

**DIFERENCIACIÓN AUTOMÁTICA EN LA
ESTABILIDAD DE SISTEMAS ELÉCTRICOS DE
POTENCIA**

TESIS

Que para obtener el grado de
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA ELÉCTRICA

presenta

Ing. Alejandro Zamora Méndez

Dr. Carlos Pérez Rojas

Director de Tesis

Universidad Michoacana de San Nicolás de Hidalgo

Diciembre 2007

Agradecimientos

A mis Padres: Ma. del Carmen Méndez y Cristóbal Zamora, quienes me apoyaron y alentaron todos los días de mi vida para llevar a cabo con éxito este objetivo.

A mis hermanos: Christian, Saúl, Andrea y Soledad, quienes me impulsaron a seguir esforzándome para lograr el éxito de mi titulación.

A mis tíos: Gerardo e Isabel, de quienes siempre encontré palabras de ánimo y fuerzas para seguir adelante.

Y por supuesto, agradezco infinitamente a mi asesor el Dr. Carlos Pérez Rojas, quien pacientemente tuvo a bien el apoyarme en la revisión y supervisión de este proyecto de Tesis, desde el inicio hasta su terminación.

Y a mis amigos: Héctor Daniel, Jorge, Jorge Alfredo, Héctor Ramiro, Roberto, José Luis, Karina, Guillermo, a quienes agradezco su confianza, apoyo y lealtad.

A DIOS por llenar mi vida de dicha y bendiciones.

Resumen

Los algoritmos de diferenciación automática aplicados al análisis de estabilidad angular transitoria en los sistemas eléctricos de potencia en el marco de referencia de coordenadas **abc** son presentados en este trabajo de Tesis. La diferenciación automática es una herramienta para el cálculo de derivadas en una forma rápida y exacta, ya sea en la forma de vector gradiente, Jacobiano o Hessiano. Esta técnica funciona aplicando sistemáticamente la regla de la cadena del cálculo diferencial a nivel de los operadores y librerías incorporadas del lenguaje de programación (C++, FORTRAN, etc), por lo que no está sujeto a errores numéricos debido a la truncación.

Se presenta un método tipo Newton para obtener el estado estacionario de una manera fácil y rápida, lo cual es de gran ayuda para el análisis de estabilidad angular transitoria ya que utilizar técnicas de punto a punto sería computacionalmente costoso en sistemas pobremente amortiguados. Se incorpora la diferenciación automática en el proceso de aceleración al estado estacionario para el cálculo de la matriz de transición de estados Φ , la cual depende del Jacobiano y por lo tanto éste se obtiene de manera rápida y exacta. Por lo tanto se podrá analizar el tiempo deseado de estado transitorio para posteriormente aplicar esta técnica de aceleración y de la solución obtenida poder concluir si el sistema es estable o inestable.

Abstract

Automatic differentiation algorithms, applied to the transient angular stability analysis in electrical power systems in the **abc** coordinate framework, are presented in this thesis. The automatic differentiation is a tool for computing exact derivatives in a quick and efficient way, in form of gradients, Jacobian and Hessian matrices. Automatic differentiation works by systematically applying the chain rule of differential calculus at the level of the operators and built-in libraries of programming languages (C++, FORTRAN). Therefore, it is not influenced by numerical errors due to truncation.

A Newton method to obtain fast steady-state response of lightly damped systems is presented. Point-to-point techniques produce high computational costs in the simulation of this kind of systems whereas the method presented in this thesis reduce the simulation time considerably, therefore this method is helpful in transient angular stability analysis. The automatic differentiation is incorporated in the acceleration to the steady-state process for the state transition matrix Φ computation; consequently the Jacobian matrix is obtained in a fast and accurate way because the state transition matrix depends of the Jacobian. Therefore, it is possible to carry out a transient analysis and afterwards, a steady-state analysis can be performed with this method in order to determine if the system is stable or unstable.

Índice

Agradecimientos	III
Resumen	V
Abstract	VII
Lista de figuras	XIII
Lista de tablas	XV
Lista de símbolos	XVII
Lista de publicaciones	XIX
1. Introducción	1
1.1. Introducción	1
1.2. Estado del arte	2
1.2.1. Estabilidad en SEP's	2
1.2.2. Estabilidad transitoria	3
1.2.3. Diferenciación automática	4
1.3. Objetivos	5
1.4. Justificación	6
1.5. Metodología	6
1.6. Descripción de capítulos	7
2. Diferenciación automática	9
2.1. Introducción	9
2.2. Codificación a mano	10
2.3. Diferenciación simbólica	11
2.4. Diferenciación numérica	11
2.5. Diferenciación automática	13
2.5.1. Regla de la cadena	14
2.5.2. Evaluación de la función	15
2.5.3. Evaluación de la derivada	18
2.5.4. Modo hacia adelante	19
2.5.5. Modo hacia atrás	19
2.6. Implementación de la diferenciación automática.	23
2.6.1. Sobrecarga de operadores	23
2.6.2. Transformación de código fuente	24
2.7. ADIFOR (Automatic Differentiation in Fortran)	24

2.7.1.	Modo híbrido de la diferenciación automática	25
2.7.2.	Cómo trabaja el sistema ADIFOR 2.0	27
2.7.3.	Preprocesador ADIFOR 2.0	28
2.7.4.	Sistema ADINTRINSICS	29
2.7.5.	Librería SPARSLINC	30
2.7.6.	Ejemplo del uso de ADIFOR	30
2.8.	Resumen	34
3.	Modelos de elementos del sistema eléctrico de potencia	35
3.1.	Introducción	35
3.2.	Métodos numéricos para la solución de EDA's	36
3.2.1.	Regla trapezoidal	37
3.2.2.	Runge-Kutta de cuarto orden	37
3.3.	Modelo de la máquina síncrona	37
3.3.1.	Ecuación de oscilación	42
3.3.2.	Sistema de excitación. Control automático de voltaje	42
3.3.3.	Primo-motores. Control automático de generación	43
3.4.	Modelo del transformador trifásico de cinco columnas	44
3.4.1.	Característica de saturación	50
3.4.2.	Matriz de conectividad delta-estrella aterrizada	52
3.5.	Modelo de líneas	53
3.6.	Modelo de capacitores	54
3.7.	Resumen	54
4.	Análisis de estabilidad angular transitoria considerando aceleración al estado estacionario	57
4.1.	Introducción	57
4.2.	Aceleración de la solución al estado estable	58
4.3.	Comparación de la matriz de transición de estados	62
4.4.	Estudio de energización del transformador considerando aceleración al estado estacionario	63
4.5.	Análisis de estabilidad angular transitoria del generador síncrono considerando aceleración al estado estacionario	66
4.6.	Análisis de estabilidad angular transitoria del generador síncrono. Caso inestable	70
4.7.	Análisis de estabilidad angular transitoria del sistema Generador-Transformador-Línea-Barra Infinita considerando aceleración al estado estacionario	72
4.7.1.	Caso de estudio	73
4.8.	Resumen	76
5.	Conclusiones	79
5.1.	Conclusiones	79
5.2.	Trabajos futuros	82
	Referencias	83

A. Parámetros de los elementos del SEP

89

Lista de figuras

1.1. Clasificación de la estabilidad en SEP's.	2
2.1. Subrutina para $y = f(x)$	15
2.2. Diagrama computacional de $y = \frac{1}{x_3} \cdot (x_1 \cos(x_4) + x_2 \sin(x_4) + ax_3^2 \cos(x_4) - bx_3^2 \sin(x_4))$	18
2.3. Implementación del modo hacia adelante.	20
2.4. Implementación del modo hacia atrás.	21
2.5. Modo hacia adelante versus modo hacia atrás para el cálculo de ∇y	22
2.6. Segmento de código generado por ADIFOR.	26
2.7. Diagrama computacional de $f = \prod_{i=1}^5 x(i)$	26
2.8. Descripción del sistema ADIFOR 2.0.	28
2.9. Programa main.f y subrutina squareroot.f.	31
2.10. Archivo simple.cmp.	31
2.11. Archivo simple.adf.	32
2.12. Código derivativo generado g_squareroot.f.	32
2.13. Controlador del código derivativo driver.f.	33
2.14. Comandos para compilar y enlazar el código derivativo ejecutable.	33
3.1. Diagrama esquemático de la máquina síncrona de p polos.	38
3.2. Sistema de excitación típico compensado.	42
3.3. Control de carga de un turbogenerador.	43
3.4. Control de carga de un hidrogenerador.	44
3.5. Transformador trifásico de cinco columnas.	45
3.6. Circuito equivalente eléctrico del transformador trifásico de cinco columnas.	45
3.7. Circuito equivalente magnético del transformador trifásico de cinco columnas.	47
3.8. Curva de saturación real y aproximada.	51
3.9. Conexión delta-estrella aterrizada.	52
3.10. Modelo de la línea trifásica con retorno por tierra.	53
3.11. Circuito equivalente con parámetros concentrados de la línea de transmisión.	53
3.12. Circuito equivalente de un banco de capacitores trifásico.	54
4.1. Diagrama de flujo, aceleración al estado estable.	61
4.2. Energización del transformador con $R_L = 1$ p.u.	64
4.3. Energización del transformador con $R_L = 20$ p.u.	64

4.4. Corrientes en el estator y rotor.	67
4.5. Variables mecánicas. a) plano de fase, b) velocidad angular, c) desplazamiento angular y d) par eléctrico.	67
4.6. Control de excitación	68
4.7. Control de velocidad	68
4.8. Caso inestable. a) ángulo, b) velocidad	70
4.9. Caso estable. a) ángulo, b) velocidad	71
4.10. Caso inestable. a) ángulo, b) velocidad	72
4.11. Sistema Generador-Transformador-Línea-Barra Infinita.	73
4.12. Corrientes en la máquina síncrona del sistema.	74
4.13. Variables mecánicas y par eléctrico del sistema.	75
4.14. Corrientes en el transformador delta-estrella aterrizada del sistema.	75

Lista de tablas

2.1. Composición de la función $y = \frac{1}{x_3} \cdot (x_1 \cos(x_4) + x_2 \sin(x_4) + ax_3^2 \cos(x_4) - bx_3^2 \sin(x_4))$	17
4.1. Comparación de Φ	63
4.2. DN vs DA para el transformador.	65
4.3. DN vs DA para el transformador reduciendo k	65
4.4. DN vs DA para el generador.	69
4.5. DN vs DA para el generador reduciendo k	69
4.6. DN vs DA para el sistema	76

Lista de símbolos

n	Número de variables independientes
m	Indicador de la última variable intermedia
$x_1 \dots x_n$	Variables independientes
$x_{n+1} \dots x_m$	Variables intermedias
$\nabla x_{n+1} \dots \nabla x_m$	Gradientes de las variables intermedias
$J_{i(n+1 \leq i \leq m)}$	Indicador de las variables originales y/o intermedias que son contenidas en $1, 2, \dots, i - 1$
e_i	i -ésimo vector base cartesiano en R^n
g	Vector gradiente de la función
V	Vector de voltajes
I	Vector de corrientes
λ	Vector de enlaces de flujo
R	Matriz de resistencias de los devanados
L	Matriz de inductancias
C	Capacitancia
ω_r	Velocidad del rotor
ω_s	Velocidad síncrona
ω_b	Velocidad base
H	Constante de inercia
T_m	Par mecánico
T_e	Par eléctrico
δ	Ángulo de carga
θ_r	Ángulo de posición del rotor, respecto a la fase a
p	Número de pares de polos
t	Tiempo
f	Frecuencia
L_{md}	Inductancia de magnetización del eje d
L_{mq}	Inductancia de magnetización del eje q
L_d	Inductancia síncrona del eje d
L_q	Inductancia síncrona del eje q
L_l	Inductancia de dispersión
T_R	Constante de tiempo del sensor
T_F	Constante de tiempo del estabilizador
T_A	Constante de tiempo del amplificador

T_E	Constante de tiempo del excitador
K_R	Ganancia del sensor
K_F	Ganancia del estabilizador
K_A	Ganancia del amplificador
K_E	Ganancia del excitador
T_g	Constante de tiempo del gobernador
T_T	Constante de tiempo de la turbina
N	Número de vueltas de un devanado
Γ	Inversa de L
\mathfrak{R}	Reluctancia
fmm	Fuerza magnetomotriz
l	Longitud
A	Área de la sección transversal
μ	Permeabilidad del material
ϕ	Flujo magnético
H	Intensidad magnética
B	Densidad magnética
Φ	Matriz de transición de estados
k	Número de puntos en un periodo de tiempo
h	Tamaño del paso
T	Periodo

Lista de publicaciones

“Automatic Differentiation in Dynamic and Steady State Analysis of the Synchronous Generator in abc” Alejandro Zamora Méndez, Carlos Pérez Rojas, Sigridt García Martínez and Héctor Daniel Villanueva Hernández. cerma (Aceptado). Electronics, Robotics and Automotive Mechanics Conference (CERMA’07), 2007.

“Cálculo del Jacobiano usando Diferenciación Automática.” Alejandro Zamora Méndez, Carlos Pérez Rojas, Sigridt García Martínez y Héctor Daniel Villanueva Hernández. Ciencia Nicolaita (Aceptado). Revista Científica de la Universidad Michoacana de San Nicolás de Hidalgo.

Capítulo 1

Introducción

1.1. Introducción

Los Sistemas Eléctricos de Potencia (SEP) son sistemas altamente no lineales y asimétricos que operan en un ambiente de constante cambio; cargas, salida en los generadores, topología y parámetros clave de operación cambian continuamente. Gracias a la aparición de las computadoras digitales y al gran avance que se ha tenido tanto en hardware como en software, es posible tener herramientas de simulación altamente eficientes en lo que respecta a rapidez de cálculo y al ahorro de memoria en la computadora, por lo que es posible realizar estudios y análisis en detalle de los SEP's en sus coordenadas de fase **abc**, y de esta forma considerar las asimetrías de la red eléctrica y las no linealidades que ésta presenta.

La investigación del presente trabajo de Tesis se enfoca al análisis de estabilidad angular transitoria de SEP's en el dominio del tiempo en el marco de referencia de coordenadas **abc**, con lo cual se analiza primeramente un tiempo de transitorio deseado, para después aplicar una técnica de aceleración al estado estable utilizando Diferenciación Automática y el resultado nos permitirá conocer si el sistema es estable o no.

1.2. Estado del arte

1.2.1. Estabilidad en SEP's

La estabilidad en sistemas de potencia es la capacidad de un sistema eléctrico de potencia, para una condición de operación inicial dada, de recuperar un estado de operación de equilibrio después de que haya sido expuesto a un disturbio físico, con la mayoría de las variables del sistema limitadas tal que prácticamente el sistema entero permanece intacto [Kundur04]. Debido a que los SEP's son dimensionalmente grandes y complejos, además de ser altamente no lineales, es necesario hacer modelos simplificados tales como los presentados en [Kundur94, Anderson94], y así, dependiendo de la naturaleza de la inestabilidad, magnitud del disturbio y escalas de tiempo, diferentes enfoques son usados para el modelo y análisis del sistema, por lo que existen diferentes definiciones y clasificaciones de la estabilidad en los SEP's [Kundur94, Anderson94, Bergen86, Saadat99, Arrillaga90]. Es importante mencionar que en ninguna de estas referencias se realiza un análisis de estabilidad en el marco de referencia de coordenadas **abc**, algunas son [Rodríguez04, Pérez04]. Una clasificación de estabilidad en sistemas de potencia es mostrada en la Figura 1.1 [Kundur04].

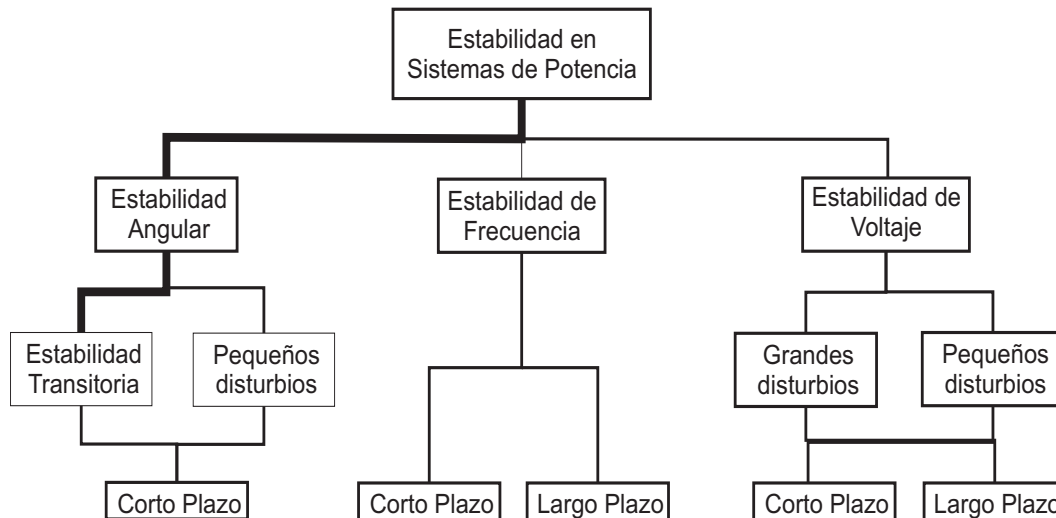


Figura 1.1: Clasificación de la estabilidad en SEP's.

