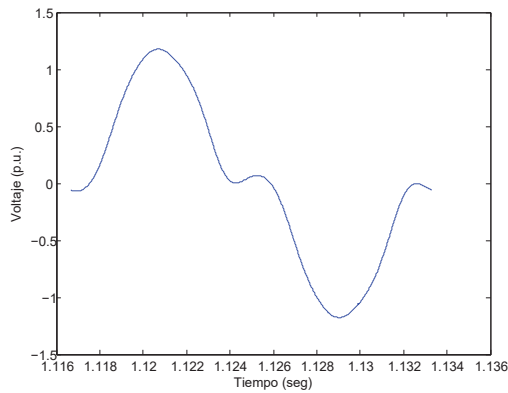


Figura 6.13: Forma de onda de voltaje en el nodo 14

Para el caso trifásico la red tiene la misma topología que para el caso monofásico. La representación en espacio de estado se realiza por medio de un conjunto de 123 EDO. En los transformadores de potencia que conforman a éste sistema se considera el efecto de saturación dado por la expresión (5.9).

En el caso trifásico se requieren 71 ciclos completos de integración para llegar al EEP tal como se muestra en la Tabla 6.5. Este resultado es el mismo para el método de FB utilizando procesamiento en paralelo.

Para este caso la solución en forma secuencial requiere de 17.802 segundos de procesamiento para llegar al EEP, lo que se puede apreciar en éste caso es que el tiempo para obtener el EEP es mayor comparado al caso monofásico, pero el rendimiento en términos del *speedup* es mejor. En la Figura 6.14 se muestra el *speedup* que se logra obtener al aplicar el procesamiento en paralelo utilizando de 1 a 8 núcleos de procesamiento y la Tabla 6.6 muestra los tiempos empleados en cada caso para llegar al EEP. Para éste caso el máximo *speedup* que se obtiene es de 4.604 utilizando 8 núcleos, esto quiere decir que la obtención del EEP se realiza 360.4% más rápido comparado a la forma secuencial.



Tabla 6.5: Proceso de convergencia. Caso de estudio IEEE-14 nodos trifásico

Ciclos completos	Error máximo
1	7.303121E+00
2	8.848844E+00
3	1.026904E+00
⋮	⋮
16	7.468659E-03
25	1.387483E-03
28	4.349737E-04
34	1.698212E-05
38	1.000026E-05
43	1.192060E-06
51	9.112667E-08
⋮	⋮
71	7.944278E-11

Tabla 6.6: Valores de *speedup* y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-14 trifásico

Número de núcleos	Tiempo de procesamiento (seg)	Speedup
1	17.802	1
2	9.831	1.810
3	7.066	2.519
4	5.820	3.067
5	4.980	3.574
6	4.432	4.023
7	4.071	4.372
8	3.866	4.604

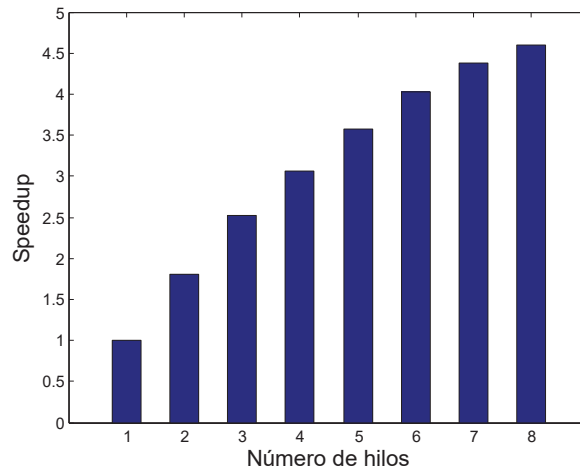


Figura 6.14: Gráfica del *speedup*. Caso de estudio IEEE-14 nodos trifásico

La Figura 6.15a muestra la forma de onda de la corriente en EEP de la fase A para la línea 20 que va del nodo 13 al nodo 14. Además la Figura 6.15b muestra su contenido armónico el cual es similar para el caso monofásico, donde se tiene que la 3ra armónica representa el 16.19%, la 5ta el 18.11% y la 9na 26.55% de la fundamental, teniendo un factor de distorsión armónica total de  $THD = 36.07\%$ , adicionalmente en la Figura 6.16 se muestra el perfil de voltaje en el nodo 14

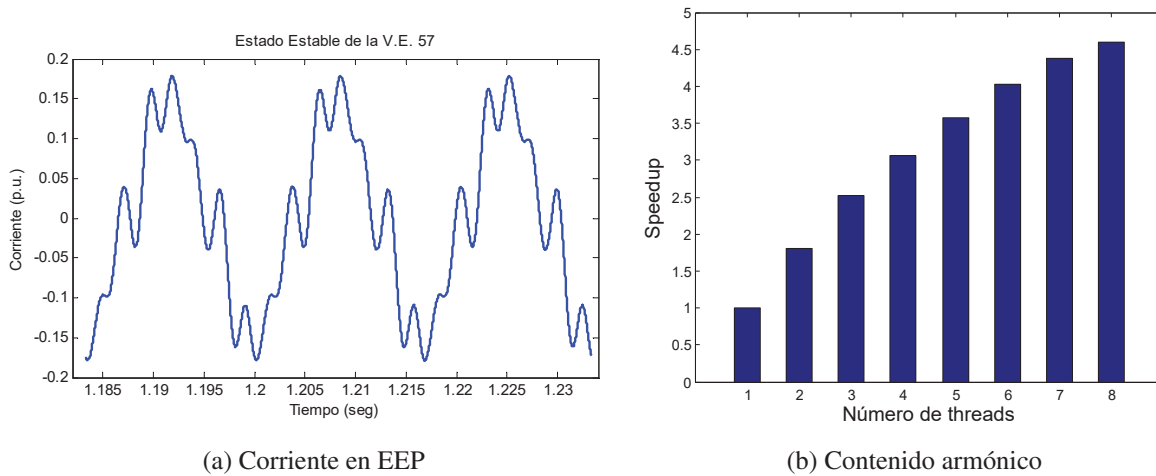


Figura 6.15: Forma de onda en EEP de la corriente en la línea 20 caso trifásico

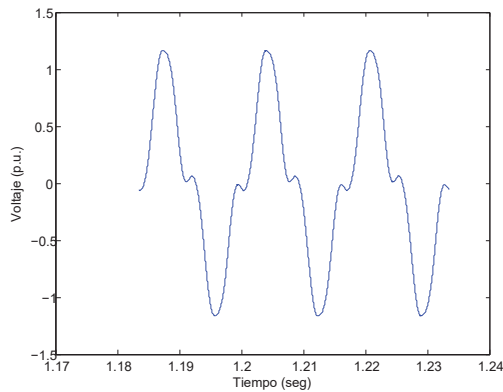


Figura 6.16: Forma de onda de voltaje en el nodo 14 caso trifásico

Como una prueba adicional, se añade un STATCOM en la configuración trifásica anterior, este se conecta en el nodo 14 con el fin de añadir distorsión armónica e incrementar el esfuerzo computacional para obtener el EEP. La topología resultante sigue siendo la misma que para los casos anteriores. La representación en espacio de estado de este sistema se realiza por medio de un conjunto de 126 EDO. En los transformadores de potencia que conforman a este sistema se considera el efecto de saturación dado por la expresión (5.9).

Teniendo activo el STATCOM en el sistema trifásico se requieren 155 ciclos completos de integración para llegar al EEP como se muestra en el proceso de convergencia de la Tabla 6.7, este resultado es el mismo para el método de FB utilizando procesamiento en paralelo.

En éste caso se aprecia un gran incremento en el tiempo de cómputo, esto se debe a la dinámica del STATCOM representada por un número adicional de ecuaciones algebraicas y su

respectivas variables estado, sin embargo la eficiencia del cálculo paralelo vuelve a incrementar su valor con respecto al caso trifásico. En éste caso se requiere de 39.567 segundos para llegar al EEP, . En la Figura 6.17 se muestra el *speedup* que se logra obtener al aplicar el procesamiento en paralelo utilizando de 1 a 8 núcleos, el cual puede seguir incrementándose al incorporar más elementos de proceso, en la Tabla 6.8 se puede ver la relación de tiempo y *speedup*. Para este caso el máximo *speedup* que se obtiene es de 4.665 utilizando 8 núcleos de proceso, esto quiere decir que la obtención del EEP se realiza 366.5% más rápido comparado a la forma secuencial.

Tabla 6.7: Proceso de convergencia. Caso de estudio IEEE-14 nodos trifásico con un STATCOM

<i>Ciclos completos</i>	<i>Error máximo</i>
1	7.011689E+00
2	8.942313E+00
3	1.862978E+00
⋮	⋮
16	1.738630E-02
25	3.558132E-03
28	2.370159E-03
34	1.056328E-03
38	6.159336E-04
71	7.191677E-06
99	1.648067E-07
⋮	⋮
155	8.880848E-11

Tabla 6.8: Valores de *speedup* y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-14 trifásico con un STATCOM

<i>Número de núcleos</i>	<i>Tiempo de procesamiento (seg)</i>	<i>Speedup</i>
1	39.567	1
2	22.025	1.796
3	15.888	2.490
4	12.990	3.045
5	11.351	3.485
6	9.990	3.960
7	9.044	4.374
8	8.481	4.665

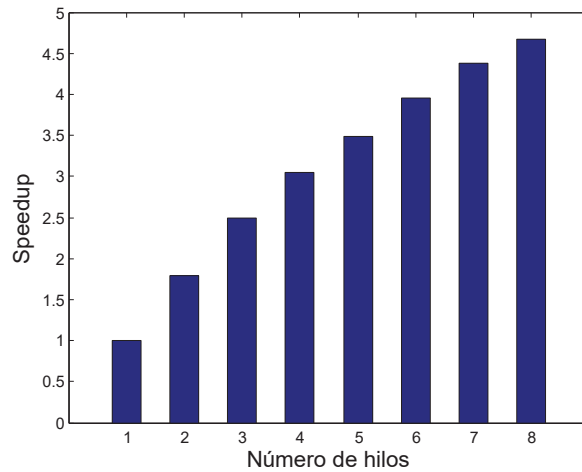
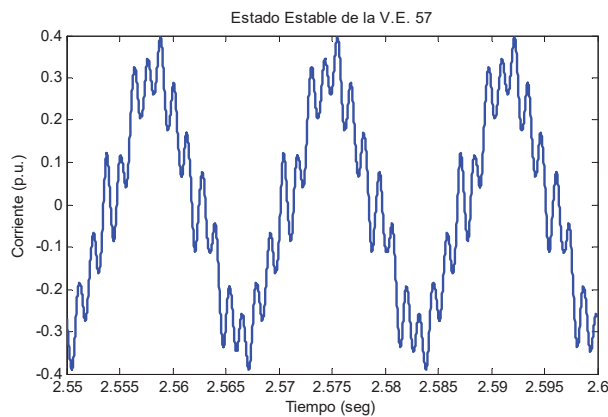
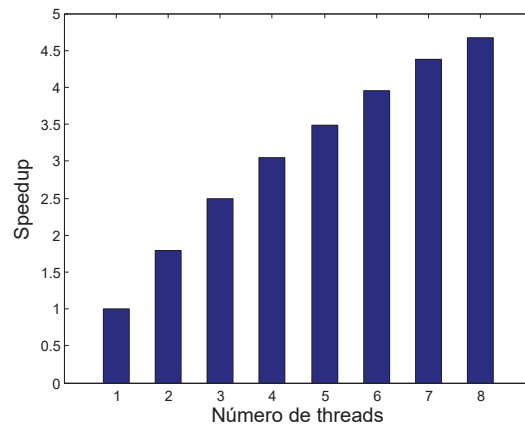


Figura 6.17: Gráfica del *speedup*. Caso de estudio IEEE-14 nodos trifásico con un STATCOM

Para éste caso en la Figura 6.7 se muestra la forma de onda del voltaje en el nodo 14 y la Figura 6.18b muestra el espectro armónico de la corriente en la línea 20 cuando se tiene activo el STATCOM, en el cual se puede observar que la 3ra armónica representa el 11.93 %, la 5ta el 6.41 %, 7ma 10.35 % y la 13va el 27.21 % de la fundamental, teniendo un factor de distorsión armónica total de  $THD = 33.0943\%$ .