En este caso, como se puede observar en la Figura 1.1, indicado por la línea más

gruesa, se estudia la estabilidad angular transitoria. Pero a diferencia de los modelos que se usan en las referencias indicadas anteriormente, aquí se aplican los modelos en coordenadas **abc** de los elementos que forman el sistema eléctrico de potencia. Este trabajo es un paso inicial para la aplicación de la Diferenciación Automática en la obtención del estado estable del sistema, por esta razón se aplica primeramente al transformador y generador en forma independiente y finalmente éstos acoplados a través de la línea de transmisión a un bus infinito. El evaluar la estabilidad angular transitoria o también conocida como estabilidad de primera oscilación, no siempre nos va a llevar a una interpretación correcta debido a la no linealidad del sistema y aunque la primera oscilación indique que el sistema va a llegar al estado estable puede que éste no sea alcanzado. De aquí la importancia de aplicar un algoritmo para obtener el estado estacionario de forma rápida y precisa, de esta manera analizar el comportamiento del desplazamiento angular y mediante la condición de operación a la cual llegue éste saber si el sistema es estable o inestable como se verá más adelante.

1.2.2. Estabilidad transitoria

Por definición [Kundur04], la estabilidad transitoria es la capacidad que tiene el sistema de potencia para mantener el sincronismo cuando es sometido a un disturbio. Tales disturbios ocurren continuamente sobre el sistema debido a pequeñas variaciones en las cargas y en la generación.

Ya que los SEP's se modelan completamente mediante un sistema de Ecuaciones Diferencio-Algebraicas (EDA's), las cuales deben ser resueltas en forma simultánea [Kundur94, Anderson94, Bergen86], se requieren de métodos numéricos eficientes para su solución [Gear71, Chapra99].

Otro enfoque interesante es la obtención rápida del estado estacionario mediante técnicas tipo Newton, tales como las desarrolladas por Aprille y Trick en circuitos de electrónica [Aprille72], este es el enfoque que se utiliza en toda la Tesis. De esta manera se puede analizar el tiempo deseado de estado transitorio y así evaluar la estabilidad transitoria, para posteriormente aplicar esta técnica de aceleración al estado estacionario y así evaluar el estado estable. Algunos trabajos que utilizan técnicas de este tipo en sistemas de potencia son [Usaola90, Semlyen95, García00b, Pérez04]. En [Usaola90, Semlyen95] traba-

jan con algoritmos híbridos en tiempo y frecuencia con elementos no lineales en el sistema. En [García00b] analizan el transformador monofásico y en [Pérez04] utiliza esta técnica en sistemas de potencia multimáquina en coordenadas **abc**.

Una parte fundamental en el proceso de la aceleración numérica es el cálculo de la matriz de transición de estados Φ . Esta investigación presenta varias formas de obtenerla, una de ellas es por medio de la diferenciación numérica, el resto es por medio de los algoritmos de Euler hacia atrás y hacia adelante, y por medio de la Regla Trapezoidal. Estos últimos métodos requieren del cálculo del Jacobiano, por lo que es necesario tener un cálculo derivativo exacto y eficiente, para esto la diferenciación automática es utilizada.

1.2.3. Diferenciación automática

La diferenciación automática es una poderosa herramienta computacional basada en la regla de la cadena [Griewank91] para evaluar las derivadas con respecto a las variables de entrada de las funciones definidas por un lenguaje de programación computacional de alto nivel, la cual sirve para obtener la información del cálculo de las derivadas y el valor de la función(es) en forma confiable, produciendo derivadas exactas instantáneamente [Griewank00].

Durante los últimos 30 años, el desarrollo de las herramientas de la diferenciación automática han sido conducidas por la necesidad de evaluar la derivada. Uno de los principales motivos para el desarrollo de estas herramientas ha sido la necesidad de valores exactos de las derivadas de primer orden y de orden superior de las funciones usadas en cálculos científicos. Como la necesidad de las herramientas para la diferenciación automática se ha intensificado, los avances en la ciencia de los lenguajes de programación y los sistemas modernos del algebra computacional han hecho la creación de tales herramientas más fácil. Numerosos investigadores se han aprovechado de estos avances creando poderosas herramientas para la diferenciación automática [Juedes91]. Algunas de estas herramientas desarrolladas son: ADOL-C [Griewank90] para las funciones escritas en C/C++, ADIFOR [Bischof94a] para programas escritos en FORTRAN, ADMIT-1 [Coleman00] y ADMAT [Coleman98, Verma] para código en Matlab, para una revisión más extensa sobre las herramientas de la diferenciación automática, ver [Juedes91, Bischof02].

Algunos trabajos que utilizan diferenciación automática en el análisis de sistemas de potencia son los presentados por [Jerolimsky94, Ibsais97, Orfanogianni00, Solodovnik01, Liu02, Zamora06]. En [Jerolimsky94] utilizan la diferenciación automática para el cálculo del Jacobiano en la simulación de sistemas de potencia, presentando como ejemplo un sistema de control de velocidad para turbinas de vapor. En [Ibsais97] la diferenciación automática es aplicada para calcular el Jacobiano y sensibilidades en flujos de potencia continuos. En [Orfanogianni00] la diferenciación automática es utilizada en la optimización de flujos de potencia con dispositivos FACTS en serie. En [Solodovnik01] esta técnica es aplicada para modelar componentes no lineales en sistemas de potencia. En [Liu02] presenta una metodología para el análisis de estabilidad de pequeña señal de sistemas de potencia con dispositivos FACTS obteniendo el Jacobiano con esta técnica. En [Zamora06] la diferenciación automática es utilizada para obtener el Jacobiano en flujos de potencia.

1.3. Objetivos

El objetivo principal es insertar la diferenciación automática en el proceso de aceleración propuesto por [Aprille72] en el cálculo de la matriz de transición de estados Φ , para obtener el estado estable y analizar si el sistema lo es o no. Esto se demuestra con aplicaciones a los sistemas de potencia modelados en coordenadas **abc**.

Objetivos particulares.

- Implementar en Fortran 77 todos los modelos del sistema de potencia en sus coordenadas de fase **abc**.
- Análisis e implementación en Fortran 77 de métodos numéricos para la solución de SEP's en sus coordenadas **abc**.
- Análisis de la estabilidad angular transitoria a los siguientes sistemas eléctricos en **abc**
 - Generador a barra infinita
 - Generador-Transformador-Línea-Barra Infinita

1.4. Justificación

En los SEP's la estabilidad angular transitoria de primera oscilación no siempre proporciona la información necesaria para determinar si el sistema es estable o inestable, requiriéndose por lo tanto un mayor tiempo de simulación. Para lo anterior la propuesta es utilizar un proceso de aceleración al estado estacionario mediante un método Newton con la finalidad de evitar la integración de las ecuaciones del sistema hasta que se llegue al estado estacionario. La diferenciación automática es incorporada en la aceleración al estado estacionario como una forma alterna de encontrar Φ que es utilizada en este proceso.

1.5. Metodología

El problema a solucionar es realizar un estudio y análisis de estabilidad angular transitoria en el generador y en el sistema Generador-Transformador-Línea-Barra Infinita en el marco de referencia de coordenadas de fase **abc**, por lo tanto, se mantendrán todos los modelos de los elementos en **abc**. Debido a que el comportamiento dinámico está definido fundamentalmente por la dinámica de los rotores de las unidades generadoras, es necesario tener un modelo del generador lo más detallado y completo posible que describa su comportamiento en una forma más cercana a la realidad. Este modelo también incluye sus controles de excitación y velocidad para tener un sistema regulado.

Otro elemento importante es el transformador trifásico, cuyo modelo es realizado partiendo de sus características magnéticas y eléctricas, lo cual nos permite observar qué pasa en su interior y qué ocasiona al sistema. La representación de la no linealidad del material ferromagnético del núcleo laminado se realizará por medio de la función arcotangente [Pérez99].

Las líneas de transmisión son modeladas también en sus coordenadas de fase **abc** con parámetros concentrados mediante ecuaciones diferenciales al igual que los bancos de capacitores. Las restricciones algebraicas son impuestas por la ley de Kirchoff de corrientes en los nodos de la red.

La solución numérica para el conjunto de EDA's del sistema de potencia se realizará por medio del Runge-Kutta de cuarto orden y mediante la Regla trapezoidal. Para esto se

realizan rutinas programadas en Fortran 77.

Del análisis de estabilidad angular transitoria se obtiene el estado estable en forma rápida mediante un método tipo Newton. Este método requiere del cálculo de la matriz de transición de estados Φ , por lo que esta investigación presenta varias formas de obtenerla. Una de ellas es por medio de la diferenciación numérica ya que es fácil de implementar. Las otras formas de obtener Φ es por medio de las fórmulas de Euler hacia atrás y hacia adelante y mediante la Regla Trapezoidal, es decir, utilizando un método implícito. Estas últimas requieren del cálculo del Jacobiano para su evaluación, y para la obtención de éste son utilizados los algoritmos de la diferenciación automática, utilizando el software llamado ADIFOR para obtener diferenciación automática de programas escritos en Fortran 77. De esta manera se hace una comparación entre estos métodos con el objetivo de analizar sus ventajas y desventajas.

El análisis de estabilidad angular transitoria del SEP se hará de acuerdo al disturbio y a las condiciones iniciales al aplicar el proceso de aceleración al estado estacionario y si el método converge al mismo estado estable inicial el sistema es estable. Por otro lado, si el método converge a otro estado estable no factible, el sistema es inestable.

1.6. Descripción de capítulos

En el Capítulo 1 se realiza una reseña breve de los antecedentes asociados al tema del presente trabajo así como el alcance que se pretende tener con esta investigación.

En el Capítulo 2 se muestra con más detalle la técnica de la diferenciación automática para la solución de derivadas, así como la descripción de tres métodos clásicos para el cálculo de las derivadas (codificación a mano, diferenciación numérica y diferenciación simbólica). Se presentarán las ventajas y desventajas de estos cuatro métodos. Se conocerán y discutirán las dos diferentes formas de implementar la diferenciación automática (transformación del código fuente y sobrecarga de operadores), así como sus dos enfoques (modo hacia adelante y modo hacia atrás), en donde la idea detrás de estos modos es utilizar la regla de la cadena para propagar las derivadas en la dirección hacia adelante o hacia atrás.

En el Capítulo 3 se describen los modelos matemáticos de los elementos que confor-

man a los SEP's. Se presenta el modelo completo de la máquina síncrona en sus coordenadas de fase **abc** incluyendo sus controles de excitación y velocidad. También es presentado el modelo del transformador trifásico de cinco columnas el cual estará en función de sus propiedades primitivas considerándose la saturación de este elemento.

En el Capítulo 4 se realizan las simulaciones para el análisis de estabilidad angular transitoria al generador y al sistema Generador-Transformador-Línea-Barra Infinita, éste último para observar la interacción de los modelos desarrollados para el generador y transformador en sus coordenadas **abc**. Se presentan resultados del transitorio y del estado estable, al cual se llega mediante un algoritmo tipo Newton para obtener en una forma rápida y fácil esta respuesta.

En el Capítulo 5 se presentan las conclusiones generales de este proyecto y posibles extensiones de la presente investigación.

Capítulo 2

Diferenciación automática

2.1. Introducción

La diferenciación frecuentemente constituye un papel importante en una gran variedad de aplicaciones de cálculo científico, tales como; estudios de optimización, análisis de sensibilidades, solución de sistemas de ecuaciones diferencio-algebraicas, solución de ecuaciones no lineales y solución de problemas inversos no lineales, en donde se requiere la información del cálculo de las derivadas, ya sea en forma de vector Gradiente, matriz Jacobiano o matriz Hessiano de una función dada, varias veces repetidamente.

Por esta razón es necesario tener un cálculo derivativo exacto y eficiente para la solución práctica de dichos problemas. Existen muchas opciones de calcular estas derivadas. Se pueden obtener manualmente, pero puede resultar difícil y/o tedioso, además de estar propensos a errores. También pueden obtenerse simbólicamente usando paquetes matemáticos tales como Maple o Mathematica pero lo malo es de que no son directamente aplicables para programas computacionales escritos en lenguajes de programación tales como C, además pueden fallar o simplemente ser computacionalmente ineficientes para problemas a gran escala. Otra forma de calcular derivadas es aproximándola usando diferenciación numérica, pero ésta puede resultar inexacta y lenta.

La diferenciación automática (DA), también conocida como diferenciación computacional, diferenciación algorítmica o diferenciación de algoritmos [Rall96], ha emergido

como una técnica alterna para calcular derivadas. Dado un programa que calcula una función numérica $F(x)$, una herramienta de la DA genera un programa que calcula $F'(x)$ (la derivada de $F(x)$). La DA trabaja sistemáticamente aplicando la regla de la cadena del cálculo diferencial al nivel de los operadores y librerías incorporadas del lenguaje, por lo que no está sujeto a los mismos errores numéricos de la diferenciación numérica. Mucha de la introducción ordinaria al cálculo diferencial se enfoca en entrenar estudiantes a producir fórmulas para derivar funciones también definidas por fórmulas. Consecuentemente la idea de evaluar derivadas en forma exacta, sin fórmulas para éstas, es novedoso para muchos. La exactitud teórica de la diferenciación automática proviene del hecho de que utiliza las mismas reglas de la diferenciación según lo aprendido en cálculo elemental, nada más que estas reglas son aplicadas a una especificación algorítmica de la función en lugar de una fórmula. Debido a esto la técnica de la diferenciación automática ha tenido un gran interés en los campos de la ingeniería y la ciencia.

Hay dos formas de implementar la diferenciación automática, una es transformando el código fuente y la otra es por medio de la sobrecarga de operadores. También existen dos enfoques de la DA, los llamados modo hacia adelante y modo hacia atrás.

2.2. Codificación a mano

La codificación a mano [Bischof94b, Orfanogianni00] fue usada extensivamente en el pasado, especialmente cuando el número de funciones a diferenciar es pequeña y las funciones son relativamente simples. Podemos obtener derivadas numéricas de funciones simples derivando las expresiones analíticas para las derivadas de primer orden y orden superior a mano y codificándolas en un lenguaje de programación de alto nivel tal como Fortran o C. La eficiencia del código resultante dependerá de la eficiencia del código derivativo, lo cual es responsabilidad del programador, por lo que para un código grande este proceso puede resultar tedioso y propenso a errores, además de que la eficiencia para el desarrollo del código requerirá de un gran esfuerzo por parte del programador, más aún, para funciones no lineales en donde las derivadas suelen ser más complicadas que la función misma. Por lo tanto, desarrollar un código derivativo a mano nos costará una cantidad considerable de

trabajo en comparación al desarrollo del código original, aunque es probable que esto resulte en un código más eficiente.

2.3. Diferenciación simbólica

La diferenciación simbólica [Ibsais97, Bischof91b, Bischof94b, Orfanogianni00] es un método ampliamente utilizado por científicos y matemáticos para el cálculo de derivadas. Existen paquetes comerciales basados en el cálculo simbólico, tales como **MAPLE**, **MACSYMA** y **MATHEMATICA** los cuales son poderosas herramientas computacionales de fácil uso que proporcionan derivadas exactas. Estos paquetes manipulan fórmulas para producir nuevas fórmulas, esto es, dada una función (fórmula), encontrará la derivada con respecto a sus variables independientes, produciendo una nueva fórmula (derivada). En general, estos paquetes de manipulación simbólica son incapaces de tratar con constructores tales como lazos o subrutinas que son inherentes en códigos computacionales. Por lo tanto se tienen dos opciones para obtener el valor numérico de la derivada

- Evaluar las expresiones en ciertos argumentos dentro del ambiente simbólico y exportar los valores numéricos al programa principal que llama a las derivadas.
- Exportar el código generado a un lenguaje de alto nivel (Fortran, C++) que puede ser directamente compilado y enlazado al programa principal.

El cálculo simbólico es sin duda una completa tecnología y digna de considerarse, pero puede no derivar buenas fórmulas computacionales. Su desventaja es debido a que corren dentro de ciertas limitaciones de los recursos del CPU, por lo que no pueden manejar procesos intensos en donde la dimensión de las matrices es grande, o en sí, si la descripción de la función es complicada, por lo que funciones que implican ramas o lazos no pueden manejarse fácilmente por la diferenciación simbólica.