(a) Corriente en EEP



(b) Contenido armónico

Figura 6.18: Forma de onda en EEP de la corriente en la línea 20 caso trifásico con STATCOM

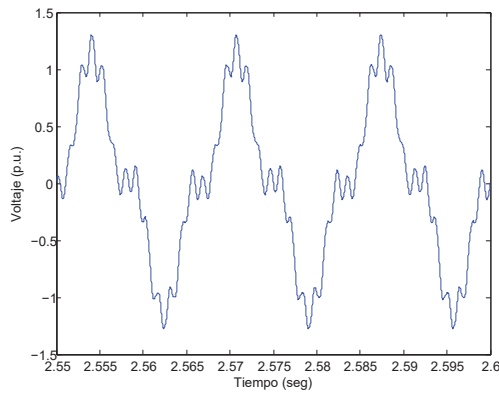


Figura 6.19: Forma de onda de voltaje en el nodo 14 con un STATCOM caso trifásico

### 6.5. Caso de estudio 3. Sistema IEEE-30 nodos modificado

Esta red de prueba tiene la topología del sistema IEEE-30 nodos [IEEE test case, 1993]. Sin embargo este ha sido modificado. El análisis se realizó para el caso monofásico y caso trifásico.

Para el caso monofásico la red contiene 41 líneas de transmisión, 30 bancos de capacitores, 1 horno de arco eléctrico y 6 generadores. Su representación en espacio de estado se modela por 79 EDO. El efecto de saturación en los transformadores de potencia está dado por la expresión (5.9). La Tabla 6.9 muestra la dinámica del error para los 173 periodos completos de integración necesarios para llegar al EEP en base al criterio de convergencia, el resultado es el mismo para el método de FB utilizando procesamiento en paralelo.

En ésta red de prueba se requiere de 18.785 segundos para llegar al EEP en forma secuencial. La Figura 6.20 muestra el *speedup* que se logra obtener al aplicar el procesamiento en paralelo utilizando de 1 a 8 núcleos y en la Tabla 6.10 se puede ver la reducción de tiempo al incorporar nuevos elementos de proceso, la cual es muy pobre, pero aun así se puede incrementar al incorporar un par más de elementos de proceso. En este caso el máximo *speedup* que se tiene es 3.093 con 8 núcleos, esto quiere decir que al utilizar un total de 8 núcleos se logra llegar al EEP 209.3 % más rápido comparado a la forma secuencial.

Tabla 6.9: Proceso de convergencia. Caso de estudio IEEE-30 nodos monofásico

Ciclos completos	Error máximo
1	1.259758E+00
2	4.803437E-01
3	1.796404E-01
⋮	⋮
16	1.421065E-05
25	1.483543E-06
38	4.256812E-07
44	2.777924E-07
58	1.135019E-07
63	8.334406E-08
71	5.099507E-08
⋮	⋮
173	9.946060E-11

Tabla 6.10: Valores de *speedup* y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-30 monofásico

Número de núcleos	Tiempo de procesamiento (seg)	Speedup
1	18.785	1
2	11.971	1.569
3	9.119	2.059
4	7.798	2.408
5	6.909	2.718
6	6.509	2.886
7	6.233	3.013
8	6.072	3.093

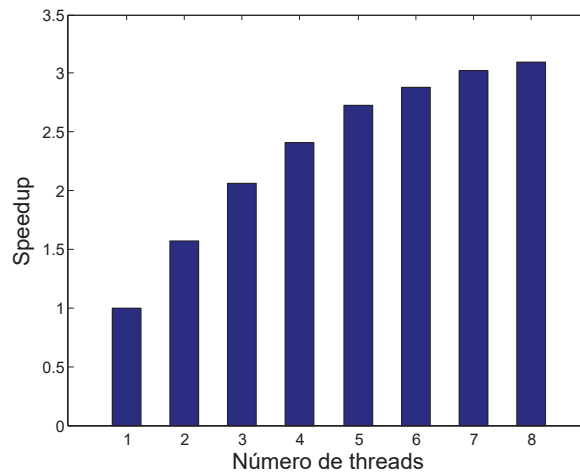


Figura 6.20: Gráfica del *speedup*. Caso de estudio IEEE-30 nodos monofásico

Para el caso trifásico la red presenta la misma topología que para el caso monofásico. La representación en espacio de estado de este sistema se realiza por medio de un conjunto de 235 EDO. En los transformadores de potencia que conforman a este sistema se considera el efecto de saturación dado por la expresión (5.9). Para este caso se requieren 174 ciclos completos de integración para llegar al EEP como se muestra en la Tabla 6.11, este resultado es el mismo para el método de FB utilizando procesamiento en paralelo.

## CAPÍTULO 6. CASOS DE ESTUDIO

Para el caso trifásico se requiere de 4 minutos y 47.356 segundos para llegar al EEP. En la Figura 6.21 se muestra el *speedup* obtenido al aplicar el procesamiento en paralelo utilizando de 1 a 8 núcleos, el cual se obtiene a partir de la Tabla 6.12. Para este caso el máximo *speedup* que se obtiene es de 6.454 con 8 núcleos, esto quiere decir que la obtención del EEP se realiza 545.4% más rápido.

Tabla 6.11: Proceso de convergencia. Caso de estudio IEEE-30 nodos trifásico

<i>Ciclos completos</i>	<i>Error máximo</i>
1	6.949678E+00
2	10.059447E+00
3	4.809664E+00
⋮	⋮
16	4.090678E-02
33	5.160559E-05
45	5.400856E-07
63	8.334411E-08
78	3.321576E-08
96	1.104285E-08
119	4.689985E-09
⋮	⋮
174	9.355949E-11

Tabla 6.12: Valores de *speedup* y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-30 trifásico

<i>Número de núcleos</i>	<i>Tiempo de procesamiento (seg)</i>	<i>Speedup</i>
1	287.356	1
2	147.547	1.947
3	101.444	2.832
4	79.154	3.630
5	64.371	4.464
6	54.903	5.233
7	49.144	5.847
8	44.523	6.454

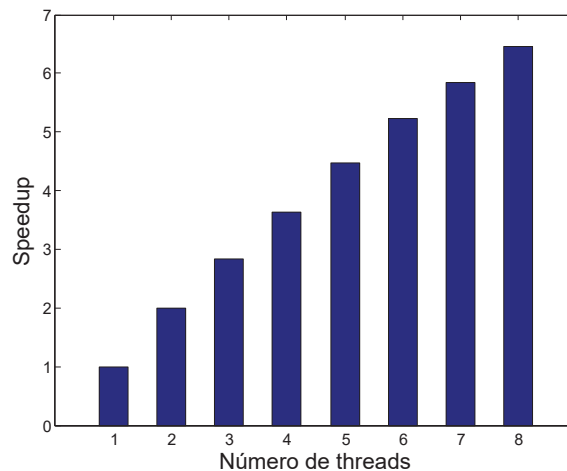


Figura 6.21: Gráfica del *speedup*. Caso de estudio IEEE-30 nodos trifásico

Para este caso de estudio la Figura 6.22a muestra la forma de onda de corriente obtenida en EEP de la línea 29 que va del nodo 10 a al nodo 22 donde está conectado el horno de arco

eléctrico y su perfil de voltaje se muestra en la Figura 6.23, así como su contenido armónico se muestra en la Figura 6.22b, donde se puede ver que la 3ra armónica representa el 8.68% de la fundamental, la 5ta representa el 3.87% y la 7ma el 0.95%, teniendo un factor de distorsión armónica total de  $THD = 9.6055\%$ .

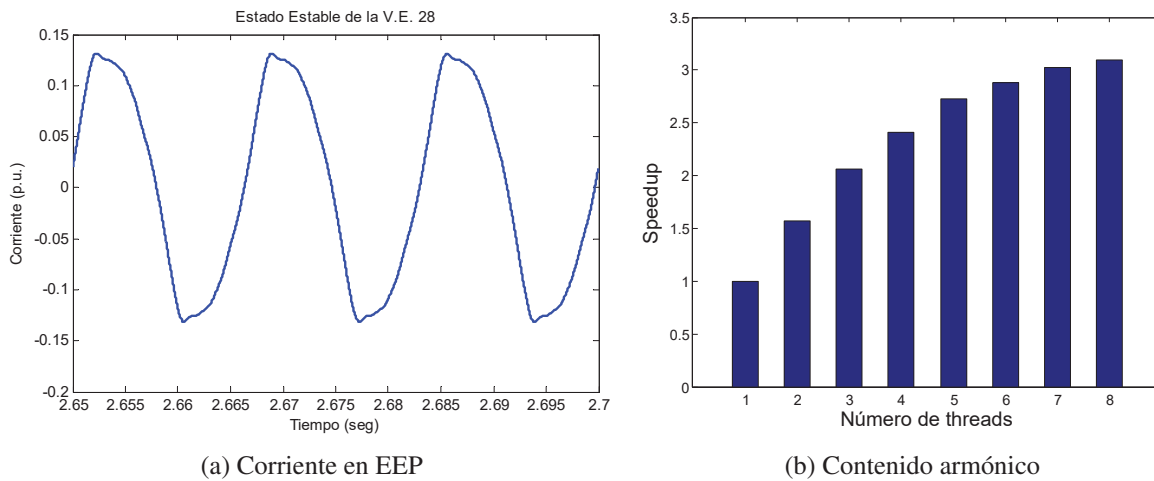


Figura 6.22: Forma de onda de corriente en EEP y contenido armónico en la línea 29

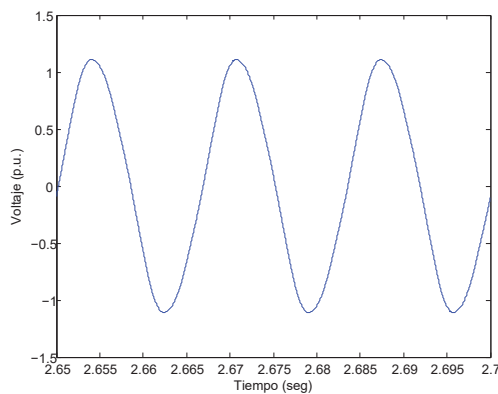


Figura 6.23: Forma de onda de voltaje en el nodo 22 caso trifásico

## 6.6. Caso de estudio 4. Sistema IEEE-57 nodos modificado

Esta red de prueba tiene la topología del sistema IEEE-57 nodos [IEEE test case, 1993], el cual ha sido modificado para su análisis. El análisis se realizó para el caso monofásico y caso trifásico.