2.4. Diferenciación numérica

En casos en donde el número de funciones a ser diferenciadas es muy grande o la complejidad de las funciones hacen que la derivación por medio de la codificación a mano sea

difícil, podemos obtener las derivadas de primer orden mediante la diferenciación numérica [Chapra99, Pérez04, Bischof94b, Orfanogianni00], mejor conocida como aproximación por diferencias divididas. Este método posee popularidad en algunas disciplinas científicas y áreas de la ingeniería ya que es muy fácil de implementar.

La aproximación por diferencias divididas a derivadas de primer orden está basada en la expansión truncada de la serie de Taylor. Existen varias aproximaciones mediante esta técnica, tales como:

- Diferencias hacia adelante.

$$\left. \frac{\partial f(x)}{\partial x_j} \right|_{x=x_0} \approx \frac{f(x_0 + h * e_j) - f(x_0)}{h} \quad (2.1)$$

- Diferencias hacia atrás.

$$\left. \frac{\partial f(x)}{\partial x_j} \right|_{x=x_0} \approx \frac{f(x_0 - h * e_j) - f(x_0)}{-h} \quad (2.2)$$

- Diferencias centrales.

$$\left. \frac{\partial f(x)}{\partial x_j} \right|_{x=x_0} \approx \frac{f(x_0 + h * e_j) - f(x_0 - h * e_j)}{2h} \quad (2.3)$$

donde h (perturbación en la variable x_j) es el tamaño del paso; esto es, la longitud del intervalo sobre el cual se hace la aproximación, y e_j es el j -ésimo vector base Cartesiano, lo cual nos asegura la perturbación para solo la j -ésima variable x_j .

El valor de la perturbación debe ser un parámetro pequeño y positivo, por lo que se elige un valor lo más pequeño posible para obtener una aproximación más exacta de la derivada, pero no tan cercano a cero ya que esto nos llevaría a resultados incorrectos o a una cancelación numérica.

La principal ventaja de esta técnica es de que solamente requerimos el código de la función original. Esto significa que para la evaluación numérica de la derivada, la función se está evaluando en diversos puntos y la diferencia en el valor de la función se utiliza como medida para obtener el valor de la derivada de la función. La principal desventaja es de que la exactitud para cada aproximación de la derivada es difícil de evaluar, ya que la elección del parámetro para la aproximación h puede resultar difícil de encontrar para

muchos problemas, en particular si el problema es altamente no lineal o si la función contiene ruido [Averick94].

También se puede observar que el costo computacional para obtener las derivadas se incrementa de acuerdo al número de variables x_j , por lo que la evaluación para las derivadas es $n + 1$ veces mayor que para la evaluación de la función misma, donde n es el número de variables de estado.

2.5. Diferenciación automática

En las secciones previas, se mostraron las técnicas tradicionales más populares para el cálculo de las derivadas, las cuales contienen importantes desventajas que las hacen imprácticas para muchas aplicaciones a gran escala. Idealmente, una técnica para el cálculo diferencial debe contener las siguientes propiedades:

- Cálculo de las derivadas automáticamente.
- Ser capaz de manejar códigos arbitrarios de alto nivel.
- Cálculo exacto de las derivadas (libres de errores de truncación).
- Cálculo de las derivadas a un costo independiente del número de variables.

La diferenciación automática [Griewank00, Rall96, Bischof93] es una tecnología relativamente reciente que posee estas cuatro propiedades, las cuales son proporcionadas por software existente para el cálculo automático de derivadas de una función general proporcionada por el usuario. La diferenciación automática evalúa el valor de la función y las derivadas con respecto a sus variables independientes o cualquier parámetro(s) de interés para el usuario en una forma rápida y exacta. Esta técnica se basa en el hecho de que todos los programas computacionales, no importa qué tan complicados sean, utilizan un sistema finito de operaciones elementales (unario o binario), según lo definido por el lenguaje de programación.

Este método también está basado en el hecho de que cualquier función, sin importar que tan compleja sea, se puede descomponer en un sistema de operaciones y funciones

elementales tales como la suma, resta, multiplicación, seno, coseno, exp, etc., por medio de la introducción de variables intermedias secuenciales.

El valor o la función calculada por el programa es simplemente una composición de estas funciones elementales. Uno puede calcular la información de la derivada de f exactamente y en una forma completamente automática, simplemente aplicando la regla de la cadena una y otra vez a la composición de estas operaciones elementales.

2.5.1. Regla de la cadena

La regla de la cadena [ChainRule99, Griewank91] permite que utilicemos nuestro conocimiento de las derivadas de las funciones $f(x)$ y $g(x)$ para encontrar la derivada de la función compuesta $f(g(x))$.

Supongamos una función $g(x)$ la cual es diferenciable en el punto x y $f(x)$ sea diferenciable al punto $g(x)$, entonces la función compuesta $f(g(x))$ es diferenciable a x . Además, dejemos $y = f(g(x))$ y $u = g(x)$, entonces

$$\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dx}$$

En otras palabras, la derivada de $f(g(x))$ es la derivada de f , evaluada en $g(x)$, multiplicado por la derivada de $g(x)$, es decir, sea $F(x) = f(g(x))$, entonces la derivada de $F'(x) = f'(g(x)) \cdot g'(x)$. Hay un número de resultados relacionados que también van bajo el nombre de la regla de la cadena. Por ejemplo, si $z = f(x, y)$, $x = g(t)$ y $y = h(t)$, entonces

$$\frac{dz}{dt} = \frac{\partial z}{\partial x} \frac{dx}{dt} + \frac{\partial z}{\partial y} \frac{dy}{dt}$$

La DA está basada en dos evaluaciones [Ibsais97]: la evaluación de la función y la evaluación de la derivada. La nomenclatura adoptada para las variables usadas por la DA es la siguiente:

n : número de variables independientes.

m : indicador de la última variable intermedia.

$x_1 \dots x_n$: variables independientes.

$x_{n+1} \dots x_m$: variables intermedias.

$\nabla x_{n+1} \dots \nabla x_m$: gradientes de las variables intermedias.

$J_{i(n+1 \leq i \leq m)}$: indicador de las variables originales y/o intermedias que son contenidas en $1, 2, \dots, i - 1$.

e_i : i -ésimo vector base cartesiano en R^n .

g : vector gradiente de la función.

2.5.2. Evaluación de la función

Dada una función $F(x_1, \dots, x_n)$, el valor de la función se calcula mediante la introducción de variables intermedias por cada operación aritmética elemental, tal como la adición y substracción, y por cada función elemental tales como las expresiones trigonométricas y exponenciales. La última variable intermedia es la que nos sirve como el valor de la función. Se observa que una variable intermedia puede depender de las variables independientes originales y/o de las variables intermedias previamente definidas. La descripción simbólica de esta evaluación de la función se muestra a continuación, consideramos la siguiente función $y = f(x)$, $f: R^n \rightarrow R$, representada por la subrutina de la Figura 2.1.

```

For  $i = n + 1, n + 2, \dots, m$ 
     $x_i = f_i(x_j)_{j \in J_i}$ 
end
 $f = x_m$ 

```

Figura 2.1: Subrutina para $y = f(x)$.

La subrutina de la Figura 2.1 representa a la función $f(x)$ como una composi-

ción de funciones elementales $\{f_i\}_{i=n+1}^m$, aquí las funciones elementales f_i dependen de las cantidades ya calculadas x_j , con $j \in J_i$, donde

$$J_i \subset \{1, 2, \dots, i-1\}, \forall i = n+1, n+2, \dots, m$$

En otras palabras f es la composición de $m-n$ funciones elementales o de librería f_i , cuyos gradientes

$$\nabla f_i = \left\langle \frac{\partial f_i}{\partial x_j} \right\rangle_{j \in J_i}$$

se asumen para que sean calculables en todos los argumentos de interés.

Por ejemplo, consideremos la siguiente función $y = f(x_1, x_2, x_3, x_4)$ dada por

$$y = \frac{1}{x_3} \cdot (x_1 \cos(x_4) + x_2 \sin(x_4) + ax_3^2 \cos(x_4) - bx_3^2 \sin(x_4)) \quad (2.4)$$

En donde se desea calcular el vector gradiente de y .

$$\nabla y = \left[\frac{\partial y}{\partial x_1} \quad \frac{\partial y}{\partial x_2} \quad \frac{\partial y}{\partial x_3} \quad \frac{\partial y}{\partial x_4} \right]^T \quad (2.5)$$

La función puede ser llevada a la forma de la Figura 2.1, descomponiéndola en funciones elementales (ver Tabla 2.1). Las variables $\{x_i\}_{i=5}^{19}$ en la Tabla 2.1, son las variables intermedias, donde los resultados de las funciones elementales o de librería son almacenadas. La Figura 2.2 demuestra cómo la función compuesta se puede visualizar por medio de un diagrama dirigido acíclicamente, donde cada nodo representa una variable, ya sea independiente o intermedia. Un trayecto que fluye del nodo x_j al nodo x_i representa una dependencia entre las dos variables. La dirección $j \rightarrow i$ asume que la variable x_i depende de la variable ya calculada x_j . Por lo tanto, hay una derivada parcial distinta de cero asociada con cada trayecto. El diagrama cuyos trayectos son etiquetados con sus derivadas parciales elementales será referido como *diagrama computacional* (Figura 2.2).

Este diagrama computacional en la Figura 2.2, exhibe visualmente el primer paso para cualquier aproximación de la DA. La expresión complicada se analiza en las variables intermedias $x_i = f_i(x_j), j \in J_i$ que posee resultados de operaciones unarias o binarias y llamadas de funciones simples. Las derivadas parciales $\frac{\partial f_i}{\partial x_j}$ de estas operaciones simples son calculadas usando las reglas de la diferenciación (regla del producto, regla del cociente,

Tabla 2.1: Composición de la función $y = \frac{1}{x_3} \cdot (x_1 \cos(x_4) + x_2 \sin(x_4) + ax_3^2 \cos(x_4) - bx_3^2 \sin(x_4))$.

VARIABLES INTERMEDIAS	FUNCIONES ELEMENTALES
$x_5 = x_3^2$	$x_5 = f_5(x_3)$
$x_6 = 1/x_3$	$x_6 = f_6(x_3)$
$x_7 = \cos(x_4)$	$x_7 = f_7(x_4)$
$x_8 = \sin(x_4)$	$x_8 = f_8(x_4)$
$x_9 = ax_5$	$x_9 = f_9(x_5)$
$x_{10} = bx_5$	$x_{10} = f_{10}(x_5)$
$x_{11} = x_7x_6$	$x_{11} = f_{11}(x_7, x_6)$
$x_{12} = x_8x_6$	$x_{12} = f_{12}(x_8, x_6)$
$x_{13} = x_1x_{11}$	$x_{13} = f_{13}(x_1, x_{11})$
$x_{14} = x_2x_{12}$	$x_{14} = f_{14}(x_2, x_{12})$
$x_{15} = x_9x_{11}$	$x_{15} = f_{15}(x_9, x_{11})$
$x_{16} = x_{12}x_{10}$	$x_{16} = f_{16}(x_{12}, x_{10})$
$x_{17} = x_{13} + x_{14}$	$x_{17} = f_{17}(x_{13}, x_{14})$
$x_{18} = x_{15} - x_{16}$	$x_{18} = f_{18}(x_{15}, x_{16})$
$x_{19} = x_{17} + x_{18}$	$x_{19} = f_{19}(x_{17}, x_{18})$
$y = x_{19}$	

primeras derivadas de las funciones elementales). Una vez construido el diagrama computacional, es posible calcular el gradiente de f acumulando las derivadas parciales elementales con la regla de la cadena.

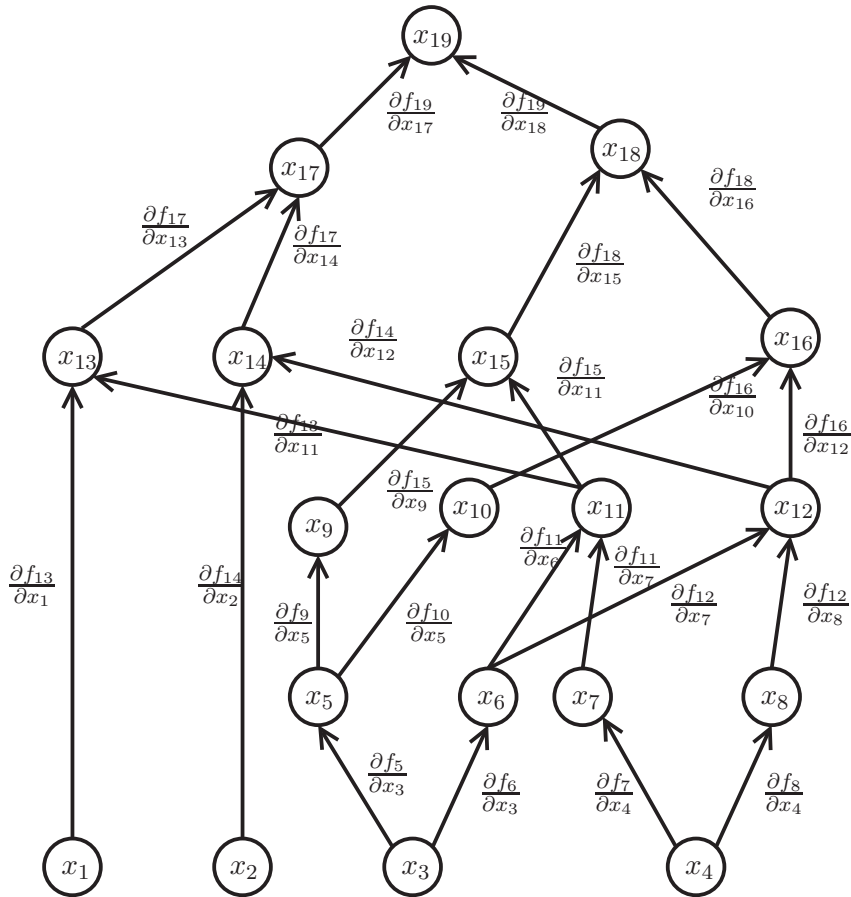


Figura 2.2: Diagrama computacional de $y = \frac{1}{x_3} \cdot (x_1 \cos(x_4) + x_2 \sin(x_4) + ax_3^2 \cos(x_4) - bx_3^2 \sin(x_4))$.

2.5.3. Evaluación de la derivada

Tradicionalmente fueron desarrolladas dos diferentes maneras de aplicar la regla de la cadena, los llamados modo hacia adelante y modo hacia atrás [Ibsais97, Griewank89, Orfanogianni00, Rall96, Griewank00, Jerolimsky94, Verma00]. En el modo hacia adelante las derivadas de las variables intermedias con respecto a las variables independientes son calculadas, mientras que en el modo hacia atrás las derivadas de la variable final (dependiente) con respecto a las variables intermedias son calculadas. Ambos modos tienen sus ventajas y desventajas por lo que se presentaran más detalladamente a continuación.

2.5.4. Modo hacia adelante

El modo hacia adelante mantiene las derivadas de cada variable intermedia con respecto a las variables independientes x_1, x_2, \dots, x_n , donde cada variable intermedia x_i tiene un vector gradiente $\nabla x_i, i = n + 1, \dots, m$. El proceso comienza en las variables independientes y trabaja de esta forma hacia las variables dependientes. La Figura 2.3 contiene una implementación del modo hacia adelante. Cada asignación de una variable intermedia x_i es aumentada por el cálculo de su vector gradiente ∇x_i . Por consiguiente, el espacio se debe asignar para $n + m$ variables escalares y $n + m$ vectores gradiente. Todos los cálculos son ciclos computacionales del gradiente, de modo que el trabajo computacional para calcular el vector gradiente depende del número de variables independientes.

En el caso de diferenciar $y = f(x), f : R^n \rightarrow R^m$, el modo hacia adelante calcula

$$\begin{bmatrix} \nabla y_1^T \\ \vdots \\ \nabla y_m^T \end{bmatrix} = \frac{\partial y}{\partial x} \begin{bmatrix} \nabla x_1^T \\ \vdots \\ \nabla x_n^T \end{bmatrix} \quad (2.6)$$

donde

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \dots & \frac{\partial y_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial y_m}{\partial x_1} & \dots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \quad (2.7)$$

Se observa que el cálculo del gradiente a través del modo hacia adelante se incrementa proporcionalmente con respecto al número de variables independientes.

2.5.5. Modo hacia atrás

En el modo hacia atrás mantenemos la derivada del resultado final con respecto a una cantidad intermedia. Estas cantidades generalmente se llaman adjuntas y miden la sensibilidad del resultado final con respecto a una cierta cantidad intermedia. En este modo asociamos un escalar \bar{x}_i con cada variable intermedia. Interpretamos cada \bar{x}_i como la derivada de una variable dependiente y con respecto a la variable intermedia $x_i, (\frac{\partial y}{\partial x_i})$.

El modo hacia atrás inicia con las variables dependientes y propaga las derivadas hacia las variables independientes acumulando las cantidades escalares adjuntas $\bar{x}_i = \frac{\partial y}{\partial x_i}$.

```

For  $i = 1 \rightarrow n$ 
   $\nabla x_i = e_i$ 
end
For  $i = n + 1 \rightarrow m$ 
   $x_i = f_i(x_j)_{j \in J_i}$ 
   $\nabla x_i = \sum_{j \in J_i} \frac{\partial f_i}{\partial x_j} \nabla x_j$ 
end
 $y = x_m$ 
 $\nabla f = \nabla x_m$ 

```

Figura 2.3: Implementación del modo hacia adelante.

De la regla de la cadena se sostiene que [Griewank89, Orfanogianni00]

$$\bar{x}_j = \sum_{i \in I_j} \frac{\partial f_i}{\partial x_j} \bar{x}_i \quad (2.8)$$

donde $I_j \equiv \{i \leq m : j \in J_i\}$. De este modo se puede observar que \bar{x}_j puede ser calculada solamente hasta que todas las \bar{x}_i con $i > j$ sean conocidas.

El modo hacia atrás consiste de un recorrido hacia adelante durante el cual se construye el diagrama computacional (variables y derivadas elementales) y de un recorrido hacia atrás para la acumulación de las derivadas. Para cada variable temporal ahora existe una cantidad escalar asociada, llevando la información de la derivada en lugar de un vector. La memoria se debe asignar para los $n + m$ escalares y todos los cálculos son con operandos escalares. En la Figura 2.4 se muestra una implementación del modo hacia atrás.

En el caso de querer diferenciar $y = f(x)$, $f : R^n \rightarrow R^m$, el modo hacia atrás calcula

$$[\bar{x}_1 \cdots \bar{x}_n] = [\bar{y}_1 \cdots \bar{y}_m] \frac{\partial y}{\partial x} \quad (2.9)$$


```

For  $i = n + 1, n + 2, \dots, m$ 
     $x_i = f_i(x_j)_{j \in J_i}$            {recorrido hacia adelante}
     $\bar{x}_i = 0$ 
end
 $y = x_m$ 
 $\bar{x}_m = 1$ 
For  $i = 1, 2, \dots, n$ 
     $\bar{x}_i = 0$ 
end
For  $i = m, m - 1, \dots, n + 1$ 
     $\bar{x}_j = \bar{x}_j + \frac{\partial f_i}{\partial x_j} \bar{x}_i, \forall j \in J_i$    {recorrido inverso}
end
For  $i = 1, 2, \dots, n$ 
     $\nabla f_i = \bar{x}_i$ 
end

```

Figura 2.4: Implementación del modo hacia atrás.

donde $\frac{\partial y}{\partial x}$ es definida en (2.7). Es decir, el modo hacia atrás permite los cálculos de combinaciones lineales de las filas de la matriz Jacobiano. Si se colocan los valores iniciales de $\bar{y}_i = 1, \bar{y}_{j \neq i} = 0$, produce $\frac{\partial y_i}{\partial x}$ en la fila de la matriz Jacobiano.

La Figura 2.5 muestra la implementación de calcular el ejemplo de la función (2.4) y su gradiente con el modo hacia adelante y el modo hacia atrás. Para los dos métodos la definición de $x_i, i = 5 \dots 19$ variables es el primer paso. Para la estimación de ∇y con el modo hacia adelante, el espacio se asigna para 19 vectores gradiente, y hay 19 ciclos computacionales del gradiente. Con el modo hacia atrás de la DA el espacio se asigna para 19 variables escalares y, aparte del último ciclo computacional del gradiente, todas las operaciones son escalares.

Modo Hacia Adelante

Inicializar:

$$\nabla x_1 = [1, 0, 0, 0]^T, \nabla x_2 = [0, 1, 0, 0]^T,$$

$$\nabla x_3 = [0, 0, 1, 0]^T, \nabla x_4 = [0, 0, 0, 1]^T,$$

$$\nabla x_i = 0, i = 5 \dots 19$$

$$\nabla x_5 = 2x_3 \cdot \nabla x_3$$

$$\nabla x_6 = -1/x_3^2 \cdot \nabla x_3$$

$$\nabla x_7 = -\sin x_4 \cdot \nabla x_4$$

$$\nabla x_8 = \cos x_4 \cdot \nabla x_4$$

$$\nabla x_9 = a \cdot \nabla x_5$$

$$\nabla x_{10} = b \cdot \nabla x_5$$

$$\nabla x_{11} = x_6 \cdot \nabla x_7 + x_7 \cdot \nabla x_6$$

$$\nabla x_{12} = x_8 \cdot \nabla x_6 + x_6 \cdot \nabla x_8$$

$$\nabla x_{13} = x_{11} \cdot \nabla x_1 + x_1 \cdot \nabla x_{11}$$

$$\nabla x_{14} = x_{12} \cdot \nabla x_2 + x_2 \cdot \nabla x_{12}$$

$$\nabla x_{15} = x_{11} \cdot \nabla x_9 + x_9 \cdot \nabla x_{11}$$

$$\nabla x_{16} = x_{12} \cdot \nabla x_{10} + x_{10} \cdot \nabla x_{12}$$

$$\nabla x_{17} = \nabla x_{13} + \nabla x_{14}$$

$$\nabla x_{18} = \nabla x_{15} - \nabla x_{16}$$

$$\nabla y = \nabla x_{17} + \nabla x_{18}$$

Modo Hacia Atrás

Inicializar:

$$\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4 = 0$$

$$\bar{x}_i = 0, i = 1 \dots 18$$

$$\bar{y} = 1$$

$$\bar{x}_{18} + = 1 \cdot \bar{y}$$

$$\bar{x}_{17} + = 1 \cdot \bar{y}$$

$$\bar{x}_{15} + = 1 \cdot \bar{x}_{18}$$

$$\bar{x}_{16} + = (-1) \cdot \bar{x}_{18}$$

$$\bar{x}_{14} + = 1 \cdot \bar{x}_{17}$$

$$\bar{x}_{13} + = 1 \cdot \bar{x}_{17}$$

$$\bar{x}_{12} + = x_{10} \cdot \bar{x}_{16}$$

$$\bar{x}_{10} + = x_{12} \cdot \bar{x}_{16}$$

$$\bar{x}_9 + = x_{11} \cdot \bar{x}_{15}$$

$$\bar{x}_{11} + = x_9 \cdot \bar{x}_{15}$$

$$\bar{x}_2 + = x_{12} \cdot \bar{x}_{14}$$

$$\bar{x}_{12} + = x_2 \cdot \bar{x}_{14}$$

$$\bar{x}_1 + = x_{11} \cdot \bar{x}_{13}$$

$$\bar{x}_{11} + = x_1 \cdot \bar{x}_{13}$$

$$\bar{x}_8 + = x_6 \cdot \bar{x}_{12}$$

$$\bar{x}_6 + = x_8 \cdot \bar{x}_{12}$$

$$\bar{x}_7 + = x_6 \cdot \bar{x}_{11}$$

$$\bar{x}_6 + = x_7 \cdot \bar{x}_{11}$$

$$\bar{x}_5 + = b \cdot \bar{x}_{10}$$

$$\bar{x}_5 + = a \cdot \bar{x}_9$$

$$\bar{x}_4 + = \cos(x_4) \cdot \bar{x}_8$$

$$\bar{x}_4 + = -\sin(x_4) \cdot \bar{x}_7$$

$$\bar{x}_3 + = -1/x_3^2 \cdot \bar{x}_6$$

$$\bar{x}_3 + = 2x_3 \cdot \bar{x}_5$$

$$\nabla y = [\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4]$$

Figura 2.5: Modo hacia adelante versus modo hacia atrás para el cálculo de ∇y .

En este ejemplo relativamente simple, con una variable dependiente y más de una variable independiente, el modo hacia atrás demuestra ser más económico en términos de

costo computacional y de requisitos de memoria. Sin embargo, a pesar de las ventajas del modo hacia atrás con respecto a complejidad, la implementación del modo hacia atrás puede ser muy complicada en el caso general, pues requiere la capacidad para acceder cada resultado intermedio en orden hacia atrás. Esto podría aumentar los requisitos de la asignación de memoria en un factor imprevisible [Orfanogianni00].

En resumen, para una función dada $\langle y_1, \dots, y_n \rangle = F(\langle x_1, \dots, x_m \rangle)$, donde n son las variables dependientes y m son las variables independientes, el modo hacia adelante probablemente será más eficiente en términos de operaciones numéricas, si el número de variables dependientes es más grande que el número de variables independientes, es decir $n \gg m$. Similarmente, para un número de variables independientes más grande que el número de variables dependientes, el modo hacia atrás es probablemente más eficiente que el modo hacia adelante, es decir para $m \gg n$ [Juedes91].

Las implementaciones actuales del modo hacia adelante y hacia atrás difieren más en términos de complejidad espacial. Una implementación típica del modo hacia adelante requiere menos de m (el número de variables independientes) veces el requisito de almacenaje de la función original. La mayoría de las implementaciones del modo hacia atrás requieren espacio adicional proporcional al número de operaciones numéricas ejecutadas por la función original.

2.6. Implementación de la diferenciación automática.

La diferenciación automática puede ser implementada en dos formas [Bischof02, Rall96]. La primera mediante la sobrecarga de operadores y la segunda transformando el código fuente. Estas implementaciones se describen a continuación.

2.6.1. Sobrecarga de operadores

Para obtener la derivada por medio de este enfoque, es necesario sobrecargar los operadores aritméticos básicos. Esto consiste en decirle al compilador que ahora cada número real será reemplazado por un par de números reales, en el cual el segundo corresponderá a la derivada.

Cada operación elemental sobre los números reales es sobrecargada, es decir, es internamente reemplazado por uno nuevo y se trabaja sobre pares de números reales, los cuales calcularán tanto su valor como el de su derivada. La ventaja de este enfoque es el hecho de que el programa original virtualmente queda sin cambios. La desventaja que se tiene del programa resultante es que correrá muy lentamente debido a la constante creación y destrucción de pares de números reales.

Este enfoque de la DA trabaja solo con lenguajes que soportan la sobrecarga de operadores, tales como C++ o Fortran 90. Algunas herramientas que utilizan este enfoque son; ADMAT [Coleman00, Coleman98, Verma] y ADOL-C [Griewank90].

2.6.2. Transformación de código fuente

Este enfoque, emplea técnicas de compilación para transformar el código fuente de un programa, en un nuevo código fuente que realice explícitamente el cálculo de las derivadas, por lo que es aplicable a cualquier lenguaje de programación.

La transformación del código fuente consiste en adherir al programa las nuevas variables, arreglos y estructuras de datos que nos mantendrán las derivadas, y en la adición de nuevas instrucciones que son las que calcularán la derivada.

La ventaja de este enfoque, es que el programa que resulta se puede compilar en un código eficiente. La desventaja es que esta transformación es enorme, por lo que no se podría realizar a mano en programas grandes, por lo tanto se generan herramientas capaces de realizar esta transformación en forma correcta y rápida. Algunas herramientas que utilizan la transformación de código fuente son; ADIFOR [Bischof94a] y TAPENADE [TAPENADE03].

2.7. ADIFOR (Automatic Differentiation in Fortran)

Existen varias herramientas sobre la diferenciación automática, por lo que una investigación más extensa sobre éstas se puede encontrar en [Juedes91].

Esta sección describe brevemente una introducción a la herramienta usada en este proyecto de Tesis (ADIFOR). Se describirá cómo usar la versión 2.0 (Revisión D) del sistema

ADIFOR [Bischof94a, Bischof91b, Bischof94b, Bischof92b, Bischof91a], la cual proporciona diferenciación automática de programas escritos en Fortran 77 para obtener derivadas de primer orden. El enfoque de transformación de código fuente para la diferenciación automática es utilizada en ADIFOR, por lo que, dada una subrutina (o colección de subrutinas) que describen a una función f , e indicando cuales variables en la lista de parámetros o bloques comunes corresponden a las variables independientes y dependientes con respecto a la diferenciación, ADIFOR analizará el programa para determinar cuales declaraciones en el código tienen que ser aumentadas con cálculos derivativos, de esta forma produce un nuevo código en Fortran 77 que calcula las derivadas de las variables dependientes con respecto a las variables independientes de la función f .

2.7.1. Modo híbrido de la diferenciación automática

ADIFOR usa un modo híbrido (Hacia adelante/Hacia atrás) [Bischof91b, Bischof94c, Bischof92b, Bischof91a] para el cálculo de derivadas, éste está principalmente basado sobre el modo hacia adelante, pero en cualquier declaración de asignación hay un paso local del modo hacia atrás. Ya que el modo hacia atrás es adecuado para la obtención de derivadas de funciones escalares, éste es aplicado en las declaraciones de asignación para acumular las derivadas parciales del lado izquierdo (definidas localmente como las variables dependientes) con respecto a cada variable del lado derecho (variables independientes locales). Al final, la aplicación de la regla de la cadena en el sentido del modo hacia adelante propagará las derivadas hacia las variables independientes.

Por ejemplo, sea el siguiente programa escrito en Fortran para el cálculo de $f = \prod_{i=1}^5 x(i)$

```
SUBROUTINE PROD(X, F)
REAL X(5), F
F = X(1) * X(2) * X(3) * X(4) * X(5)
RETURN
END
```

ADIFOR produce un programa cuya sección computacional es mostrada en la Figura 2.6, el cual calcula tanto el valor de la función como el de sus derivadas. El diagrama computacional para el código generado se muestra en la Figura 2.7.

```

r$3 = x(1) * x(2)
r$5 = r$3 * x(3)
r$7 = r$5 * x(4)
-----
r$4bar = x(5) * x(4)
r$5bar = x(5) * r$5  =>
r$6bar = r$4bar * x(3)
r$7bar = r$4bar * r$3
r$8bar = r$6bar * x(2)
r$9bar = r$6bar * x(1)
-----
do  g$i$ = 1, g$p$
    g$f(g$i$) =  r$8bar * g$x(g$i$, 1)      Modo Hacia Adelante:
                + r$9bar * g$x(g$i$, 2)
                + r$7bar * g$x(g$i$, 3)  => Ensamble de ∇f de  $\frac{\partial f}{\partial x(i)}$ 
                + r$5bar * g$x(g$i$, 4)
                + r$7 * g$x(g$i$, 5)      y ∇x(i),  i = 1, ..., 5
enddo
f = r$7 * x(5)      }      Cálculo del valor de la función
    
```

Figura 2.6: Segmento de código generado por ADIFOR.

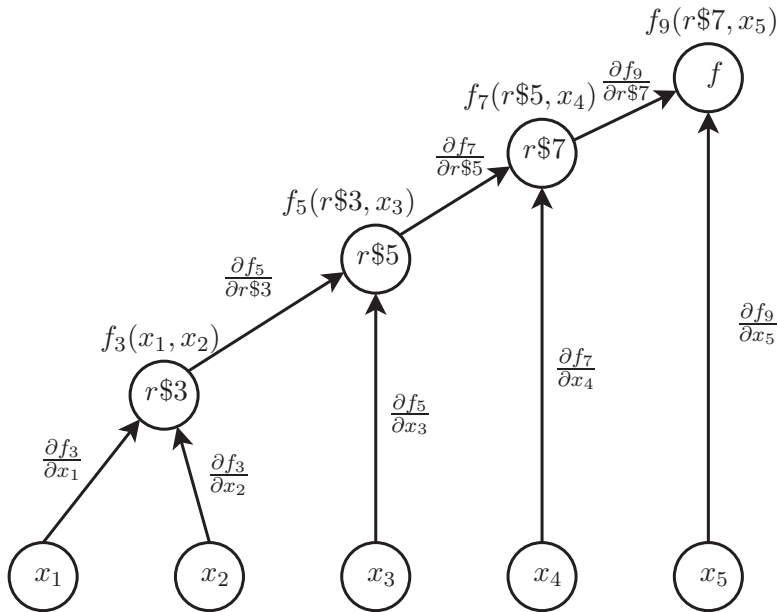


Figura 2.7: Diagrama computacional de $f = \prod_{i=1}^5 x(i)$.