Para el caso monofásico la red contiene 78 líneas de transmisión, 57 bancos de capacitores, 1 rama magnetizante y 7 generadores. Su representación en espacio de estado se modela



por medio de 143 EDO. El efecto de saturación en los transformadores de potencia está dado por la expresión (5.9). La Tabla 6.13 muestra el proceso de convergencia para los 115 periodos completos de integración necesarios para llegar al EEP, en base al criterio de convergencia propuesto en esta tesis, el resultado es el mismo para el método de FB utilizando procesamiento en paralelo.

Para el caso monofásico de esta red de prueba, el tiempo de ejecución secuencial necesario para alcanzar el EEP es de 37.726 segundos. La Figura 6.24 muestra el *speedup* que se logra obtener al aplicar el procesamiento en paralelo utilizando de 1 a 8 núcleos, el cual se obtiene a partir de la Tabla 6.14. Para este caso, el máximo *speedup* que se obtiene es de 4.915 con 8 núcleos, esto quiere decir que la obtención del EEP se realiza 391.5% más rápido.

Tabla 6.13: Proceso de convergencia. Caso de estudio IEEE-57 nodos monofásico

<i>Ciclos completos</i>	<i>Error máximo</i>
1	2.067590E+00
2	8.513420E-01
3	5.211737E-01
⋮	⋮
26	3.343098E-05
37	6.733111E-06
45	2.147200E-06
63	1.645673E-07
77	2.232365E-08
83	9.483002E-09
91	3.028208E-09
⋮	⋮
115	9.859330E-11

Tabla 6.14: Valores de *speedup* y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-57 monofásico

<i>Número de núcleos</i>	<i>Tiempo de procesamiento (seg)</i>	<i>Speedup</i>
1	37.726	1
2	21.453	1.758
3	15.362	2.455
4	12.109	3.115
5	10.275	3.671
6	8.978	4.202
7	8.263	4.565
8	7.675	4.915

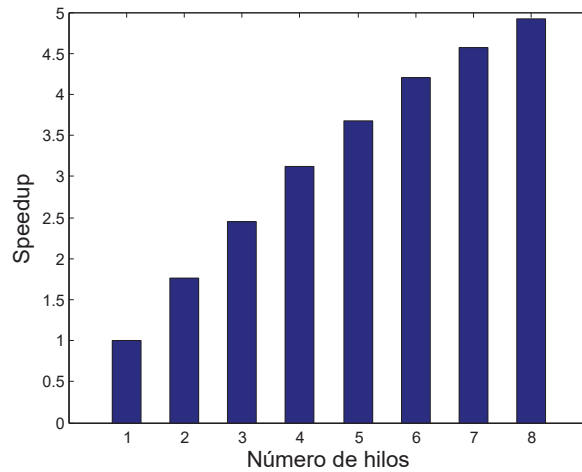


Figura 6.24: Gráfica del *speedup*. Caso de estudio IEEE-57 nodos monofásico

Para el caso trifásico la red presenta la misma topología que para el caso monofásico. La representación en espacio de estado de este sistema se realiza por medio de un conjunto de 429 EDO. En los transformadores de potencia que conforman a este sistema se considera el efecto de saturación dado por la expresión (5.9). Para este caso se requieren 115 ciclos completos de integración para llegar al EEP como se muestra en la Tabla 6.15, este resultado es el mismo para el método de FB utilizando procesamiento en paralelo.

Para el caso trifásico se requiere de 10 minutos y 35.035 segundos para llegar al EEP. En la Figura 6.25 se muestra el *speedup* obtenido al aplicar el procesamiento en paralelo utilizando de 1 a 8 núcleos, el cual se obtiene a partir de la Tabla 6.16 . Para este caso el máximo *speedup* que se obtiene es de 7.239 con 8 núcleos, esto quiere decir que la obtención del EEP se realiza 623.9% más rápido.

Tabla 6.15: Proceso de convergencia. Caso de estudio IEEE-57 nodos trifásico

Ciclos completos	Error máximo
1	7.040215E+00
2	10.638094E+00
3	3.909279E+00
⋮	⋮
16	1.365578E-02
23	2.476030E-03
35	1.287709E-05
43	2.856705E-06
58	3.358877E-07
66	1.072590E-07
79	1.678129E-08
⋮	⋮
115	9.859290E-11

Tabla 6.16: Valores de *speedup* y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-57 trifásico

Número de núcleos	Tiempo de procesamiento (seg)	Speedup
1	635.035	1
2	322.622	1.968
3	218.747	2.903
4	166.144	3.822
5	134.716	4.713
6	113.938	5.573
7	99.409	6.388
8	87.716	7.239

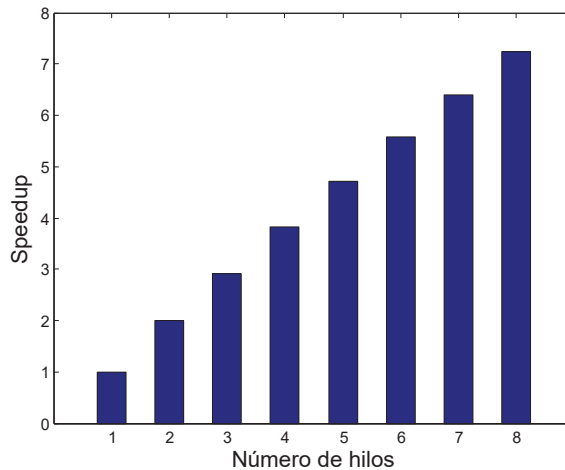


Figura 6.25: Gráfica del *speedup*. Caso de estudio IEEE-57 nodos trifásico

Para este caso de estudio la Figura 6.26a muestra la forma de onda de corriente obtenida en EEP de la línea 78 que va del nodo 56 a al nodo 57 donde está conectada una rama magnetizante y se presenta su perfil de voltaje en la Figura 6.27, también en la Figura 6.22b se muestra su contenido armónico, donde se puede ver que la 3ra armónica representa el 43.92% de la fundamental, la 5ta representa el 36.81% y la 7ma el 42.78%, teniendo un factor de distorsión armónica total de  $THD = 77.0683\%$ .

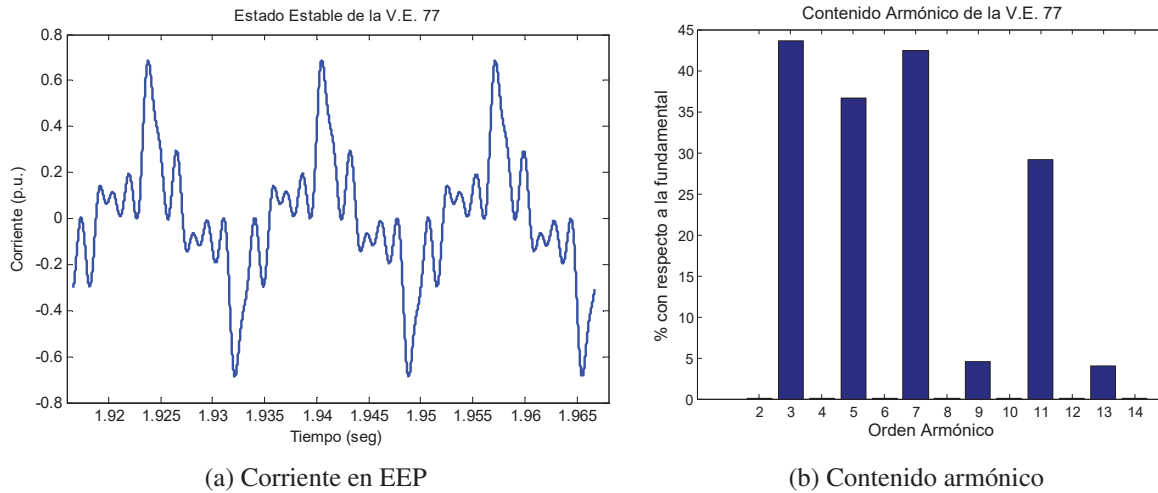


Figura 6.26: Forma de onda de corriente en EEP y contenido armónico en la línea 78

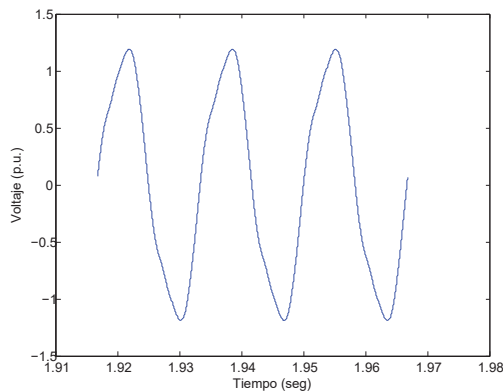


Figura 6.27: Forma de onda de voltaje en el nodo 57 caso trifásico

## 6.7. Caso de estudio 6. Sistema IEEE-118 nodos modificado

La red de prueba que se contempla para este caso de estudio tiene la topología del sistema IEEE-118 nodos [IEEE test case, 1993], el cual ha sido modificado. El análisis se realizó para el caso monofásico y caso trifásico.

Para el caso monofásico la red contiene 186 líneas de transmisión, 118 bancos de capacitores, 51 generadores y un horno de arco eléctrico en el nodo 29. Su representación en espacio de estado se modela por 357 EDO. El efecto de saturación en los transformadores de potencia está dado por la expresión (5.9). La Tabla 6.17 muestra el proceso de convergencia para obtener el EEP en un total de 137 periodos completos de integración al criterio de convergencia, el resultado es el mismo para el método de FB utilizando procesamiento en paralelo.

Para ésta red de prueba se requiere de 4 minutos y 19.793 segundos para llegar al EEP en forma secuencial. La Figura 6.18 muestra el *speedup* que se logra obtener al aplicar el procesamiento en paralelo utilizando de 1 a 8 núcleos, el cual se obtiene a partir de la Tabla 6.18. De la gráfica se aprecia que el máximo *speedup* que se tiene es 6.716 con 8 núcleos, esto quiere decir que al utilizar un total de 8 núcleos se logra llegar al EEP 571.6% más rápido en comparación a su forma secuencial.