El código generado por ADIFOR en la Figura 2.6, es un código ya limpio [Bischof94a], es decir que ya se eliminaron algunas variables utilizadas durante el proceso de transformación del código, las cuales ya no se requieren para el cálculo de la función y sus derivadas, en pocas palabras este código ya está optimizado por el mismo ADIFOR. Por esta razón las variables mostradas en el código no siguen un orden ascendente continuo. El signo \$ es usado para enfatizar las variables generadas por ADIFOR. La variable **g\$p** denota el número de derivadas (direccionales) a ser calculadas. Por lo que si la variable **x** (vector que contiene las variables independientes) es inicializada a un valor deseado x_0 , **g\$p** = 5 y el arreglo **g\$x** a ser la matriz identidad de tamaño 5×5 , con estas consideraciones el vector de salida **g\$f** contendrá las derivadas de la función f con respecto a las variables independientes x_i , para $i = 1, \dots, 5$, $\frac{\partial f(x)}{\partial x}|_{x=x_0}$. Nótese que ninguna de las subexpresiones comunes $x(i) * x(j)$ son recalculadas en la sección del modo hacia atrás para la obtención de $\frac{\partial f}{\partial x(i)}$, esto debido a que el producto total se calcula mediante un árbol binario y las propias piezas del producto son reutilizadas para el cálculo derivativo.

La Figura 2.7 muestra el diagrama computacional donde se observa la descomposición de la función y las derivadas parciales intermedias, cuyo conocimiento es fácil de obtener simplemente aplicando las reglas de la diferenciación a las funciones intermedias generadas. Estas derivadas parciales intermedias son utilizadas para el cálculo de las derivadas parciales de la función original con respecto a las variables independientes, es decir $\frac{\partial f}{\partial x(i)}$ para $i = 1, \dots, 5$.

2.7.2. Cómo trabaja el sistema ADIFOR 2.0

El sistema ADIFOR 2.0 consiste de tres principales componentes (ver Figura 2.8):

- Preprocesador ADIFOR 2.0.** El preprocesador ADIFOR 2.0 analiza sintácticamente el código, realiza ciertas normalizaciones al código, determina qué variables tienen que ser aumentadas con objetos derivativos y genera el código derivativo con plantillas a cada llamada de las funciones intrínsecas de Fortran 77, además, si se desea, llama a las rutinas de SPARSLINC.

- **Sistema ADINTRINSICS.** El sistema ADINTRINSICS expande las llamadas a las funciones intrínsecas de Fortran 77 al código de Fortran 77.
- **Librería SPARSLINC.** La librería SPARSLINC proporciona ayuda transparente para la dispersidad en los cálculos derivativos.

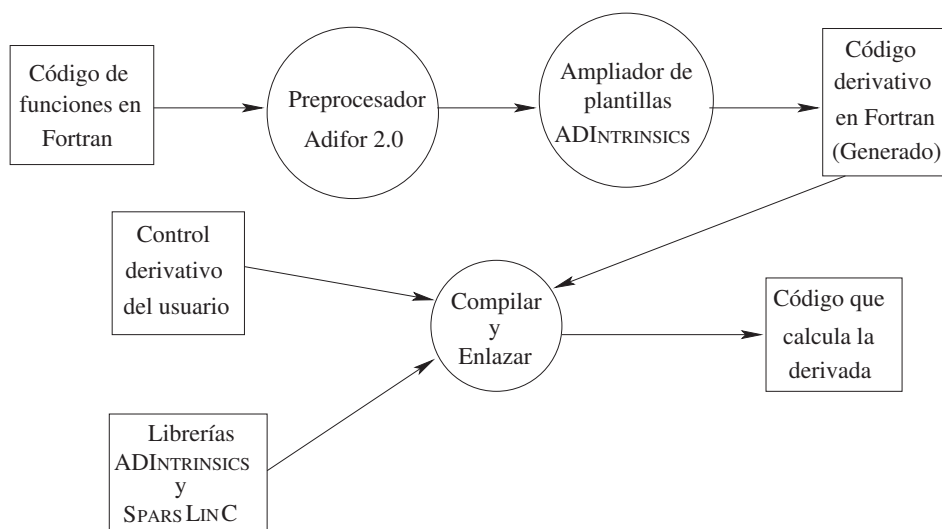


Figura 2.8: Descripción del sistema ADIFOR 2.0.

2.7.3. Preprocesador ADIFOR 2.0

El preprocesador ADIFOR 2.0 es la primer fase para la generación de código derivativo. El preprocesador realiza las siguientes fases:

- *Canonización del Código.* En esta fase, el código en Fortran es esencialmente reescrito para cumplir con las reglas de la transformación del código fuente impuestas por ADIFOR 2.0, esto consiste principalmente del proceso de romper las asignaciones del lado derecho en pequeñas piezas y almacenar los resultados intermedios a las variables temporales. La declaración de funciones es extendida en código en línea y ciertas modificaciones se hacen para asegurar que todas las variables que aparecen sobre el lado derecho de una declaración sean del mismo tipo.

- *Nominación de Variables.* En esta fase las variables *activas* son detectadas. Las variables activas son aquellas cuyos valores dependen de los valores de las variables independientes y tienen influencia en los valores de las variables dependientes en el cálculo derivativo. Las variables que no influyen en las variables dependientes del cálculo derivativo son llamadas variables *pasivas*.
- *Generación de Código.* Después de que las variables activas han sido nominadas, el código derivativo es generado para cada declaración que contenga una variable activa, además de que los objetos derivativos son asignados. Por lo que, para cada declaración, primero se genera el código derivativo usando el enfoque del modo hacia atrás de la diferenciación automática descrita en la Sección 2.5.5. Mediante esta forma se reescribe el código original (entrada) y se proveen fórmulas para las derivadas locales, también se eliminarán las multiplicaciones por 1.0 y las sumas a 0.0 reduciéndose el número de variables a ser asignadas. Las llamadas a funciones intrínsecas de Fortran son reemplazadas por plantillas que serán revisadas por el sistema ADINTRINSICS.

2.7.4. Sistema ADINTRINSICS

El sistema ADINTRINSICS es activado con las llamadas de funciones intrínsecas de Fortran. Provee al ADIFOR con reglas que definen el comportamiento del código generado para los eventos con excepción, casos donde la función elemental es evaluada en los puntos donde la derivada de la función no está definida. Por ejemplo, para la función $\mathbf{z} = \mathbf{sqrt}(\mathbf{x})$ una excepción estaría dada para $\mathbf{x} = \mathbf{0.0}$.

El sistema ADIFOR 2.0 confía en el sistema ADINTRINSICS, el cual es desarrollado principalmente, tal que el usuario pueda fácilmente modificar el comportamiento del ADIFOR en los casos en donde la opción por default resulte ser inapropiada. El sistema ADINTRINSICS también permite cambiar entre diferentes gustos el mensaje del reporte de error.

En resumen, el sistema ADINTRINSICS provee un mecanismo para:

- definir un comportamiento razonable por omisión en casos donde la derivada de una función de Fortran 77 no esté definida,

- provee un mecanismo para reportar el error el cual da información detallada de las excepciones ocurridas,
- también el usuario fácilmente puede modificar el manejo de excepciones, y
- fácilmente puede extender este manejador para nuevas funciones.

Para una descripción más detallada sobre el sistema ADINTRINSICS ver [Bischof94a, Bischof94b].

2.7.5. Librería SPARSLINC

La librería SPARSLINC (Sparse Linear Combinations) es una librería escrita en ANSI C para la implementación de la dispersidad de la operación de combinaciones lineales de vectores.

$$w = \sum_{i=1}^k \alpha_i u_i \quad (2.10)$$

La ecuación (2.10) es la operación principal en la propagación de derivadas (regla de la cadena), donde los vectores u_i son los objetos derivativos de las variables activas y los multiplicadores α_i son las derivadas locales escalares.

La librería SPARSLINC calcula el patrón de dispersidad de las derivadas como un subproducto del proceso de la DA. Los usuarios tienen que especificar la opción de dispersión antes de correr el ADIFOR, e inicializar la matriz semilla [Bischof94a] que es la matriz identidad. En la salida los usuarios obtienen el Jacobiano en una estructura de datos dispersa.

2.7.6. Ejemplo del uso de ADIFOR

Se mostrará el manejo de ADIFOR 2.0 usando las configuraciones por default mediante el simple programa mostrado en la Figura 2.9 [Bischof94a]. El procedimiento **squareroot** asigna la raíz cuadrada del valor de la variable **x** a la variable **y**. Se hará una explicación limitada de los pasos requeridos para el cálculo de la derivada de **squareroot** a un valor especificado por el usuario utilizando ADIFOR. En [Bischof92a] se muestra otro ejemplo paso a paso de como utilizar ADIFOR y todas sus opciones.

<pre> program main real x, y read *, x call squareroot(x, y) print *, y end </pre>	<pre> subroutine squareroot(x, y) real x, y y = sqrt(x) end </pre>
--	--

Figura 2.9: Programa main.f y subrutina squareroot.f.

La secuencia a seguir es:

1. Construir un archivo con extensión *.cmp, que liste los nombres de todos los archivos fuente escritos en FORTRAN 77 que constituyen al programa del ejemplo (**simple.cmp**). La Figura 2.10 muestra este archivo.

<pre> main.f squareroot.f </pre>

Figura 2.10: Archivo simple.cmp.

2. Construir un archivo con extensión *.adf (**simple.adf**), el cual le dice al Preprocesador ADIFOR que derive el procedimiento nombrado **squareroot** con la variable independiente **x** y la variable dependiente **y**, es decir, genera el código para calcular la derivada $\frac{dy}{dx}$. El archivo es mostrado en la Figura 2.11. La opción AD_TOP le dice al ADIFOR el nombre de la rutina a ser diferenciada, la opción AD_PMAX es el máximo número de variables independientes de la función a ser diferenciada, la opción AD_IVARS lista los nombres de las variables independientes, la opción AD_DVARS lista los nombres de las variables dependientes y la opción AD_PROG contiene el archivo *.cmp.
3. Generar en el subdirectorio **output_files** [Bischof94a], mediante el comando **Adi-for2.1 AD_SCRIPT = simple.adf** el archivo **g_squareroot**. Este código generado se muestra en la Figura 2.12. Este código contiene el aumento derivativo creado por

```

AD_TOP = squareroot
AD_PMAX = 1
AD_IVARS = x
AD_DVARS = y
AD_PROG = simple.cmp

```

Figura 2.11: Archivo simple.adf.

ADIFOR para la función a derivar (squareroot.f), el cual calcula tanto el valor de la función como el de sus derivadas.

```

subroutine g_squareroot(g_p_, x, g_x, ldg_x, y, g_y, ldg_y)
real x, y
integer g_pmax_
parameter (g_pmax_ = 1)
integer g_i_, g_p_, ldg_y, ldg_x
real r1_p, r2_v, g_y(ldg_y), g_x(ldg_x)
integer g_ehfid
data g_ehfid /0/
if (g_p_ .gt. g_pmax_) then
print *, 'Parameter g_p_ is greater than g_pmax_'
stop
endif
r2_v = sqrt(x)
if ( x .gt. 0.0e0 ) then
r1_p = 1.0e0 / (2.0e0 * r2_v)
else
call ehufSV (9,x, r2_v, r1_p, 'g_squareroot.f',37)
endif
do g_i_ = 1, g_p_
g_y(g_i_) = r1_p * g_x(g_i_)
enddo
y = r2_v
end

```

Figura 2.12: Código derivativo generado g_squareroot.f.

4. Crear el control del código derivativo **driver.f** como se muestra en la Figura 2.13.

Este código de control invoca a **g_squareroot** con un valor de **x** especificado por el

usuario para calcular el valor de y y su derivada $\frac{dy}{dx}$.

```

program driver
real x, y
real g_x(1), g_y(1)
read*, x
g_x(1) = 1.0
call g_squareroot(1, x, g_x, 1, y, g_y, 1)
call ehrp
print*, y
end

```

Figura 2.13: Controlador del código derivativo driver.f.

5. Compilar y enlazar **driver.f**, **g_squareroot.f** y las librerías del manejador de excepciones ADINTRINSICS usando los comandos mostrados en la Figura 2.14, para construir el archivo ejecutable para el cálculo derivativo deseado.

```

g77 -c driver.f
g77 -o output_files/g_squareroot.f
$AD_LIB/lib/ReqADIntrinsics-$AD_OS.o $AD_LIB/lib/libADIntrinsics-$AD_OS.a

```

Figura 2.14: Comandos para compilar y enlazar el código derivativo ejecutable.

La solución analítica para este caso es

$$y = \sqrt{x} \qquad \frac{dy}{dx} = \frac{1}{2\sqrt{x}}$$

Por lo tanto para un valor de $x = 2.0$ se obtiene

	ADIFOR	Cálculo analítico
y	1.414214	1.414214
$\frac{dy}{dx}$	0.3535534	0.3535534

Por lo que se observa que son los mismos valores.

2.8. Resumen

A lo largo de este Capítulo se mostraron algunas de las formas más comunes de calcular derivadas, destacando entre ellas la así llamada diferenciación automática, esto debido a las desventajas que poseen las demás herramientas las cuales las hacen imprácticas para muchas aplicaciones grandes. La DA es una poderosa herramienta con la cual obtenemos tanto el valor de la función, como sus derivadas de manera rápida y exacta en forma automática.

Se mostraron las diferentes formas de aplicar la regla de la cadena a la diferenciación automática, las cuales son el modo hacia adelante y el modo hacia atrás, presentándose sus ventajas y desventajas.

Se analizaron las dos formas de implementar la diferenciación automática (sobrecarga de operadores y transformación del código fuente), así como sus ventajas y desventajas, utilizando ADIFOR para el presente trabajo de Tesis, cuya implementación de la DA es por medio de la transformación del código fuente.

Se describió brevemente sobre el uso de la herramienta ADIFOR (Automatic Differentiation in Fortran), la cual proporciona DA de programas escritos en Fortran 77 para obtener derivadas de primer orden y así poder aplicarla para la obtención rápida al estado estable en sistemas eléctricos de potencia (ver Capítulo 4).

Capítulo 3

Modelos de elementos del sistema eléctrico de potencia

3.1. Introducción

En los sistemas eléctricos de potencia existen elementos tales como el generador, líneas, transformadores y cargas, por lo que se requiere de un gran conocimiento y modelos matemáticos más realistas y detallados sobre éstos, para de igual forma obtener resultados de su análisis muy cercanos a la realidad.

En los sistemas eléctricos actuales la mayor parte de la generación de energía eléctrica se lleva a cabo por medio de generadores síncronos, los cuales deben suministrar la energía eléctrica lo más económica y con la mayor calidad que sea posible, para esto hay que mantener el voltaje y frecuencia dentro los límites permisibles. Por esta razón se requiere de dos controles principales: el Control Automático de Frecuencia-Carga y el Control Automático de Voltaje, los cuales controlan la potencia real y reactiva respectivamente. Por lo tanto es importante tener un modelo, lo más completo posible, que describa el comportamiento eléctrico y electromecánico de la máquina síncrona. El establecer un modelo implica pasar de una máquina física a un modelo matemático. Se profundizará el análisis de la máquina síncrona tomando como referencia sus circuitos magnéticos, con lo cual se obtienen las inductancias en función de las características y dimensiones del material del estator y rotor,

para una descripción más detallada sobre este modelo ver [Pérez04].

El transformador de potencia juega un papel importante en la transmisión y conversión de energía, ya que se utiliza en los sistemas de transmisión para elevar los niveles de voltaje y así reducir pérdidas durante la transmisión de energía eléctrica a grandes distancias, posteriormente el transformador reduce el nivel de voltaje para que llegue a los centros de consumo. Se modelará el transformador trifásico de cinco columnas el cual estará en función de sus propiedades primitivas, donde la representación de la no linealidad del material del núcleo laminado se hará por medio de la función arcotangente [Pérez04, Pérez99].

El modelo de las líneas de transmisión es en coordenadas **abc** considerando que se conocen los datos en estas coordenadas y reducidos a la matriz de 3x3, es decir que ya están incluidos los efectos de retorno por tierra y los hilos de guarda. Con respecto al modelo de cargas no es necesario para nuestro caso, ya que los casos propuestos son a barra infinita.

Se describirán los métodos numéricos más utilizados para la solución numérica de los modelos matemáticos que conforman a los sistemas eléctricos de potencia, los cuales son, la Regla Trapezoidal y el Runge-Kutta de Cuarto Orden [Gear71, Chapra99].

3.2. Métodos numéricos para la solución de EDA's

Debido a que los sistemas eléctricos de potencia están conformados por sistemas de ecuaciones diferencio-algebraicas (EDA's), es necesario utilizar algún método numérico para su solución. Aquí, las ecuaciones diferenciales describen la dinámica del sistema, mientras que las ecuaciones algebraicas describen a la red y a las restricciones algebraicas del generador. La solución de estos dos sistemas de ecuaciones definen el estado electromecánico del sistema para cualquier instante de tiempo.

El sistema de EDA's a resolver tiene la forma

$$y' = f(t, y, z) \tag{3.1}$$

$$0 = g(t, y, z) \tag{3.2}$$

Los métodos más recomendados debido a su simplicidad en su implementación, así como sus propiedades de estabilidad y exactitud, son la Regla Trapezoidal y el Runge-Kutta

de Cuarto Orden.

3.2.1. Regla trapezoidal

La formulación de la Regla Trapezoidal es

$$y_n = y_{n-1} + \frac{h}{2}[f(t_n, y_n) + f(t_{n-1}, y_{n-1})] \quad (3.3)$$

La Regla Trapezoidal es un método implícito A-estable [Gear71] y requiere de un proceso iterativo, por lo que el sistema se evalúa n veces hasta que se cumpla una tolerancia dada, esto por cada instante de integración. El proceso iterativo utilizado en este trabajo es el método de Newton con una tolerancia de $tolRT = 10^{-3}$.

3.2.2. Runge-Kutta de cuarto orden

La formulación del Runge-Kutta de cuarto orden es

$$\begin{aligned} K_1 &= f(t_{n-1}, y_{n-1}) \\ K_2 &= f\left(t_{n-1} + \frac{h}{2}, y_{n-1} + \frac{K_1}{2}\right) \\ K_3 &= f\left(t_{n-1} + \frac{h}{2}, y_{n-1} + \frac{K_2}{2}\right) \\ K_4 &= f(t_{n-1} + h, y_{n-1} + K_3) \end{aligned} \quad (3.4)$$

$$y_n = y_{n-1} + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \quad (3.5)$$

El método del Runge-Kutta de cuarto orden es un método explícito y en este caso el sistema se evalúa cuatro veces por cada instante de integración.

3.3. Modelo de la máquina síncrona

En el modelo del generador síncrono [Pérez04, Krause87, Krause89], se considera la simetría en las fases y carcasa. En la Figura 3.1 se muestra el diagrama esquemático para un generador de \mathbf{p} polos.

En la Figura 3.1 se muestra la distribución de los devanados en los ejes de la máquina síncrona, tanto en el estator ($\mathbf{a}, \mathbf{b}, \mathbf{c}$), como en el rotor, donde se tienen los devanados

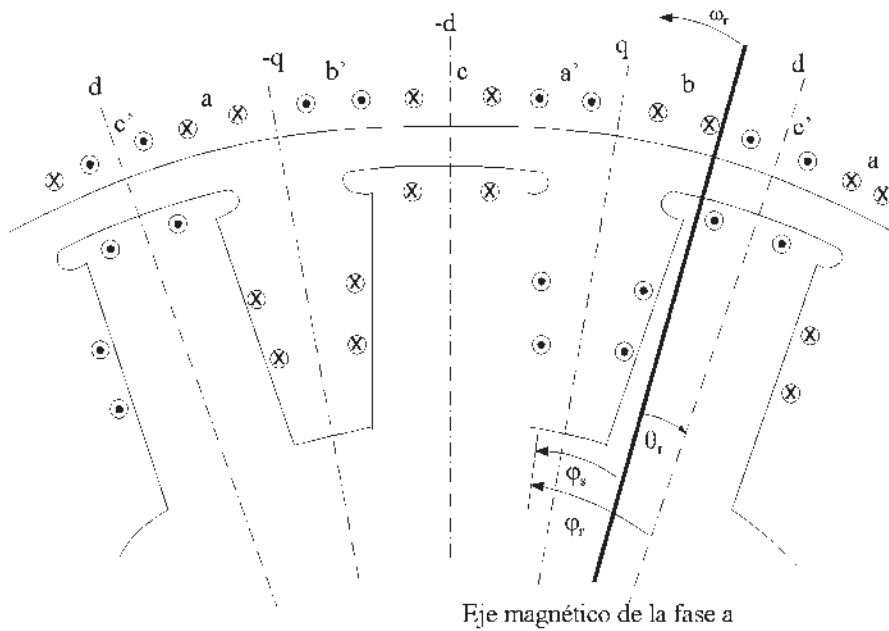


Figura 3.1: Diagrama esquemático de la máquina síncrona de p polos.

de campo y los devanados de amortiguamiento (**f,g,kd,kq**). Los devanados del estator son distribuidos y separados 120° eléctricos entre sí, con número de vueltas N_s y resistencia r_s . Como los devanados se consideran balanceados, el número de vueltas y la resistencia en cada uno de ellos es el mismo.

El rotor tiene el devanado de campo y tres devanados amortiguadores por cada par de polos, sobre el eje directo (eje **d**) se encuentra los devanados de campo (**f**) y uno de amortiguamiento (**kd**) con número de vueltas y resistencia igual a N_f, r_f y N_{kd}, r_{kd} , respectivamente. Los devanados (**g**) y (**kq**) con parámetros de N_g, r_g y N_{kq}, r_{kq} , respectivamente, se encuentran sobre el eje de cuadratura (eje **q**), el cual está 90° adelante del eje **d** y éste a su vez se encuentra formando un ángulo θ_r con respecto a la referencia física en el estator. El giro del rotor (con velocidad angular ω_r) es en sentido contrario al movimiento de las manecillas del reloj, tal y como se indica en la Figura 3.1.

El modelo del transitorio eléctrico de la máquina síncrona se describe por las

ecuaciones de voltaje para cada devanado. En forma compacta es

$$V = RI + \frac{d}{dt}\lambda \quad (3.6)$$

$$\lambda = LI \quad (3.7)$$

donde

$\mathbf{V} = [v_a \ v_b \ v_c \ v_f \ v_g \ v_{kd} \ v_{kq}]^T$ Vector de voltajes, contiene el voltaje en las fases, el campo y en los devanados de amortiguamiento, éstos últimos por estar cortocircuitados su voltaje es cero.

$\mathbf{I} = [-i_a \ -i_b \ -i_c \ i_f \ i_g \ i_{kd} \ i_{kq}]^T$ Vector de corrientes en el estator y rotor, nótese el signo de las corrientes en el estator, esto indica que se está operando como generador.

$\lambda = [\lambda_a \ \lambda_b \ \lambda_c \ \lambda_f \ \lambda_g \ \lambda_{kd} \ \lambda_{kq}]^T$ Vector de enlaces de flujo en el estator y en el rotor; se relaciona con el vector de corrientes de acuerdo a (3.7).

La matriz de resistencias R de la máquina síncrona es una matriz diagonal, donde se tienen las resistencias de los devanados de las fases, del campo y de los devanados de amortiguamiento. La matriz de inductancias \mathbf{L} de la máquina síncrona tiene la siguiente forma

$$\left[\begin{array}{ccc|ccc} L_{aa} & L_{ab} & L_{ac} & L_{af} & L_{ag} & L_{akd} & L_{akq} \\ L_{ba} & L_{bb} & L_{bc} & L_{bf} & L_{bg} & L_{bkd} & L_{bkq} \\ L_{ca} & L_{cb} & L_{cc} & L_{cf} & L_{cg} & L_{ckd} & L_{ckq} \\ \hline L_{fa} & L_{fb} & L_{fc} & L_{ff} & 0 & L_{fkd} & 0 \\ L_{ga} & L_{gb} & L_{gc} & 0 & L_{gg} & 0 & L_{gkq} \\ L_{kda} & L_{kdb} & L_{kdc} & L_{kdf} & 0 & L_{kdkd} & 0 \\ L_{kqa} & L_{kqb} & L_{kqc} & 0 & L_{kqg} & 0 & L_{kqkq} \end{array} \right] \quad (3.8)$$

Aquí \mathbf{L} depende únicamente de los parámetros primitivos de la máquina [Pérez04],

es decir, parámetros que solo posee el diseñador y fabricante. Por lo tanto los elementos de la matriz \mathbf{L} son [Pérez04, Krause87, Krause89]

Inductancias propias en el estator

$$L_{aa} = L_l + L_A + L_B \cos 2(\theta) \quad (3.9)$$

$$L_{bb} = L_l + L_A + L_B \cos 2(\theta - \frac{2}{3}\pi) \quad (3.10)$$

$$L_{cc} = L_l + L_A + L_B \cos 2(\theta + \frac{2}{3}\pi) \quad (3.11)$$

Inductancias mutuas en el estator

$$L_{ab} = L_{ba} = -\frac{1}{2}L_A + L_B \cos 2(\theta - \frac{\pi}{3}) \quad (3.12)$$

$$L_{bc} = L_{cb} = -\frac{1}{2}L_A + L_B \cos 2(\theta) \quad (3.13)$$

$$L_{ac} = L_{ca} = -\frac{1}{2}L_A + L_B \cos 2(\theta + \frac{\pi}{3}) \quad (3.14)$$

Inductancias propias del rotor

$$L_{ff}, L_{gg}, L_{kdkd} \quad y \quad L_{kqkq} \quad (3.15)$$

Inductancias mutuas del rotor

$$L_{fkd} = L_{kdf} \quad y \quad L_{gkq} = L_{kqg} \quad (3.16)$$

Inductancias estator-rotor

$$L_{af} = L_{fa} = L_{md} \cos(\theta) \quad (3.17)$$

$$L_{bf} = L_{fb} = L_{md} \cos(\theta - \frac{2}{3}\pi) \quad (3.18)$$

$$L_{cf} = L_{fc} = L_{md} \cos(\theta + \frac{2}{3}\pi) \quad (3.19)$$

$$L_{akd} = L_{kda} = L_{md} \cos(\theta) \quad (3.20)$$

$$L_{bkd} = L_{kdb} = L_{md} \cos(\theta - \frac{2}{3}\pi) \quad (3.21)$$

$$L_{ckd} = L_{kdc} = L_{md} \cos(\theta + \frac{2}{3}\pi) \quad (3.22)$$

$$L_{ag} = L_{ga} = -L_{mq} \sin(\theta) \quad (3.23)$$

$$L_{bg} = L_{gb} = -L_{mq} \sin\left(\theta - \frac{2}{3}\pi\right) \quad (3.24)$$

$$L_{cg} = L_{gc} = -L_{mq} \sin\left(\theta + \frac{2}{3}\pi\right) \quad (3.25)$$

$$L_{akq} = L_{kqa} = -L_{mq} \sin(\theta) \quad (3.26)$$

$$L_{bkq} = L_{kqb} = -L_{mq} \sin\left(\theta - \frac{2}{3}\pi\right) \quad (3.27)$$

$$L_{ckq} = L_{kqc} = -L_{mq} \sin\left(\theta + \frac{2}{3}\pi\right) \quad (3.28)$$

Los valores de L_A , L_B , L_{md} y L_{mq} son los parámetros estándar en **abc**, los cuales se obtienen a partir de los datos proporcionados por el fabricante en **dq0**, como sigue

$$L_A = \frac{1}{3}(L_d + L_q) - \frac{2}{3}L_l \quad (3.29)$$

$$L_B = \frac{1}{3}(L_d - L_q) \quad (3.30)$$

$$L_{md} = (L_d - L_l) \quad (3.31)$$

$$L_{mq} = (L_q - L_l) \quad (3.32)$$

$$L_{ff} = (L_{fl} + L_{md}) \quad (3.33)$$

$$L_{gg} = (L_{gl} + L_{mq}) \quad (3.34)$$

$$L_{kdkd} = (L_{kdl} + L_{md}) \quad (3.35)$$

$$L_{kqkq} = (L_{kql} + L_{mq}) \quad (3.36)$$

La expresión de par eléctrico, en función de corrientes y enlaces de flujo en coordenadas **abc**, está dada por

$$T_e = \frac{p}{3\sqrt{3}} \{ \lambda_a(i_b - i_c) + \lambda_b(i_c - i_a) + \lambda_c(i_a - i_b) \} \quad (3.37)$$

3.3.1. Ecuación de oscilación

El modelo que describe el comportamiento mecánico del rotor se representa por la suma de pares en la flecha del generador. La ecuación para este modelo se conoce como ecuación de oscilación.

$$\frac{d}{dt}\omega_r = \frac{\omega_b}{2H}(T_m - T_e) \quad (3.38)$$

$$\frac{d}{dt}\delta = (\omega_r - \omega_s) \quad (3.39)$$

donde, ω_r y ω_s , son la velocidad del rotor y velocidad síncrona en p.u., H es la constante de inercia en *segundos*, δ es el ángulo de carga en *radianes*, T_m y T_e son el par mecánico y eléctrico en p.u.

El par eléctrico se obtiene de (3.37). El modelo de la máquina síncrona es el conjunto de las ecuaciones (3.6) a (3.39), donde las inductancias dependen de las características magnéticas del material y de la posición del rotor.

3.3.2. Sistema de excitación. Control automático de voltaje

El control de voltaje en los generadores se lleva a cabo mediante los sistemas de excitación. Se utilizará un sistema de excitación típico compensado [Pérez04, Anderson94, Kundur94, Bergen86], el cual se ilustra en la Figura 3.2.

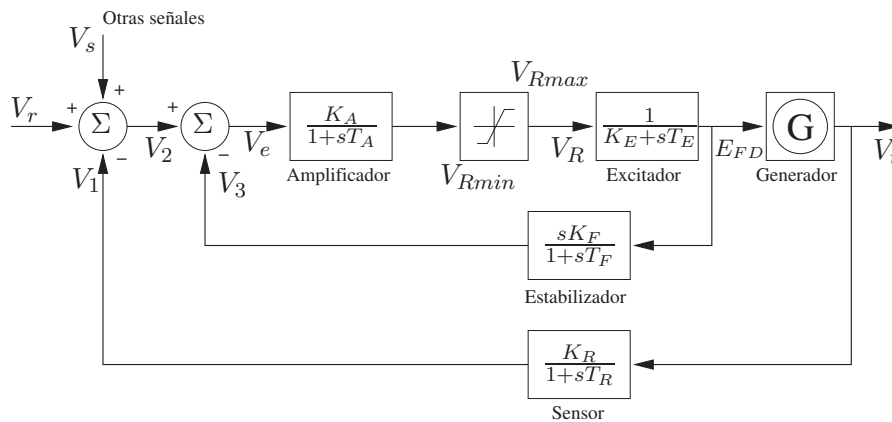


Figura 3.2: Sistema de excitación típico compensado.

Las ecuaciones del sistema de excitación en variables de estado es

$$\begin{bmatrix} \dot{V}_1 \\ \dot{V}_3 \\ \dot{V}_R \\ \dot{E}_{FD} \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_R} & 0 & 0 & 0 \\ 0 & -\frac{1}{T_F} & \frac{K_F}{T_R T_E} & -\frac{K_F K_E}{T_R T_E} \\ -\frac{K_A}{T_A} & -\frac{K_A}{T_A} & -\frac{1}{T_A} & 0 \\ 0 & 0 & \frac{1}{T_E} & -\frac{K_E}{T_E} \end{bmatrix} \begin{bmatrix} V_1 \\ V_3 \\ V_R \\ E_{FD} \end{bmatrix} + \begin{bmatrix} \frac{V_t K_R}{T_R} \\ 0 \\ \frac{(V_s + V_r) K_A}{T_A} \\ 0 \end{bmatrix} \quad (3.40)$$

3.3.3. Primo-motores. Control automático de generación

La finalidad del control automático de generación es mantener la potencia real de salida de una unidad generadora, así como controlar la frecuencia de un sistema interconectado para mantenerla idealmente constante.

La frecuencia de un sistema depende del balance de potencia real en el mismo, por lo tanto, un cambio en la potencia real inducirá un cambio en la frecuencia, por lo que el control de generación debe compensar este cambio de potencia real desde o hacia el primo-motor y debe regresar en el menor tiempo posible a la frecuencia nominal. El modelo simplificado para el control de frecuencia-carga en un turbogenerador y en un hidrogenerador [Pérez04, Saadat99], son ilustrados en las Figuras 3.3 y 3.4, respectivamente.

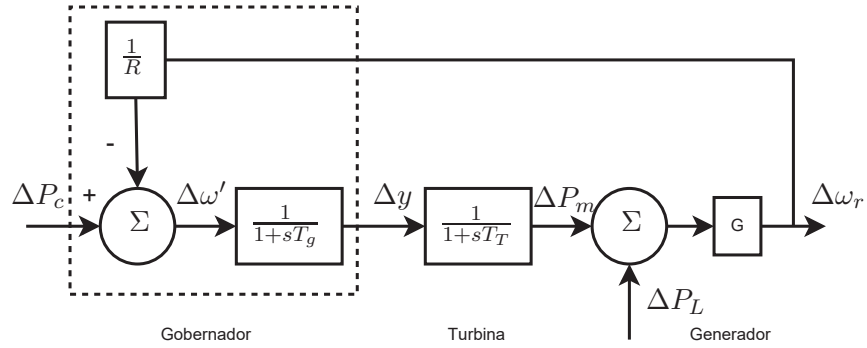


Figura 3.3: Control de carga de un turbogenerador.