Tabla 6.17: Proceso de convergencia. Caso de estudio IEEE-118 nodos monofásico

Ciclos completos	Error máximo
1	5.900755E-01
2	1.633806E-01
3	1.056212E-01
⋮	⋮
27	1.852909E-03
35	5.662482E-04
49	8.726159E-05
54	3.175735E-05
63	8.294659E-06
77	1.239807E-06
85	3.066467E-07
⋮	⋮
137	9.651080E-11

Tabla 6.18: Valores de *speedup* y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-118 monofásico

Número de núcleos	Tiempo de procesamiento (seg)	Speedup
1	259.793	1
2	136.542	1.902
3	92.611	2.805
4	70.980	3.660
5	57.928	4.484
6	49.522	5.246
7	42.612	6.096
8	38.680	6.716

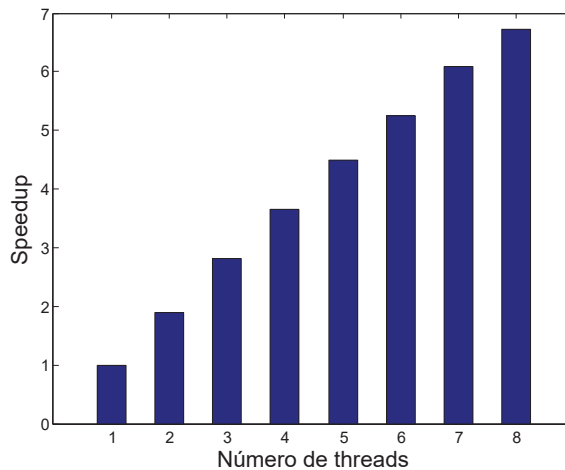


Figura 6.28: Gráfica del *speedup*. Caso de estudio IEEE-118 nodos monofásico

Para el caso trifásico la red presenta la misma topología que para el caso monofásico. La representación en espacio de estado de este sistema se realiza por medio de un conjunto de 1071 EDO. En los transformadores de potencia que conforman a este sistema se considera el efecto de saturación dado por la expresión (5.9). Para este caso se requieren 159 ciclos completos de integración para llegar al EEP como se muestra en la Tabla 6.19, este resultado es el mismo para el método de FB utilizando procesamiento en paralelo.

Para el caso trifásico se requiere de 45 minutos y 2.501 segundos para llegar al EEP. En la Figura 6.29 se muestra el *speedup* obtenido al aplicar el procesamiento en paralelo utilizando de 1 a 8 núcleos, el cual se obtiene a partir de la Tabla 6.20. Para este caso el máximo *speedup* que se obtiene es de 7.729 con 8 núcleos, esto quiere decir que la obtención del EEP se realiza 672.9% más rápido.

Tabla 6.19: Proceso de convergencia. Caso de estudio IEEE-118 nodos trifásico

<i>Ciclos completos</i>	<i>Error máximo</i>
1	6.874378E+00
2	9.732590E+00
3	7.990562E-01
⋮	⋮
16	2.442427E-01
27	2.962813E-02
38	6.860309E-03
50	1.220994E-04
65	1.615204E-05
80	2.136653E-06
96	2.469867E-07
⋮	⋮
159	9.819207E-11

Tabla 6.20: Valores de *speedup* y tiempo utilizando de 1 a 8 elementos de proceso. Caso de estudio IEEE-14 trifásico

<i>Número de núcleos</i>	<i>Tiempo de procesamiento (seg)</i>	<i>Speedup</i>
1	2702.501	1
2	1319.917	2.047
3	889.735	3.037
4	671.593	4.024
5	542.047	4.985
6	454.093	5.951
7	398.636	6.779
8	349.645	7.729

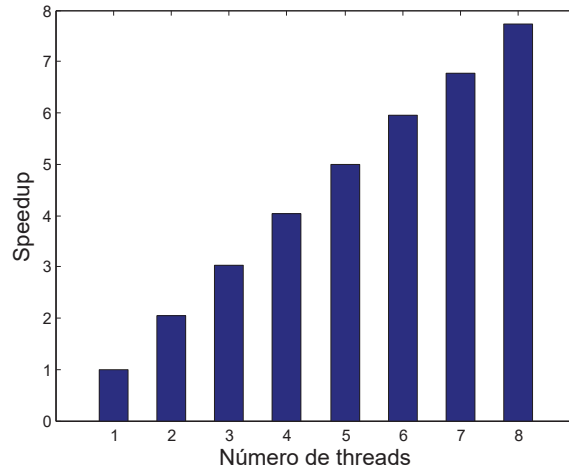


Figura 6.29: Gráfica del *speedup*. Caso de estudio IEEE-118 nodos trifásico

Para este caso de estudio la Figura 6.30a muestra la forma de onda de corriente obtenida en EEP de la línea 40 que va del nodo 29 a al nodo 31 donde está conectado el arco eléctrico, también en la Figura 6.30b se muestra su contenido armónico, donde se puede ver que la 3ra armónica representa el 11.84 % de la fundamental, la 5ta representa el 3.35 % y la 13va el 1.9 %, teniendo un factor de distorsión armónica total de  $THD = 12.593\%$ . Para éste caso de estudio el perfil de voltaje en comparación a los anteriores, véase la Figura 6.31.

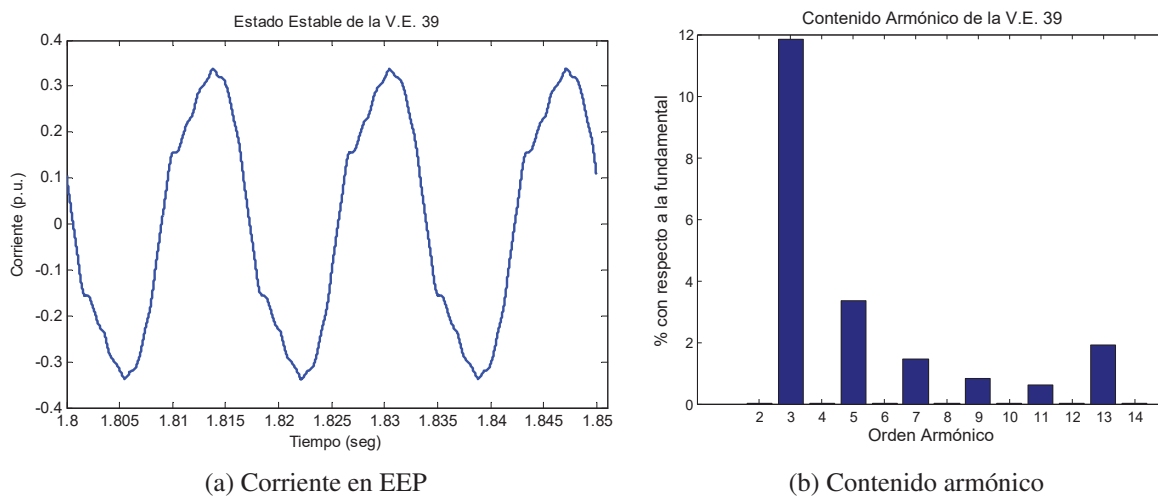


Figura 6.30: Forma de onda de corriente en EEP y contenido armónico en la línea 40

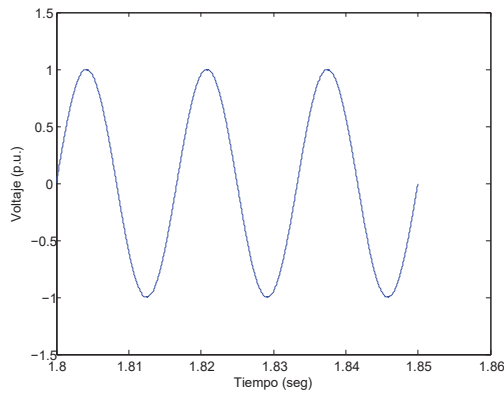


Figura 6.31: Forma de onda de voltaje en el nodo 31 para el caso trifásico

## 6.8. Conclusiones

Con los resultados obtenidos se ha mostrado que la aplicación de las técnicas de procesamiento en paralelo de grano fino en el análisis de los sistemas eléctricos de potencia es un elemento que permite disminuir de manera considerable los tiempos de cómputo requeridos para los diversos casos de estudio presentados en este capítulo. Se observó que la representación de los SEP en *espacio de estado* presenta una gran ventaja para el uso del procesamiento en paralelo. Además se demostró que mientras las dimensiones de la red se incrementan, los beneficios de la paralelización son mayores. Esto se respalda con los valores de *speedup* obtenido para cada uno de los casos trifásicos en los cuales las dimensiones son mayores a los casos monofásicos. De los seis casos de estudio que se presentan en esta tesis, los mejores resultados se obtienen en el caso de estudio 6 en el cual se obtiene un *speedup* de 7.7297 el cual representa una eficiencia de utilización de los elementos de proceso del 96.62%. Hay que señalar que para sistemas muy pequeños la paralelización es ineficiente como se observó en el primer caso de estudio en el cual se obtuvo un *speedup* de 0.5, esto quiere decir que se ejecuta en el doble de tiempo, lo cual se debe a que el tiempo necesario para la comunicación entre los elementos de proceso supera a los ahorros de tiempo generados por la implementación del procesamiento en paralelo. Por lo tanto los mejores resultados asociados a la aplicación del procesamiento en paralelo de grano fino se tienen en sistemas de grandes dimensiones, como es el caso de los SEP prácticos. Los resultados obtenidos de los casos de estudio presentados en este capítulo muestran que un análisis en los algoritmos numéricos permite encontrar áreas paralelizables de los mismos. Estas paralelizaciones permiten por un lado reducir el tiempo de cómputo mientras que por otro lado permiten hacer un uso más eficiente de los modernos equipos de cómputo que cuentan con más de un elemento de proceso.



## Capítulo 7

# Conclusiones generales y recomendaciones para trabajos futuros

En esta tesis se aplicaron técnicas de procesamiento en paralelo de grano fino para eficientar el método de integración de Runge-Kutta de cuarto orden, el cual fue utilizado para predecir el comportamiento transitorio y de estado estacionario periódico de redes eléctricas de potencia mediante el método convencional de FB.

En base a la propuesta e implementación de la aplicación de técnicas de procesamiento en paralelo de grano fino en el análisis de los SEP analizados en el capítulo 6 de esta tesis se llegan a las siguientes conclusiones generales.

### 7.1. Conclusiones generales

- En esta tesis se aplicaron de manera satisfactoria técnicas de procesamiento en paralelo de grano fino basadas en la plataforma OpenMP en el análisis en estado transitorio y en estado estacionario periódico, en el análisis de redes eléctricas de grandes dimensiones.
- En los diferentes casos de estudio se reporta el *speedup* obtenido al incorporar de 1 a 8 elementos de proceso. En estos casos de estudio se observó que la eficiencia obtenida se aumenta a medida que se incrementan las dimensiones del problema a resolver. Éste aumento en la eficiencia se ve claramente al hacer los estudios monofásico y trifásico de la misma red, donde el *speedup* es siempre mayor en los casos trifásicos, e incluso se logran eficiencias muy cercanas al 100%, como en el caso de prueba IEEE-118 nodos trifásico con 2 núcleos de procesamiento. Sin embargo, hay que señalar que para sistemas muy pequeños como es el caso del sistema de 3 nodos modelado por 9 EDOs, la aplicación

del procesamiento en paralelo es ineficiente ya que los tiempos de comunicación entre elementos de proceso que aunque son muy pequeño resultan mayores que los tiempos ahorrados utilizando más de un elemento de proceso.