La representación en variables de estado del control de carga para el turbogenerador es

$$\begin{bmatrix} \Delta \dot{y} \\ \Delta \dot{P}_m \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_g} & 0 \\ \frac{1}{T_T} & -\frac{1}{T_T} \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta P_m \end{bmatrix} + \begin{bmatrix} \frac{\Delta \omega'}{T_g} \\ 0 \end{bmatrix} \quad (3.41)$$

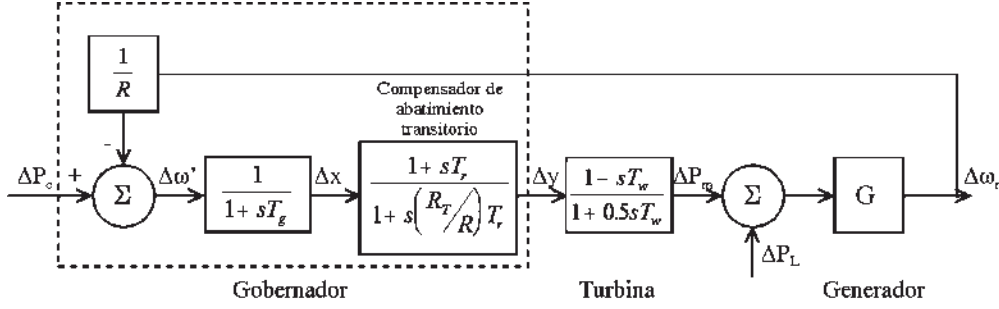


Figura 3.4: Control de carga de un hidrogenerador.

y para el hidrogenerador es

$$\begin{bmatrix} \Delta \dot{x} \\ \Delta \dot{y} \\ \Delta \dot{P}_m \end{bmatrix} = \begin{bmatrix} -\frac{1}{T_g} & 0 & 0 \\ \frac{R}{R_T T_r} (1 - \frac{T_r}{T_g}) & -\frac{R}{R_T T_r} & 0 \\ -\frac{2R}{R_T T_r} (1 - \frac{T_r}{T_g}) & \frac{2}{T_w} (1 + \frac{T_w R}{R_T T_r}) & -\frac{2}{T_w} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta P_m \end{bmatrix} + \begin{bmatrix} \frac{1}{T_g} \\ \frac{R}{R_T T_g} \\ \frac{2R}{R_T T_g} \end{bmatrix} \Delta \omega' \quad (3.42)$$

3.4. Modelo del transformador trifásico de cinco columnas

Se implementará el modelo del transformador trifásico de cinco columnas [García00a, Pérez04] en función de sus propiedades primitivas, es decir, por medio de reluctancias, lo cual permite tener un control total sobre él, debido a que se puede considerar un número de n devanados por fase, así como las propiedades magnéticas del material. Para este trabajo, la Figura 3.5 muestra el transformador trifásico de cinco columnas el cual se toma como base para el desarrollo del modelo en función de las reluctancias magnéticas del mismo, la reducción al trifásico de tres columnas resulta directo como se verá del modelo resultante. En la Figura 3.6 se observa el circuito eléctrico equivalente del transformador, el objetivo es obtener \mathbf{L} y/o su inversa $\mathbf{L}^{-1} = \mathbf{\Gamma}$, donde este parámetro es el que va a estar cambiando para cada paso de integración de acuerdo a la no linealidad del material magnético del troquel del transformador. También en esta misma Figura se indica la ventaja que se tiene con este modelo, ya que en términos generales se modela la matriz de inductancias multipuertos no lineal $\mathbf{\Gamma}$ del transformador, dejando libres sus terminales para que la alimentación pueda ser derivada de cualquier conexión trifásica que se requiera del transformador, así como poder agregar otros dispositivos en las terminales del mismo. Además, se observa que se

pueden manejar n devanados en el primario y secundario por fase, lo cual permite modelar directamente tap's al conectarlos en serie de acuerdo a la Figura 3.6.

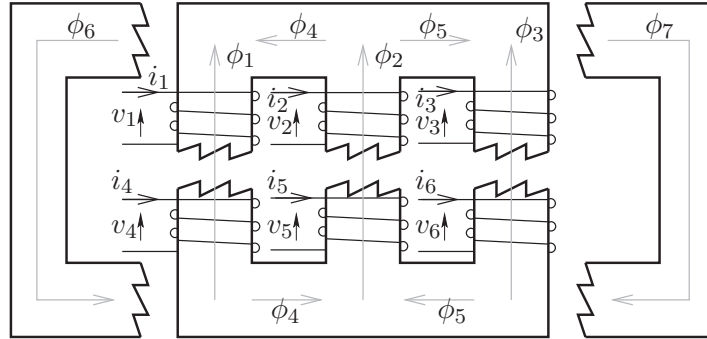


Figura 3.5: Transformador trifásico de cinco columnas.

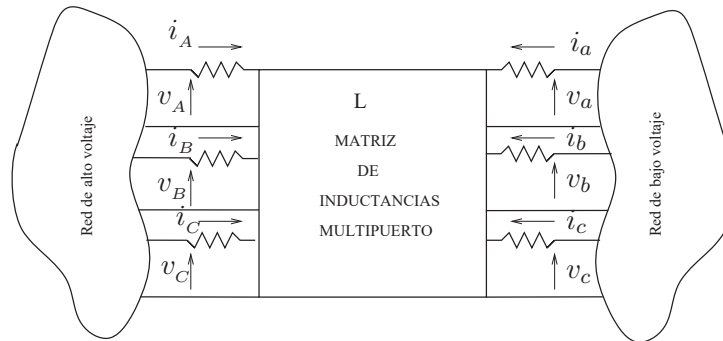


Figura 3.6: Circuito equivalente eléctrico del transformador trifásico de cinco columnas.

Se observa en la Figura 3.6 que la matriz de inductancias multipuerto es la que depende de las características físicas del transformador, con lo cual se puede manipular la exactitud que se desea para el modelo y si se desea de tres o cinco columnas, o más aún, si tiene diferente tipo de material magnético en las diferentes regiones del troquel del transformador.

Cuando se modela el transformador lineal no hay mucha diferencia entre elegir como variables de estado las corrientes o los enlaces de flujo. Sin embargo, en el modelo no lineal resulta más conveniente elegir como variables de estado los enlaces de flujo, ya que

estos cambian más lento que las corrientes que los producen, lo que conlleva en sí una mayor estabilidad computacional del modelo.

Eligiendo los enlaces de flujo como variables de estado, la ecuación de voltajes para los devanados del transformador es

$$\frac{d\lambda}{dt} = v - Ri \quad (3.43)$$

donde λ , v e i son vectores de 6×1 y corresponden a los enlaces de flujo, voltajes y corrientes respectivamente en cada uno de los devanados del transformador, R representa la resistencia que tienen los devanados, siendo ésta una matriz diagonal de 6×6 . La relación que existe entre las corrientes y los enlaces de flujo es

$$i = L^{-1}\lambda = \Gamma\lambda \quad (3.44)$$

El sistema de ecuaciones diferencio-algebraicas (3.43) y (3.44) forman el modelo del transformador trifásico. Se observa que el modelo depende del cálculo de Γ , por lo que se desarrollará su cálculo a continuación.

Como \mathbf{L} , y por consecuencia $\mathbf{\Gamma}$ dependen exclusivamente de la geometría magnética del transformador, su obtención se basa directamente de la Figura 3.7. Aquí, las reluctancias de las columnas donde se colocan los devanados primario y secundario son \mathfrak{R}_1 , \mathfrak{R}_2 y \mathfrak{R}_3 . Las reluctancias en los yugos son \mathfrak{R}_4 y \mathfrak{R}_5 , aclarando que \mathfrak{R}_4 es la reluctancia neta de los yugos inferior y superior de la parte izquierda de la Figura 3.7, similarmente \mathfrak{R}_5 en la parte derecha de la Figura 3.7. Las reluctancias \mathfrak{R}_6 y \mathfrak{R}_7 corresponden a las columnas externas. De este modo, las reluctancias \mathfrak{R}_1 a \mathfrak{R}_7 son no lineales y cambian de acuerdo a la característica magnética del material, normalmente esta característica es la misma para los siete segmentos, ya que el núcleo del transformador está fabricado del mismo material.

Debido a la dificultad para medir o calcular las reluctancias de columnas y yugos que siguen una trayectoria por el aire, éstas se incluyen en los segmentos del núcleo adecuados y sus valores son proporcionales a los segmentos que relacione. Por lo tanto, \mathfrak{R}_{l1} , \mathfrak{R}_{l2} y \mathfrak{R}_{l3} son las reluctancias de dispersión en el aire que está entre los devanados primario y secundario, \mathfrak{R}_8 , \mathfrak{R}_9 y \mathfrak{R}_{10} , son las reluctancias de dispersión a través del aire y el tanque. Éstas últimas no son tan importantes en el transformador de cinco columnas ya que se iría la

en forma compacta la matriz de coeficientes queda

$$\left[\begin{array}{c|c} A & B \\ \hline B^T & C \end{array} \right]$$

donde

$$\begin{aligned} \mathfrak{R}_{11} &= \mathfrak{R}_1 + \mathfrak{R}_{l1} & \mathfrak{R}_{14} &= -\mathfrak{R}_{l1} \\ \mathfrak{R}_{22} &= \mathfrak{R}_2 + \mathfrak{R}_{l2} & \mathfrak{R}_{25} &= -\mathfrak{R}_{l2} \\ \mathfrak{R}_{33} &= \mathfrak{R}_3 + \mathfrak{R}_{l3} & \mathfrak{R}_{36} &= -\mathfrak{R}_{l3} \\ \mathfrak{R}_{44} &= \mathfrak{R}_4 + \mathfrak{R}_9 + \mathfrak{R}_{l1} & \mathfrak{R}_{47} &= \mathfrak{R}_4 + \mathfrak{R}_9 \\ \mathfrak{R}_{48} &= \mathfrak{R}_4 & \mathfrak{R}_{55} &= \mathfrak{R}_5 + \mathfrak{R}_7 + \mathfrak{R}_{l2} \\ \mathfrak{R}_{56} &= \mathfrak{R}_7 & \mathfrak{R}_{58} &= \mathfrak{R}_5 + \mathfrak{R}_7 \\ \mathfrak{R}_{510} &= \mathfrak{R}_7 & \mathfrak{R}_{66} &= \mathfrak{R}_7 + \mathfrak{R}_{l3} \\ \mathfrak{R}_{68} &= \mathfrak{R}_7 & \mathfrak{R}_{610} &= \mathfrak{R}_7 \\ \mathfrak{R}_{77} &= \mathfrak{R}_4 + \mathfrak{R}_6 + \mathfrak{R}_9 & \mathfrak{R}_{78} &= \mathfrak{R}_4 + \mathfrak{R}_6 \\ \mathfrak{R}_{79} &= \mathfrak{R}_6 & \mathfrak{R}_{88} &= \mathfrak{R}_4 + \mathfrak{R}_5 + \mathfrak{R}_6 + \mathfrak{R}_7 \\ \mathfrak{R}_{89} &= \mathfrak{R}_6 & \mathfrak{R}_{810} &= \mathfrak{R}_7 \\ \mathfrak{R}_{99} &= \mathfrak{R}_6 + \mathfrak{R}_8 & \mathfrak{R}_{1010} &= \mathfrak{R}_7 + \mathfrak{R}_{l0} \end{aligned} \tag{3.46}$$

ya que la matriz es simétrica, se cumple que

$$\mathfrak{R}_{ij} = \mathfrak{R}_{ji} \tag{3.47}$$

donde el cálculo de cada reluctancia se realiza mediante

$$\mathfrak{R}_k = \frac{l_k}{\mu_k A_k} \tag{3.48}$$

donde l_k , A_k y μ_k son la longitud, área de la sección transversal y permeabilidad del k -ésimo segmento del núcleo.

Debido a que un solo flujo de malla pasa por cada una de las fuentes magnetomotrices, las fmm 's aplicadas a las mallas de 7 a 10 son cero, por lo tanto, aplicando la reducción de Kron a (3.45) como sigue

$$A^m = A - BC^{-1}B^T \tag{3.49}$$

se reducirá a

$$\overbrace{\begin{bmatrix} \mathfrak{R}_{11}^m & & & & \mathfrak{R}_{14}^m & & & & \\ & \mathfrak{R}_{22}^m & & & & \mathfrak{R}_{25}^m & & & \\ & & \mathfrak{R}_{33}^m & & & & \mathfrak{R}_{36}^m & & \\ \mathfrak{R}_{41}^m & & & \mathfrak{R}_{44}^m & \mathfrak{R}_{45}^m & \mathfrak{R}_{46}^m & & & \\ & \mathfrak{R}_{52}^m & & \mathfrak{R}_{54}^m & \mathfrak{R}_{55}^m & \mathfrak{R}_{56}^m & & & \\ & & \mathfrak{R}_{63}^m & \mathfrak{R}_{64}^m & \mathfrak{R}_{65}^m & \mathfrak{R}_{66}^m & & & \end{bmatrix}}^{A^m} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} \quad (3.50)$$

Se puede demostrar fácilmente que (3.50) es igual a

$$\begin{bmatrix} \frac{\mathfrak{R}_{11}^m}{N_1^2} & & & & \frac{\mathfrak{R}_{14}^m}{N_1 N_4} & & & & \\ & \frac{\mathfrak{R}_{22}^m}{N_2^2} & & & & \frac{\mathfrak{R}_{25}^m}{N_2 N_5} & & & \\ & & \frac{\mathfrak{R}_{33}^m}{N_3^2} & & & & \frac{\mathfrak{R}_{36}^m}{N_3 N_6} & & \\ \frac{\mathfrak{R}_{41}^m}{N_4 N_1} & & & \frac{\mathfrak{R}_{44}^m}{N_4^2} & \frac{\mathfrak{R}_{45}^m}{N_4 N_5} & \frac{\mathfrak{R}_{46}^m}{N_4 N_6} & & & \\ & \frac{\mathfrak{R}_{52}^m}{N_5 N_2} & & \frac{\mathfrak{R}_{54}^m}{N_5 N_4} & \frac{\mathfrak{R}_{55}^m}{N_5^2} & \frac{\mathfrak{R}_{56}^m}{N_5 N_6} & & & \\ & & \frac{\mathfrak{R}_{63}^m}{N_6 N_3} & \frac{\mathfrak{R}_{64}^m}{N_6 N_4} & \frac{\mathfrak{R}_{65}^m}{N_6 N_5} & \frac{\mathfrak{R}_{66}^m}{N_6^2} & & & \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ \lambda_5 \\ \lambda_6 \end{bmatrix} = \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \end{bmatrix} \quad (3.51)$$

donde

$$N_k \phi_k = \lambda_k \quad (3.52)$$

$$N_k i_k = f_k$$

La actualización de las permeabilidades μ_k , $k = 1 \dots 7$, consiste en obtener la H en cada uno de los segmentos del núcleo del transformador, para lo cual primero se calculan las fmm 's nodales fn_k , $k = 1 \dots 3$, como sigue

$$fn_k = f_k + f_{k+3} + \mathfrak{R}_k \frac{\lambda_k}{N_k} \quad k = 1 \dots 3 \quad (3.53)$$

esto habilita el cálculo de H para cada segmento, por lo que para las columnas con devanado se tiene

$$H_k = \frac{f_k + f_{k+3} - fn_k}{l_k} \quad (3.54)$$

para los yugos se tiene

$$H_{k+3} = \frac{fn_k - fn_{k+1}}{l_{k+3}} \quad k = 1 \dots 2 \quad (3.55)$$

y finalmente para las columnas externas tenemos

$$H_6 = \frac{fn_1}{l_6} \quad (3.56)$$

$$H_7 = \frac{fn_3}{l_7} \quad (3.57)$$

Con la H en cada segmento se obtiene la B para el mismo de la curva de saturación y se actualiza la permeabilidad como sigue

$$\mu_k = \frac{B_k}{H_k} \quad k = 1 \dots 7 \quad (3.58)$$

Si se trabaja en *por unidad (p.u.)*, de (3.50) y (3.51) se induce que bajo esta referencia $\phi_k = \lambda_k$, $fmm_k = i_k$, para $k = 1 \dots 7$ y $A^m = \Gamma$, la cual es una matriz simétrica y dispersa. Por lo tanto, la formación directa de Γ implica únicamente un manejo de la característica no lineal de saturación B - H en *p.u.*

3.4.1. Característica de saturación

La aproximación de la característica de saturación se realiza mediante la función arcotangente [Pérez04, Pérez99], puesto que dicha función por sí sola tiene la forma de la saturación, por lo tanto lo único que se tiene que hacer es normalizarla y obtener el mínimo número de parámetros para tener control sobre ella. La función propuesta para la aproximación de la curva mostrada en la Figura 3.8, está dada por [Pérez99].

$$\lambda_m(i_m) = \lambda_n \operatorname{atan}(mi_m) + \Delta\lambda i_m \quad (3.59)$$