- La implementación de la propuesta de procesamiento en paralelo realizada en esta tesis permitió obtener una máxima eficiencia de 96.621 % para el caso de la red prueba IEEE-118, obteniendo un speedup de 7.729 utilizando 8 núcleos.
- En esta tesis se tomó como base el código de C/C++ utilizado en [Ramos, 2007] para generar el conjunto de EDO de forma automática, el cual fue exitosamente modificado para obtener la configuración en espacio de estado del sistema e incorporar nuevas rutinas para poder hacer posible el análisis en nuevos dispositivos FACTS, como es el caso del STATCOM, el cual tiene un alto grado de complejidad y su incorporación a la red no siempre resulta trivial.
- Los resultados obtenidos de los casos de estudio presentados en esta tesis demuestran que un análisis detallado de los algoritmos numéricos usados en el análisis de los SEP permite encontrar áreas paralelizables de los mismos. Estas paralelizaciones logran dos objetivos: reducir el tiempo de cómputo y hacer un uso más eficiente de los modernos equipos de cómputo que cuentan con más de un elemento de proceso.
- Los algoritmos desarrollados de análisis pueden ser incorporados en simuladores de sistemas eléctricos lo cual permitirá poder analizar de manera eficiente diversas redes eléctricas con cualquier tipo de equipo conectado en las mismas así como diversas configuraciones.

## 7.2. Recomendaciones para trabajos futuros

Durante la realización de esta tesis, se observó que se puede extender la investigación a manera de trabajos futuros asociados con esta tesis, mediante la aplicación de los siguientes puntos.

- Incorporar modelos híbridos de procesamiento en paralelo, los cuales incluirían plataformas de procesamiento multicomputadoras que permitan incorporar métodos de aceleración y con esto reducir aun más el tiempo de análisis.
- Aplicar la metodología desarrollada en estudios dinámicos en tiempo real.
- Incorporar modelos trifásicos de los dispositivos FACTS , tales como el CSCT, RCT, etc.

## CAPÍTULO 7. CONCLUSIONES GENERALES Y RECOMENDACIONES PARA TRABAJOS FUTUROS

---

- Unificar el simulador de tal forma que el mismo simulador pueda hacer análisis monofásico y trifásico así como generar el espectro armónico contenido en EEP para los elementos de interés.
- Desarrollar modelos más detallados de generadores, transformadores, líneas de transmisión, etc.
- Desarrollar modelos de fuentes de energía renovables tales como aerogeneradores, sistemas fotovoltaicos, celdas de combustible, etc.

# Bibliografía

- [Acha *et al.*, 1990] E. Acha, A. Semlyen, y N. Rajakovic “A Harmonic Domain Computation Package for Nonlinear Problems and its Application to Electric Arcs,” IEEE Transactions on Power Delivery, Vol. 5, No. 3, Julio 1990, págs. 1390-1397.
- [Adachi *et al.*, 1989] T. Adachi, S. Kai, T. Iriyama, “ A Parallel Implementation of Nonlinear Steady State Analysis Based on Time-Domain Newton-Raphson Algorithm”, IEEE International Symposium on Circuits and Systems, Vol. 2, págs. 889-892, 1989.
- [Aprilie y Trick, 1972] T. J. Aprille, T. N. Trick “A Compute Algorithm to Determinate the Steady State Response of Nonlinear Oscilators”, IEEE Transactions on Circuit Theory, Vol. CT-19, No. 4, Julio 1972, págs. 354-360.
- [Ascher y Petzold, 1998] U. M. Ascher, L. R. Petzold, “Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations”, SIAM, 1ra edición, Philadelphia, USA, 1998.
- [Burden y Faires, 1998] R. L. Burden, J. D. Faires, “Análisis Numérico”, International Thomson Editores, 6ta edición, 1998.
- [Butcher, 1963] J. C. Butcher, “Coefficients for the study of Runge-Kutta processes”, Math. Soc., Vol. 3, pp. 202-206, Australia, 1963.
- [Calvo, 1998] D. M. Calvo, “Los Métodos de Runge-Kutta en la Noluición Numérica de Ecuaciones Diferenciales”, Discurso de Ingreso a la Academia de Ciencias Exactas, Física, Química y Naturales de Zaragoza Leído por el Académico, Zaragoza, 1998.
- [Cañizares, 2000] C. A. Cañizares, ”Power Flow and Transient Stabiity Models of FACTS Controllers for Voltage and Angle Stability Studies”, Power Engineering Society Winter Meeting. Singapore: IEEE, 2000. 1447-1454.

## BIBLIOGRAFÍA

---

- [Carrillo y Centeno, 2010] A. Carrillo, I. Ll. Centeno, "Cálculo Simbólico y Grafico con MAPLE", RA-MA Editorial, 2010.
- [Chandra *et al.*, 2001] R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald and R. Menon, "Parallel Programming in OpenMp" Morgan Kaufmann Publishers, Academic Press, San Francisco, USA, 2001.
- [Chen, 1984] C. Chi-Tsong, "Linear System Theory and Design", HRW, Series in Electrical and Computer Engineering, 1984.
- [Chua y Ushida, 1981] L. O. Chua, A. Ushida "Algorithms for Computing Almost Periodic Steady-State Response of Nonlinear Systems to Multiple Input Frequencies", IEEE Transactions on Circuits and Systems, Vol. CAS-28, No. 10, Octubre 1981, págs. 953-971.
- [Colon y Trick, 1973] F. R. Colon, T. N. Trick, "Fast Periodic Steady State Analysis for Large Signal Electronic Circuits", IEEE Journal of Solid- State Circuits, Vol. SC-8, No. 4, August 1973, pp. 260 – 269.
- [Dahlquist, 1959] G. Dahlquist, "Stability and error bounds in the numerical integration of ordinary differential equations", Trans. of the Royal Inst. of Techn., Stockholm, Nr. 130, 1959.
- [Dick y Watson, 1981] E. P. Dick, W. Watson, "Transformer Models for Transient Studies Based on Field Measurements", IEEE Transactions on Power Apparatus and Systems, Vol. 100, No. 1, Enero 1981, págs. 409-419.
- [Dommel, 1969] H. W. Dommel, "Digital Computer Solution of Electromagnetic Transients in Single and Multiphase Networks", IEEE Transactions on Power Apparatus and Systems, Vol. PAS-88, No. 4, Abril 1969, págs.388-399.
- [Dongarra, 2003] Edited by Jack Dongarra, "Sourcebook of Parallel Computing", Editorial Morgan Kaufmann Publishers, Estados Unidos 2003.
- [Dorf, 1993] R. C. Dorf, "The Electrical Engineering Handbook", IEEE PRESS, EUA, 1993.
- [Foster, 1994] I. Foster, "Designing and Building Parallel Programs", Addison Wesley, 1994.

## BIBLIOGRAFÍA

---

- [Garcia *et al.*, 2001] N. Garcia, E. Acha, A. Medina, “Swift Time Domain Solutions of Electric Systems Using Parallel Processing”, Proceedings of the Sixth International Conference IASTED, Rodas, Grecia, Julio 2001, págs. 172-177.
- [Garcia y Acha, 2004] N. Garcia, E. Acha, “Periodic Steady-State Analysis of Large Scale Electric Systems Using Poincare Map and Parallel Processing”, IEEE Transactions on Power Systems, Vol. 19, No, 4, November 2004, págs. 1784-1793.
- [Grigorieff, 1977] R. D. Grigorieff, “Numerik gewöhnlicher Differentialgleichungen”, Bd. 2: Mehrschrittverfahren, Stuttgart. B. G. Teubner. 1977.
- [Hak *et al.*, 1997] H. Hak-Guhn, J. Park, B. Lee, “Analysis of Thyristor Controlled Series Compensator Dynamics Using the State Variable Approach of a Periodic System Model”, IEEE Transactions on Power Delivery, Vol. 12, No. 4, Octubre 1997. págs. 1744-1750.
- [Halpin, 2003] S. M. Halpin, “Tutorial on Harmonics Modeling and Simulation” Symposium Quality and Security of Electric Power Delivery Systems, Montreal, Canada, 2003.
- [Hayt y Kemmerly, 1993] W. H. Hayt Jr., J. E. Kemmerly, “Análisis de Circuitos en Ingeniería” McGraw-Hill, Quinta Edición, 1993.
- [Henrici, 1962] P. Henrici, “Discrete variable method in Ordinary differential equations”, John Wiley & Sons, Inc., New York, 1962.
- [Hingorani y Gyugyi, 2000] N. H. Hingorani, L. Gyugyi, “Understanding FACTS Concepts and Technology of Flexible AC Transmission Systems”, IEEE Press. E.U.A., 2000.
- [Hornbeck, 1975] R. Hornbeck, “Numerical Methods”. Quantum Publishers, Inc. 1975. EUA.
- [IEEE test case, 1993] R. Christie, “Power System Test Cases Archive”, University of Washington, College of Engineering, Electrical Engineering. 1993. Disponible en: <http://www.ee.washington.edu/pstca/>
- [Kundert *et al.*, 1990] K. S. Kundert, J. K. While, A. Sangiovanni-Vincentelli, “Steady-State Methods for Simulating Analog and Microwave Circuit”, Boston, Kluwer Academic Publishers, 1990.

- [Kron, 1963] G. Kron, "Diakoptics", Macdonald. Londres, 1963.
- [Lewis y Berg, 1998] B. Lewis, D. J. Berg, "Multithreaded Programming with Pthreads", Prentice Hall, 1998.
- [Liu y Meyer, 1987] T. H. Liu, W. S. Meyer, "Electromagnetic Transients Program Rule Book", Bonneville Power Administration, Portland, Oregon, June 1987.
- [Mariños *et al.*, 1994] Z. A. Mariños, J. L. R. Pereira, Jr. Carneiro, "Fast Harmonic Power Flow Calculation Using Parallel Processing", IEE Proc.-Gener. Transm. Distrib., Vol 141, No. 1. Enero 1994, págs. 27-32.
- [MathWorks, 2009] The MathWorks Inc. MATLAB 7 Mathematics, 2009.
- [Medina *et al.*, 2003] A. Medina, A. Ramos-Paz, C. R. Fuerte-Esquivel, "Swift Computation of the Periodic Steady State Solution of Power Systems Containing TCSCs", Electrical Power and Energy Systems. No. 25, 2003, págs. 689-694.
- [Medina *et al.*, 2003b] A. Medina, A. Ramos-Paz, C. R. Fuerte-Esquivel, "Periodic Steady State Solution of Electric Systems With Nonlinear Components Using Parallel Processing", IEEE Transaction Power Systems, Vol. 18, No. 2, págs. 963-965, Mayo 2003.
- [Medina y Ramos, 2005] A. Medina, A. Ramos-Paz, "PVM and MT Parallel Processing Platforms Applied to the Computation of Driving Point Impedances in Power Systems", WSEAS Transactions on Circuits and Systems. No. 11, Vol. 4, Noviembre 2005. 1702-1709.
- [Medina *et al.*, 2006] A. Medina, A. Ramos-Paz, C. R. Fuerte-Esquivel, "Efficient Computation of the Period Steady State Solution of Nonlinear Electric Systems Applying Parallel Processing Techniques", The International Journal for Computation and Mathematics in Electrical & Electronic Engineering, Vol. 25, No. 4, 2006.
- [Miller, 1982] T. J. E. Miller, "Reactive Power Control in Electric Systems", John Wiley and Sons. E.U.A. 1982.
- [Miller y Boxer, 2005] Russ Miller y Laurence Boxer, "Algorithms Sequential & Parallel: A unified Approach", Segunda Edición, Editorial Thomson, Estados Unidos 2005.

## BIBLIOGRAFÍA

---

- [Nakamura, 1992] S. Nakamura, “Métodos Numéricos Aplicados con Software”, Pearson Edición, 1ra edición, 1992.
- [Nabavi y Iravani, 1996] A. Nabavi-Niaki, M. R. Iravani, “Steady-State and Dynamic Models of Unified Power Flow Controller for Power System Studies”, IEEE Transaction Power Systems, Vol. 11, No. 4, págs. 1937-1943, November 1996.
- [Napoles y Negron, 1994] J. E. Napoles y C. Negron, “De la mecánica analítica a las Ecuaciones Diferenciales Ordinarias. Algunos apuntes históricos”, Revista Lull, Vol. 17, No. 32, págs. 190-206, 1994.
- [Nichols *et al.*, 1996] B. Nichols, D. Buttler, J. P. Farrell, “Pthreads Programming”, O’Reilly Media, 1ra edición, USA. 1996.
- [Ogata, 1993] K. Ogata, “Ingeniería de Control Moderna”, Prentice Hall, 1993.
- [openmp.org] [www.openmp.org](http://www.openmp.org), ”OpenMp Application Program Interface: version 2.5 May 2005 ” OpenMp Architecture Review Board 1997-2005.
- [Parker y Chua, 1989] T. S. Parker y L. O. Chua, “Practical Numerical Algorithms for Chaotic Systems”, Springer-Verlag, 1989.
- [Press *et al.*, 2007] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes The Art of Scientific Computing”, 3ra edición, Cambridge University Press, 2007.
- [Quinn, 2004] M. J. Quinn, “Parallel Programming in C with MPI and OpenMP”, McGraw-Hill, 1ra edición, New York, USA, 2004.
- [Ramos, 2007] A. Ramos-Paz, “Técnica para la Generación Automática de Ecuaciones Diferenciales No Autónomas para Representar el Comportamiento Dinámico de Sistemas Eléctricos No- Lineales Incorporando Herramientas Avanzadas de Cómputo”, Tesis Doctoral, Facultad de Ingeniería Eléctrica, U.M.S.N.H. 2007.
- [Rauber y Runger, 2010] T. Rauber, G. Runger, “Parallel Programing For Multicore and Cluster Systems”, Springer, New York, USA, 2010.
- [Raymod y DeCarlo, 1989] A. Raymod, DeCarlo, “A State Variable Approach with Numerical Implementation”, Prentice Hall, Englewood Cliffs, 1989.



## BIBLIOGRAFÍA

---

- [Runge, 1895] C. Runge, “Ueber die numerische Auflosung von Differentialgleichungen”, *Math. Ann.*, Vol 46, pp. 167-178, 1895.
- [Shampine *et al.*, 1997] L. F. Shampine, R. C. Jr. Allen, S. Pruess, “Fundamentals of numerical computing”, 1ra edición, John Wiley & Sons, 1997.
- [Sato, 2002] M. Sato, “OpenMP: parallel programing API shared memory multiprocessors and on-chip multiprocessors”, University of Tsukuba, Tsukuba Japan, 2002.
- [Segundo y Medina, 2009] J. Segundo-Ramírez, A. Medina, “Modeling of FACTS Devices Based on SPWM VSCs”, *IEEE Transactions on power delivery*, Vol. 24, No. 4, pp. 1815-1823, October 2009.
- [Semlyen y Medina, 1995] A. Semlyen, A. Medina, “Computation of the periodic steady state in systems with nonlinear components using a hybrid time and frequency domain methodology”, *IEEE Trans. Power Syst.*, Vol. 10, No. 3, pp. 1498–1504, Aug. 1995.
- [Stavrakakis *et al.*, 1990] G. S. Stavrakakis, C. Lefas, A. Pouliezos, “Parallel Processing Computer Implementation of a Real Time DC Motor Drive Fault Detection Algorithm”, *IEE Proceedings*, Vol. 137, No. 5 Septiembre 1990. págs. 309-313.
- [Tavakoli y Hamill, 2005] M. Tavakoli-Bina, D. C. Hamill. “Average Circuit Model for Angle Controlled STATCOM”, *IEE Proceiding Electric Power Application* 152, pp. 653-659, No. 3, May 2005 .
- [Usaola, 1990] J. Usaola-García, “Régimen Permanente de Sistemas Eléctricos de Potencia con Elementos No lineales Mediante un Procedimiento Híbrido de Análisis en los Dominios del Tiempo y de la Frecuencia”, Tesis de Doctorado, Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros Industriales, 1990.
- [Uzunovic *et al.*, 1997] E. Uzunovic, C. A. Cañizares, J. Reeve, “Fundamental Frequency Model of Static Synchronous Compensator”, *NAPS. Laramie*, 1997. 49-54.
- [Uzunovic *et al.*, 1998] E. Uzunovic, C. A. Cañizares, J. Reeve. “Fundamental Frequency Model of Unified Power Flow Controller”, *NAPS. Cleveland*, 1998. 294-299.

## BIBLIOGRAFÍA

---

- [Watson y Arrillaga, 2003] N. Watson, J. Arrillaga, “Power systems electromagnetic transients simulation”, IEE POWER AND ENERGY SERIES 39, The Institution of Electrical Engineers (IEE), London, United Kingdom, 2003.
- [Werler y Glavitsch, 1993] K. Werler, H. Glavitsch, “Computation of Transients by Parallel Processing”, IEEE Transactions on Power Delivery, Vol. 8, No. 3, Julio 1993.
- [Wolfram, 2011] Wolfram, “Wolfram Mathematica 8 Documentation Center”, Disponible en: <http://reference.wolfram.com/mathematica/guide/Mathematica.html>.
- [Zill y Cullen, 2005] D. G. Zill, M. R. Cullen, “Differential Equations with Boundary-Value Problems”, Brooks Cole, 6ta edición, 2005.
- [Zwillinger, 1997] D. Zwillinger, “Handbook of Differential Equations”, Academic Press, 3ra edición, 1997.

# Apéndice A

## Datos de los Sistemas de Prueba

A continuación se presentan los datos utilizados para los seis casos de estudio.

### A.1. Sistema de 3 nodos

El sistema de prueba de 3 nodos está compuesto de 9 elementos.

Tabla A.1: Número de nodos y componentes del sistema eléctrico de 3 nodos.

<i>Nodos</i>	<i>Líneas de transmisión</i>	<i>Bancos de capacitores</i>	<i>Generadores</i>	<i>Ramas magnetizantes</i>
3	3	3	1	2

Tabla A.2: Datos de las líneas de transmisión del sistema eléctrico de 3 nodos.

<i>Número de elemento</i>	<i>Nodo de envío</i>	<i>Nodo de recepción</i>	<i>R (p.u.)</i>	<i>L (p.u.)</i>
L_1	1	2	0.01	0.1
L_2	1	3	0.01	0.1
L_3	2	3	0.01	0.1

Tabla A.3: Datos de los bancos de capacitores del sistema eléctrico de 3 nodos.

<i>Número de elemento</i>	<i>Nodo de conexión</i>	<i>Nodo de referencia</i>	<i>C (p.u.)</i>
C_1	1	0	0.1
C_2	2	0	0.1
C_3	3	0	0.1

APÉNDICE A. DATOS DE LOS SISTEMAS DE PRUEBA

---

Tabla A.4: Datos de los generadores del sistema eléctrico de 3 nodos.

<i>Número de elemento</i>	<i>Nodo de conexión</i>	<i>Nodo de referencia</i>	<i>Amplitud máxima</i>	<i>Ángulo de fase</i>
G_1	1	0	1.0	0.0

Tabla A.5: Datos de las ramas magnetizantes del sistema eléctrico de 3 nodos.

<i>Número de elemento</i>	<i>Nodo de conexión</i>	<i>Nodo de referencia</i>	<i>R (p.u.)</i>	<i>Exponente</i>
RM_1	2	0	0.1	0.1
RM_2	3	0	0.1	0.1

## A.2. Sistema IEEE-14 nodos

Tabla A.6: Número de nodos y componentes del sistema eléctrico IEEE-14 nodos

<i>Nodos</i>	<i>Líneas de transmisión</i>	<i>Bancos de capacitores</i>	<i>Generadores</i>	<i>Ramas magnetizantes</i>	<i>STATCOM</i>
14	20	14	2	5	1

Tabla A.7: Datos de las líneas de transmisión del sistema eléctrico IEEE-14 nodos

<i>Número de elemento</i>	<i>Nodo de envío</i>	<i>Nodo de recepción</i>	<i>R (p.u.)</i>	<i>L (p.u.)</i>	<i>Número de elemento</i>	<i>Nodo de envío</i>	<i>Nodo de recepción</i>	<i>R (p.u.)</i>	<i>L (p.u.)</i>
L_1	1	2	0.01	0.1	L_11	6	11	0.01	0.1
L_2	1	5	0.01	0.1	L_12	6	12	0.01	0.1
L_3	2	3	0.01	0.1	L_13	6	13	0.01	0.1
L_4	2	4	0.01	0.1	L_14	7	8	0.01	0.1
L_5	2	5	0.01	0.1	L_15	7	9	0.01	0.1
L_6	3	4	0.01	0.1	L_16	9	10	0.01	0.1
L_7	5	4	0.01	0.1	L_17	9	14	0.01	0.1
L_8	4	7	0.01	0.1	L_18	10	11	0.01	0.1
L_9	4	9	0.01	0.1	L_19	12	13	0.01	0.1
L_10	5	6	0.01	0.1	L_20	13	14	0.01	0.1

APÉNDICE A. DATOS DE LOS SISTEMAS DE PRUEBA

Tabla A.8: Datos de los bancos de capacitores del sistema eléctrico IEEE-14 nodos

Número de elemento	Nodo de conexión	Nodo de referencia	C (p.u.)	Número de elemento	Nodo de conexión	Nodo de referencia	C (p.u.)
C_1	1	0	0.1	C_8	8	0	0.1
C_2	2	0	0.1	C_9	9	0	0.1
C_3	3	0	0.1	C_10	10	0	0.1
C_4	4	0	0.1	C_11	11	0	0.1
C_5	5	0	0.1	C_12	12	0	0.1
C_6	6	0	0.1	C_13	13	0	0.1
C_7	7	0	0.1	C_14	14	0	0.1

Tabla A.9: Datos de los generadores del sistema eléctrico IEEE-14 nodos

Número de elemento	Nodo de conexión	Nodo de referencia	Amplitud máxima	Ángulo de fase
G_1	1	0	1.0	0.0
G_2	2	0	1.0	0.0

Tabla A.10: Datos de las ramas magnetizantes del sistema eléctrico IEEE-14 nodos

Número de elemento	Nodo de conexión	Nodo de referencia	R (p.u.)	L (p.u.)	Exponente
RM_1	4	0	0.1	0.1	5.0
RM_2	5	0	0.1	0.1	5.0
RM_3	6	0	0.1	0.1	5.0
RM_4	9	0	0.1	0.1	5.0
RM_5	14	0	0.1	0.1	5.0

Tabla A.11: Datos de los STATCOM's del sistema eléctrico IEEE-14 nodos

Número de elemento	Nodo de conexión	Nodo de referencia	R (p.u.)	L (p.u.)	rs (p.u.)	Vcd (p.u.)	me	mf
STM_1	14	0	0.05	1.885	0.001	5	0.7	15

### A.3. Sistema IEEE-30 nodos

Tabla A.12: Número de nodos y componentes del sistema eléctrico IEEE-30 nodos

<i>Nodos</i>	<i>Líneas de transmisión</i>	<i>Bancos de capacitores</i>	<i>Generadores</i>	<i>Hornos de arco</i>
30	41	30	6	1

Tabla A.13: Datos de las líneas de transmisión del sistema eléctrico IEEE-30 nodos

<i>Número de elemento</i>	<i>Nodo de envío</i>	<i>Nodo de recepción</i>	<i>R (p.u.)</i>	<i>L (p.u.)</i>	<i>Número de elemento</i>	<i>Nodo de envío</i>	<i>Nodo de recepción</i>	<i>R (p.u.)</i>	<i>L (p.u.)</i>
L_1	1	2	0.0192	0.0575	L_22	16	17	0.0824	0.1923
L_2	1	3	0.0452	0.1852	L_23	15	18	0.1073	0.2185
L_3	2	4	0.0570	0.1737	L_24	18	19	0.0639	0.1292
L_4	3	4	0.0132	0.0379	L_25	19	20	0.0340	0.0680
L_5	2	5	0.0472	0.1983	L_26	10	20	0.0936	0.2090
L_6	2	6	0.0581	0.1763	L_27	10	17	0.0324	0.0845
L_7	4	6	0.0119	0.0414	L_28	10	21	0.0348	0.0749
L_8	4	12	0.00001	0.2560	L_29	10	22	0.0727	0.1499
L_9	5	7	0.0460	0.1160	L_30	12	13	0.00001	0.1400
L_10	6	7	0.0267	0.0820	L_31	21	22	0.0116	0.0236
L_11	6	8	0.0120	0.0420	L_32	15	23	0.1000	0.2020
L_12	6	9	0.00001	0.2080	L_33	22	24	0.1150	0.1790
L_13	6	10	0.00001	0.5560	L_34	23	24	0.1320	0.2700
L_14	6	28	0.0169	0.0599	L_35	24	25	0.1885	0.3292
L_15	8	28	0.0636	0.2000	L_36	25	26	0.2544	0.3800
L_16	9	10	0.00001	0.1100	L_37	25	27	0.1093	0.2087
L_17	9	11	0.00001	0.2080	L_38	27	29	0.2198	0.4153
L_18	12	14	0.1231	0.2559	L_39	27	30	0.3202	0.6027
L_19	12	15	0.0662	0.1304	L_40	28	27	0.00001	0.3960
L_20	12	16	0.0945	0.1987	L_41	29	30	0.2399	0.4533
L_21	14	15	0.2210	0.1997					

Tabla A.14: Datos de los bancos de capacitores del sistema eléctrico IEEE-30 nodos

<i>Número de elemento</i>	<i>Nodo de conexión</i>	<i>Nodo de referencia</i>	<i>C (p.u.)</i>	<i>Número de elemento</i>	<i>Nodo de conexión</i>	<i>Nodo de referencia</i>	<i>C (p.u.)</i>
C_1	1	0	0.1	C_16	16	0	0.1
C_2	2	0	0.1	C_17	17	0	0.1
C_3	3	0	0.1	C_18	18	0	0.1
C_4	4	0	0.1	C_19	19	0	0.1

APÉNDICE A. DATOS DE LOS SISTEMAS DE PRUEBA

C_5	5	0	0.1	C_20	20	0	0.1
C_6	6	0	0.1	C_21	21	0	0.1
C_7	7	0	0.1	C_22	22	0	0.1
C_8	8	0	0.1	C_23	23	0	0.1
C_9	9	0	0.1	C_24	24	0	0.1
C_10	10	0	0.1	C_25	25	0	0.1
C_11	11	0	0.1	C_26	26	0	0.1
C_12	12	0	0.1	C_27	27	0	0.1
C_13	13	0	0.1	C_28	28	0	0.1
C_14	14	0	0.1	C_29	29	0	0.1
C_15	15	0	0.1	C_30	30	0	0.1

Tabla A.15: Datos de los generadores del sistema eléctrico IEEE-30 nodos

<i>Número de elemento</i>	<i>Nodo de conexión</i>	<i>Nodo de referencia</i>	<i>Amplitud máxima</i>	<i>Ángulo de fase</i>
G_1	1	0	1.0	0.0
G_2	2	0	1.0	0.0
G_3	5	0	1.0	0.0
G_4	8	0	1.0	0.0
G_5	11	0	1.0	0.0
G_6	13	0	1.0	0.0

Tabla A.16: Datos de los hornos de arco eléctrico del sistema IEEE-30 nodos

<i>Número de elemento</i>	<i>Nodo de conexión</i>	<i>Nodo de referencia</i>	L (p.u.)	k1	k2	k3	n	m	<i>Condición inicial</i>
H_1	22	0	0.1	0.008	0.005	3.0	2.0	2.0	0.1

## A.4. Sistema IEEE-57 nodos

Tabla A.17: Número de nodos y componentes del sistema eléctrico IEEE-57 nodos

<i>Nodos</i>	<i>Líneas de transmisión</i>	<i>Bancos de capacitores</i>	<i>Generadores</i>	<i>Ramas magnetizantes</i>
57	78	57	7	1

Tabla A.18: Datos de las líneas de transmisión del sistema eléctrico IEEE-57 nodos

<i>Número de elemento</i>	<i>Nodo de envío</i>	<i>Nodo de recepción</i>	<i>R (p.u.)</i>	<i>L (p.u.)</i>	<i>Número de elemento</i>	<i>Nodo de envío</i>	<i>Nodo de recepción</i>	<i>R (p.u.)</i>	<i>L (p.u.)</i>
L_1	1	2	0.0083	0.028	L_40	22	23	0.0099	0.0152
L_2	1	15	0.0178	0.091	L_41	22	38	0.0192	0.0295
L_3	1	16	0.0454	0.206	L_42	23	24	0.166	0.256
L_4	1	17	0.0238	0.108	L_43	24	25	0.01	1.182
L_5	2	3	0.0298	0.085	L_44	24	26	0.01	0.0473
L_6	3	4	0.0112	0.0366	L_45	25	30	0.135	0.202
L_7	3	15	0.0162	0.053	L_46	26	27	0.165	0.254
L_8	4	5	0.0625	0.132	L_47	27	28	0.0618	0.0954
L_9	4	6	0.043	0.148	L_48	28	29	0.0418	0.0587
L_10	4	18	0.01	0.555	L_49	29	52	0.1442	0.187
L_11	5	6	0.0302	0.0641	L_50	30	31	0.326	0.497
L_12	6	7	0.02	0.102	L_51	31	32	0.507	0.755
L_13	6	8	0.0339	0.173	L_52	32	33	0.0392	0.036
L_14	7	8	0.0139	0.0712	L_53	32	34	0.01	0.953
L_15	7	29	0.01	0.0648	L_54	34	35	0.052	0.078
L_16	8	9	0.0099	0.0505	L_55	35	36	0.043	0.0537
L_17	9	10	0.0369	0.1679	L_56	36	37	0.029	0.0366
L_18	9	11	0.0258	0.0848	L_57	36	40	0.03	0.0466
L_19	9	12	0.0648	0.295	L_58	37	38	0.0651	0.1009
L_20	9	13	0.0481	0.158	L_59	37	39	0.0239	0.0379
L_21	9	55	0.01	0.1205	L_60	38	44	0.0289	0.0585
L_22	10	12	0.0277	0.1262	L_61	38	48	0.0312	0.0482
L_23	10	51	0.01	0.0712	L_62	38	49	0.115	0.177
L_24	11	13	0.0223	0.0732	L_63	39	57	0.01	1.355
L_25	11	41	0.01	0.749	L_64	40	56	0.01	1.195
L_26	11	43	0.01	0.153	L_65	41	42	0.207	0.352
L_27	12	13	0.0178	0.058	L_66	41	43	0.01	0.412
L_28	12	16	0.018	0.0813	L_67	41	56	0.553	0.549
L_29	12	17	0.0397	0.179	L_68	42	56	0.2125	0.354
L_30	13	14	0.0132	0.0434	L_69	44	45	0.0624	0.1242



APÉNDICE A. DATOS DE LOS SISTEMAS DE PRUEBA

L_31	13	15	0.0269	0.0869	L_70	46	47	0.023	0.068
L_32	13	49	0.01	0.191	L_71	47	48	0.0182	0.0233
L_33	14	15	0.0171	0.0547	L_72	48	49	0.0834	0.129
L_34	14	46	0.01	0.0735	L_73	49	50	0.0801	0.128
L_35	15	45	0.01	0.1042	L_74	50	51	0.1386	0.22
L_36	18	19	0.461	0.685	L_75	52	53	0.0762	0.0984
L_37	19	20	0.283	0.434	L_76	53	54	0.1878	0.232
L_38	20	21	0.01	0.7767	L_77	54	55	0.1732	0.2265
L_39	21	22	0.0736	0.117	L_78	56	57	0.174	0.26

Tabla A.19: Datos de los bancos de capacitores del sistema eléctrico IEEE-57 nodos

Número de elemento	Nodo de conexión	Nodo de referencia	C (p.u.)	Número de elemento	Nodo de conexión	Nodo de referencia	C (p.u.)
C_1	1	0	0.1	C_30	30	0	0.1
C_2	2	0	0.1	C_31	31	0	0.1
C_3	3	0	0.1	C_32	32	0	0.1
C_4	4	0	0.1	C_33	33	0	0.1
C_5	5	0	0.1	C_34	34	0	0.1
C_6	6	0	0.1	C_35	35	0	0.1
C_7	7	0	0.1	C_36	36	0	0.1
C_8	8	0	0.1	C_37	37	0	0.1
C_9	9	0	0.1	C_38	38	0	0.1
C_10	10	0	0.1	C_39	39	0	0.1
C_11	11	0	0.1	C_40	40	0	0.1
C_12	12	0	0.1	C_41	41	0	0.1
C_13	13	0	0.1	C_42	42	0	0.1
C_14	14	0	0.1	C_43	43	0	0.1
C_15	15	0	0.1	C_44	44	0	0.1
C_16	16	0	0.1	C_45	45	0	0.1
C_17	17	0	0.1	C_46	46	0	0.1
C_18	18	0	0.1	C_47	47	0	0.1
C_19	19	0	0.1	C_48	48	0	0.1
C_20	20	0	0.1	C_49	49	0	0.1
C_21	21	0	0.1	C_50	50	0	0.1
C_22	22	0	0.1	C_51	51	0	0.1
C_23	23	0	0.1	C_52	52	0	0.1
C_24	24	0	0.1	C_53	53	0	0.1
C_25	25	0	0.1	C_54	54	0	0.1
C_26	26	0	0.1	C_55	55	0	0.1
C_27	27	0	0.1	C_56	56	0	0.1
C_28	28	0	0.1	C_57	57	0	0.1

APÉNDICE A. DATOS DE LOS SISTEMAS DE PRUEBA

---

C_29	29	0	0.1				
------	----	---	-----	--	--	--	--

Tabla A.20: Datos de los generadores del sistema eléctrico IEEE-57 nodos

<i>Número de elemento</i>	<i>Nodo de conexión</i>	<i>Nodo de referencia</i>	<i>Amplitud máxima</i>	<i>Ángulo de fase</i>
G_1	1	0	1.04	0.0
G_2	2	0	1.01	0.0
G_3	3	0	0.985	0.0
G_4	6	0	0.98	0.0
G_5	8	0	1.005	0.0
G_6	9	0	0.98	0.0
G_7	12	0	1.015	0.0

Tabla A.21: Datos de las ramas magnetizantes del sistema eléctrico IEEE-57 nodos

<i>Número de elemento</i>	<i>Nodo de conexión</i>	<i>Nodo de referencia</i>	<i>R (p.u.)</i>	<i>L (p.u.)</i>	<i>Exponente</i>
RM_1	57	0	0.1	0.1	5.0

# Apéndice B

## Cantidades en Por Unidad de Frecuencia Angular

El sistema por unidad (p.u., per-unit en inglés) de cualquier cantidad se define como la relación entre esta cantidad y la cantidad base y se expresa como un decimal (B.1). En el área de la Ingeniería Eléctrica y los sistemas eléctricos de potencia, normalmente las cantidades de potencia, voltaje, corriente e impedancia son representadas por valores en por unidad en lugar de cantidades reales. Estas cantidades para el mismo orden relativo son tratadas con el mismo grado de exactitud [Watson y Arrillaga, 2003].

$$p.u. = \frac{\text{Cantidad real}}{\text{Cantidad base}} \quad (\text{B.1})$$

Debido a que el uso de cantidades en por unidad es de uso común, es comprensible que en ocasiones ciertos autores omitan la explicación y no especifican la metodología que usaron para la transformación de cantidades, lo que puede provocar diferencias en la utilización de datos, como es el caso de los sistemas de prueba de IEEE donde las cantidades de capacitancia e inductancia están expresadas en *por unidad de frecuencia angular*, las cuales deben ser transformadas a cantidades reales de capacitancia e inductancia cuando se utilizan programas de simulación transitoria como lo es ATP, o de lo contrario podría existir inconsistencias en las simulaciones.

En las hojas de datos de los sistemas de prueba de IEEE las cantidades de capacitancia e inductancia son representadas en cantidades en por unidad de frecuencia angular. La metodología seguida para obtener las cantidades de capacitancia e inductancia en por unidad de frecuencia angular, se presenta en el siguiente desarrollo:

*Reactancia capacitiva:*

$$X_C = \frac{1}{2 \cdot \pi \cdot f \cdot C} = \frac{1}{\omega \cdot C} \quad (\text{B.2})$$

donde:

$$C = \frac{1}{\omega \cdot X_C} \quad (\text{B.3})$$

Para representar la capacitancia en cantidades en por unidad de frecuencia angular se tendría:

$$C_{p.u.} = \left( \frac{1}{\omega \cdot X_C} \right) \cdot \omega \quad (\text{B.4})$$

$$C_{p.u.} = \frac{1}{X_C} \quad (\text{B.5})$$

*Reactancia inductiva:*

$$X_L = 2 \cdot \pi \cdot f \cdot L = \omega \cdot L \quad (\text{B.6})$$

donde:

$$L = \frac{X_L}{\omega} \quad (\text{B.7})$$

Para representar a la inductancia en cantidades en por unidad de frecuencia angular se tendría:

$$L_{p.u.} = \left( \frac{X_L}{\omega} \right) \cdot \omega \quad (\text{B.8})$$

$$L_{p.u.} = X_L \quad (\text{B.9})$$

Por lo tanto cuando se forma el conjunto de EDO se tienen que utilizar cantidades reales de capacitancia e inductancia o podría existir el riesgo de no convergencia del sistema al EEP.

Para el caso del capacitor se tiene que:

$$i_C(t) = C \cdot \frac{dV_C}{dt} \quad (\text{B.10})$$

$$\frac{dV_C}{dt} = \frac{i_C(t)}{C} \quad (\text{B.11})$$

Convirtiendo a cantidades reales se tiene que:

$$\frac{dV_C}{dt} = \frac{i_C(t)}{C_{p.u.}} = \frac{i_C(t)}{\frac{C_{p.u.}}{\omega}} \quad (\text{B.12})$$

por lo tanto se tiene:

$$\frac{dV_C}{dt} = \omega \cdot \frac{i_C(t)}{C_{p.u.}} \quad (\text{B.13})$$

Para el caso del inductor se tiene que:

$$V_L(t) = L \cdot \frac{di_L}{dt} \quad (\text{B.14})$$

$$\frac{di_L}{dt} = \frac{V_L(t)}{L} \quad (\text{B.15})$$

Convirtiendo a cantidades reales se tiene que:

$$\frac{di_L}{dt} = \frac{V_L(t)}{L_{p.u.}} = \frac{V_L(t)}{\frac{L_{p.u.}}{\omega}} \quad (\text{B.16})$$

por lo tanto se tiene:

$$\frac{di_L}{dt} = \omega \cdot \frac{V_L(t)}{L_{p.u.}} \quad (\text{B.17})$$

Por lo que en esta tesis cada una de ecuaciones diferenciales que modelan a los diferentes elementos del sistema son multiplicadas por la velocidad angular ( $\omega$ ).

## Apéndice C

### Rama magnetizante en ATP

En los sistemas eléctricos de potencia existen elementos no lineales cuyos modelos se tienen que representar por medio de aproximaciones. En programas de simulación transitoria de circuitos eléctricos tales como ASTAP o SPACE no se presentan los modelos para estos elementos, pero esto no quiere decir que no existan, sino que el investigador tiene que realizar trabajo adicional para poder representar la dinámica de estos elementos no lineales con los cuales se desea realizar alguna simulación.

El fenómeno de saturación en transformadores de potencia es muy importante para la predicción del contenido armónico en sistemas eléctricos. No obstante, los fabricantes raramente dan a conocer las características de saturación de los transformadores. Convencionalmente cada curva de saturación puede estar dispuesta como una serie de puntos ( $\phi - i$ ), almacenados en una computadora para su uso posterior, por ejemplo, para determinar la corriente de magnetización. Sin embargo, es ineficiente obtener la corriente de magnetización punto por punto, debido a que este procedimiento puede requerir de un número considerable de puntos para una solución precisa y por lo tanto el tiempo de cómputo puede ser considerable.

La representación de la dinámica de estos elementos no lineales puede ser una tarea difícil y tardada ya que en ocasiones se logra solo con una metodología de prueba y error. En la Tabla C.1 se muestran los seis puntos utilizados para representar la saturación de la rama magnetizante en el modelo de ATP en términos de corriente y flujo magnético.

Tabla C.1: Valores utilizados en el modelo de la rama magnetizante en ATP

Corriente (Amp)	Flujo magnético (Wb-V)
0.0012	2.801E-6
0.0014	3.25E-6
0.022	0.00142
0.181	0.00201
1.23	0.00363
20	0.00445

La Figura (C.1) muestra la curva de saturación utilizada para representar a la rama magnetizante en el programa de simulación ATP.

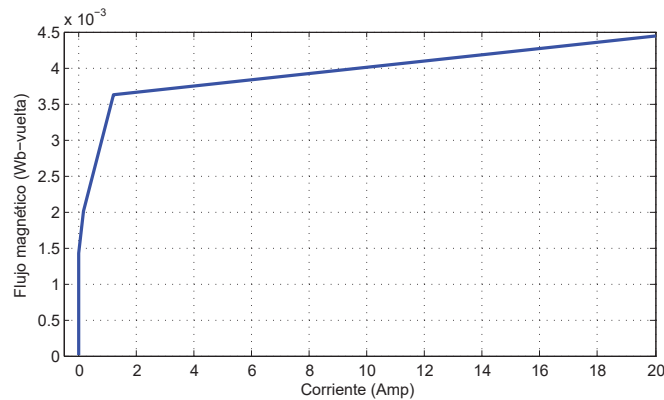


Figura C.1: Curva de saturación utilizada para la rama magnetizante en ATP

Esta representación de las características de saturación para transformadores de potencia modernos pueden ser obtenidas aplicando el método polinomial. La aproximación a las características de saturación experimentales puede realizarse por medio del polinomio propuesto en [Dick y Watson, 1981] de la forma:

$$i = a \cdot \varphi + b \cdot \varphi^\eta \tag{C.1}$$

donde:

$a$  Coeficiente correspondiente a la pendiente inicial.

$\eta$  Medida de la agudeza de la rodilla. Es un número impar debido a que la Ecuación (C.1) tiene simetría impar.

Posteriormente  $a$  y  $\eta$  pueden ser seleccionadas, el coeficiente  $b$  puede ser obtenido de manera que la curva pase por el punto  $i_j$  y  $\varphi_j$  y despejando  $b$ .

Para representar en forma polinomial la respuesta obtenida, las características de saturación seleccionada requiere una corriente  $i = 20$  cuando  $\varphi = 0.00445$ , con un coeficiente correspondiente a la pendiente en la parte lineal de  $a = 0.00116$  y por lo tanto  $\eta = 5$ , es una buena aproximación. La ecuación elegida es entonces:

$$i(t) = 0.00116 \cdot \varphi(t) + 1.1459 \times 10^{13} \cdot \varphi(t)^5 \quad (C.2)$$

La Figura (C.2) muestra la forma de saturación utilizando los valores de la tabla C.1 en ATP y la respuesta dada por la Ecuación (C.2).

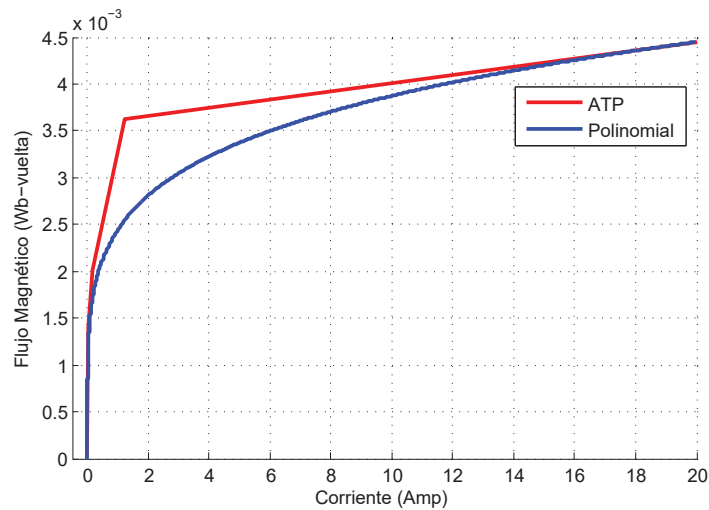


Figura C.2: Respuesta obtenida en ATP y con el método polinomial