Se observa que (3.59) depende de tres constantes que se calculan a partir de los datos que se tienen en la curva real, como sigue

$$\lambda_n = \lambda_x \frac{2}{\pi} \quad (3.60)$$

$$m = \frac{\Delta\lambda}{\Delta i} \frac{\pi}{2\lambda_s} \quad (3.61)$$

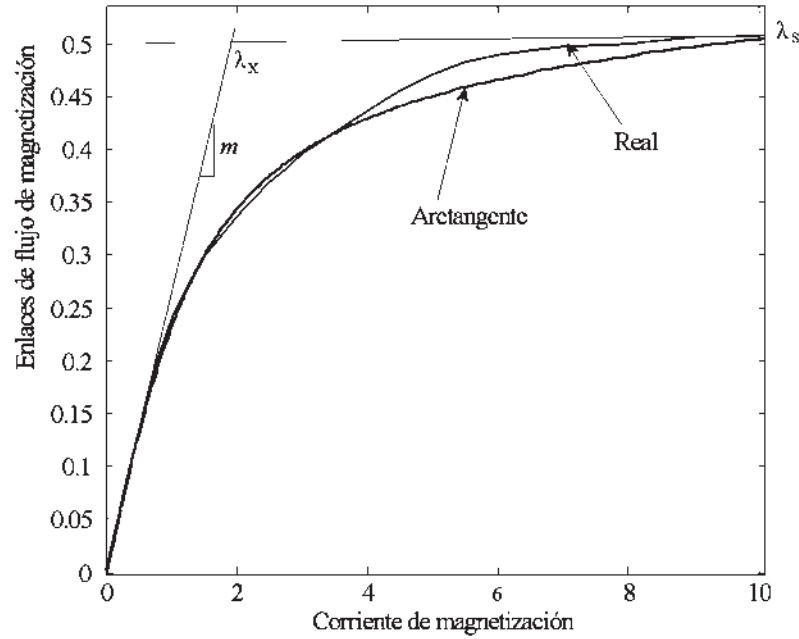


Figura 3.8: Curva de saturación real y aproximada.

$$\Delta\lambda = \frac{[\lambda_s - \lambda_n \operatorname{atan}(mi_s)]}{i_s} \quad (3.62)$$

donde

- λ_x es el valor de λ_m en el punto de cruce de las pendientes indicadas en la Figura 3.8.
- λ_n es el valor de λ_x normalizada al valor máximo que tiende la función arcotangente, $\pi/2$.
- λ_s es el valor de saturación de λ_m de la característica real.
- m es la pendiente inicial de la curva, normalizada al valor máximo del arcotangente, $\pi/2$, y al valor de saturación λ_s .
- i_s es el valor de i para cuando se tiene el punto λ_s .
- $\Delta\lambda$ es el incremento lineal de λ_m .
- λ_m enlaces de flujo de magnetización.
- i_m corriente de magnetización.

Se observa que el cálculo de las constantes (3.60)-(3.62) es simple y sus parámetros son directos de obtener a partir de la característica real. Estas constantes nos permiten tener

un control total en (3.59) e incluso hacer un ajuste más fino. La curva del valor real mostrada en la Figura 3.8 corresponde a la ilustrada en [Krause89]. Los valores de las constantes para obtener la curva de saturación aproximada son [Pérez00]; $\lambda_n = 0.3215$, $m = 0.8642$ y $\Delta\lambda = 0.0037$.

3.4.2. Matriz de conectividad delta-estrella aterrizada

La matriz de conectividad representa la configuración física del transformador. La ecuación que representa la relación entre las corrientes de línea y las corrientes de fase para una conexión delta-estrella aterrizada según la Figura 3.9 es la siguiente

$$\begin{bmatrix} i_A \\ i_B \\ i_C \\ i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \end{bmatrix} \quad (3.63)$$

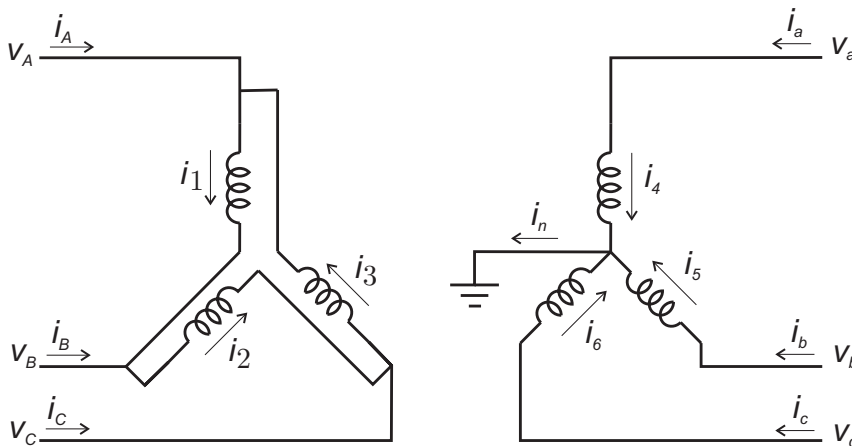


Figura 3.9: Conexión delta-estrella aterrizada.

3.5. Modelo de líneas

El modelo de las líneas de transmisión usado es el de un circuito **RLC** [Pérez04, Saadat99, Bergen86], la cual es una de las representaciones más simple de la línea de transmisión, este equivalente se encuentra en términos de parámetros concentrados por fase, así como sus respectivos circuitos mutuos. La Figura 3.10 muestra el circuito equivalente trifásico.

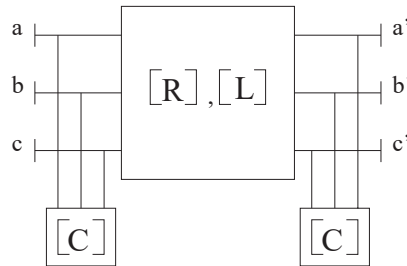


Figura 3.10: Modelo de la línea trifásica con retorno por tierra.

El circuito equivalente con parámetros concentrados de la impedancia serie de la línea de transmisión es el indicado en la Figura 3.11.

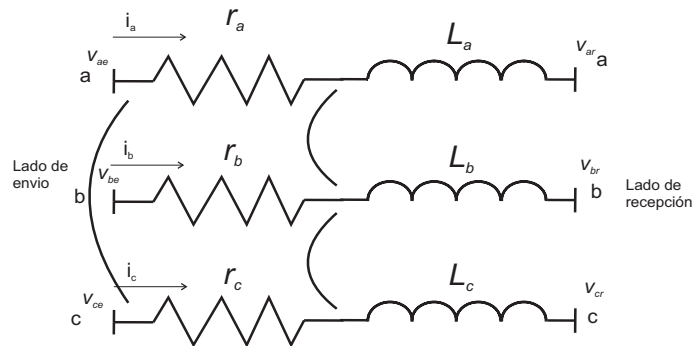


Figura 3.11: Circuito equivalente con parámetros concentrados de la línea de transmisión.

El sistema de ecuaciones de los elementos serie de la línea de transmisión es

$$\begin{bmatrix} v_{ae} - v_{ar} \\ v_{be} - v_{br} \\ v_{ce} - v_{cr} \end{bmatrix} = \begin{bmatrix} L_{aa'} & L_{ab'} & L_{ac'} \\ L_{ba'} & L_{bb'} & L_{bc'} \\ L_{ca'} & L_{cb'} & L_{cc'} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} + \begin{bmatrix} r_{aa'} & r_{ab'} & r_{ac'} \\ r_{ba'} & r_{bb'} & r_{bc'} \\ r_{ca'} & r_{cb'} & r_{cc'} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \quad (3.64)$$

3.6. Modelo de capacitores

Los efectos capacitivos se consideran como bancos de capacitores y su conexión se muestra en la Figura 3.12. La ecuación matricial de corriente para el capacitor es

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} = \begin{bmatrix} C_{aa} & C_{ab} & C_{ac} \\ C_{ba} & C_{bb} & C_{bc} \\ C_{ca} & C_{cb} & C_{cc} \end{bmatrix} \frac{d}{dt} \begin{bmatrix} v_a \\ v_b \\ v_c \end{bmatrix} \quad (3.65)$$

donde $i_a, i_b, i_c, v_a, v_b, v_c$ son las corrientes y voltajes en las fases abc.

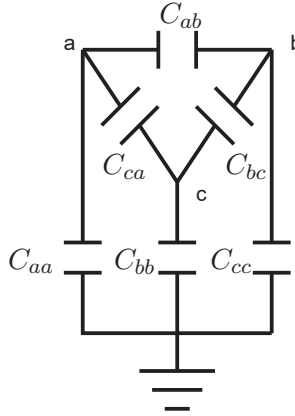


Figura 3.12: Circuito equivalente de un banco de capacitores trifásico.

3.7. Resumen

En este Capítulo se presentaron los modelos de los principales elementos que conforman a los sistemas eléctricos de potencia, todos estos en sus coordenadas **abc**.

En el modelo de la máquina síncrona se presentaron las ecuaciones que describen su comportamiento dinámico. Este modelo de la máquina síncrona es posible aplicarlo tanto a hidrogeneradores como a turbogeneradores, con el simple hecho de hacer las modificaciones necesarias al modelo general planteado. También fueron presentados sus controles de velocidad y excitación.

Se presentó el modelo del transformador trifásico de cinco columnas en base a su circuito magnético sin transformaciones de dualidad. De acuerdo con el concepto primitivo del modelo del transformador es fácil insertar cambios para obtener el modelo del transformador trifásico de tres y cuatro columnas, así como transformadores trifásicos de más devanados y las diferentes conexiones de los mismos.

El modelo para la línea de transmisión presentado es por medio de la representación más simple que hay, un circuito RLC, donde el efecto capacitivo de la línea se modelan como bancos de capacitores.

También se mostraron los métodos de mayor aplicación en la solución numérica de los modelos matemáticos que conforman a los elementos de los sistemas eléctricos de potencia, esto debido a las amplias características de estabilidad, exactitud y facilidad en su implementación computacional. Estos métodos presentados fueron, la Regla Trapezoidal (implícito) y el Runge-Kutta de Cuarto Orden (explícito).

Capítulo 4

Análisis de estabilidad angular transitoria considerando aceleración al estado estacionario

4.1. Introducción

Del análisis numérico de sistemas no lineales con entradas periódicas y una respuesta periódica estable, se puede obtener la respuesta periódica en estado estacionario para un estado inicial dado mediante la aplicación de un método de integración al sistema de ecuaciones hasta que la respuesta se haga periódica. En sistemas pobremente amortiguados este proceso de integración se extiende sobre muchos periodos de tiempo haciéndolo costoso computacionalmente, por lo tanto es factible utilizar técnicas de aceleración al estado estacionario tales como las desarrolladas por Aprille y Trick [Aprille72, Li00]. Es necesario indicar que la propuesta de esta Tesis para el proceso de aceleración al estado estacionario está basada en el trabajo de Aprille y Trick [Aprille72].

Esta técnica se basa en el método de Newton a un mapeo de la ecuación a solucionar. Si existe una órbita estable, el algoritmo de Newton converge a ella rápidamente siempre y cuando las condiciones iniciales se encuentren en la vecindad de esta órbita.

Este Capítulo presenta la simulación y resultados del análisis de estabilidad angular

transitoria para el Generador y el sistema Generador-Transformador-Línea-Barra Infinita, considerando la aceleración numérica al estado estacionario.

4.2. Aceleración de la solución al estado estable

Partiendo de que los SEP's están conformados por sistemas de EDA's de la forma

$$y' = f(t, y, z) \tag{4.1}$$

$$0 = g(t, y, z) \tag{4.2}$$

donde y y f son vectores de dimensión n , y z son las variables algebraicas, además f es periódica en t de periodo T , es continua en t, y , y su primer derivada parcial respecto a y es continua para toda y y $-\infty < t < \infty$. Por lo tanto se tendrá una solución periódica de (4.1) y (4.2) en el periodo T .

El objetivo es determinar la solución periódica de (4.1) y (4.2) sobre el intervalo $[0, T]$ partiendo de una condición inicial, además de satisfacer la condición frontera

$$y(0) = y(T) \tag{4.3}$$

integrando (4.1) se tiene

$$y(T) = \int_0^T f(y, \tau) d\tau + y(0) \tag{4.4}$$

bajo las restricciones algebraicas (4.2), se puede expresar (4.1) en términos del mapeo

$$y_0 = E(y_0) \tag{4.5}$$

donde

$$y_0 = y(0) \quad E(y_0) = \int_0^T f(y, \tau) d\tau + y_0 \tag{4.6}$$

por lo tanto $y(t)$ satisface (4.1) para $0 \leq t \leq T$.

La solución del estado estable se obtiene aplicando el método de Newton a (4.5)

$$y_0^{l+1} = y_0^l + [I - E'(y_0^l)]^{-1} [E(y_0^l) - y_0^l] \tag{4.7}$$

donde

$$E'(y_0^l) = \Phi(T, 0; y_0^l) \quad (4.8)$$

donde $\Phi(T, 0; y_0^l)$ es la matriz de transición de estados, ésta puede ser aproximada numéricamente mediante la diferenciación numérica [Chua75], o mediante los métodos que a continuación se describen.

Partiendo de que $z^i(t)$ es la solución de la ecuación diferencial [Aprille72]

$$\dot{z}^i = F(t; y^i)z^i \quad (4.9)$$

La solución a (4.9) es

$$z^i(t) = \Phi(t, 0; y_0^i)z^i(0) \quad (4.10)$$

Por lo tanto si aplicamos la fórmula de Euler Hacia Atrás (EHA) a (4.9), obtenemos

$$\hat{z}(h) = \hat{z}(0) + hF(\hat{y}(h))\hat{z}(h) \quad (4.11)$$

$$\hat{z}(h) = [I - hF(\hat{y}(h))]^{-1}\hat{z}(0) \quad (4.12)$$

además

$$z(T) \approx \hat{z}(kh) = \prod_{i=1}^k [I - hF(\hat{y}((k-i+1)h))]^{-1}z(0) \quad (4.13)$$

por comparación

$$\Phi(T, 0; y_0) \approx \prod_{i=1}^k [I - hF(\hat{y}((k-i+1)h))]^{-1} \quad (4.14)$$

Similarmente, aplicando la fórmula de Euler Hacia aDelante (EHD) a (4.9), obtenemos

$$\hat{z}(h) = \hat{z}(0) + hF(\hat{y}(0))\hat{z}(0) \quad (4.15)$$

$$\hat{z}(h) = [I + hF(\hat{y}(0))]\hat{z}(0) \quad (4.16)$$

además

$$z(T) \approx \hat{z}(kh) = \prod_{i=1}^k [I + hF(\hat{y}(ih))]z(0) \quad (4.17)$$

por lo tanto, comparando

$$\Phi(T, 0; y_0) \approx \prod_{i=1}^k [I + hF(\hat{y}(ih))] \quad (4.18)$$

también, aplicando la Regla Trapezoidal (RT) a (4.9), obtenemos

$$\hat{z}(h) = \hat{z}(0) + h/2[F(\hat{y}(h))\hat{z}(h) + F(\hat{y}(0))\hat{z}(0)] \quad (4.19)$$

$$\hat{z}(h) = [I - h/2F(\hat{y}(h))]^{-1}[I + h/2F(\hat{y}(0))]\hat{z}(0) \quad (4.20)$$

además

$$z(T) \approx \hat{z}(kh) = \prod_{i=1}^k ([I - h/2F(\hat{y}((k-i+1)h))]^{-1}[I + hF(\hat{y}(ih))])z(0) \quad (4.21)$$

comparando

$$\Phi(T, 0; y_0) \approx \prod_{i=1}^k ([I - h/2F(\hat{y}((k-i+1)h))]^{-1}[I + hF(\hat{y}(ih))]) \quad (4.22)$$

donde $F(y)$ es el Jacobiano y $h = T/k$ es el tamaño del paso para los tres casos, por lo tanto (4.14), (4.18) y (4.22) pueden resolverse mediante la diferenciación automática.

El parámetro k indica el número de puntos en que se evalúa el sistema durante un periodo de tiempo. Este parámetro es importante en el cálculo de Φ , ya que como se verá más adelante se observa que no se requieren de tantos puntos para su evaluación, con lo cual se obtiene más rápida Φ , esto debido a que se requieren de menos productos. Para todos los casos de estudio la elección de este parámetro fue según el que mejor resultados presentó.

Esta técnica numérica se aplica al sistema a resolver en periodos T y el criterio de convergencia del método de Newton se realiza en base a una tolerancia proporcionada por el usuario, en este trabajo la tolerancia utilizada en todos los casos es de $tol = 10^{-10}$. La secuencia por pasos para la solución del sistema de EDA's con aceleración al estado estable se muestra en la Figura 4.1. Primeramente se parte de condiciones iniciales consistentes para el SEP definido por una EDA, se resuelve este sistema de EDA's mediante los métodos clásicos tales como el Runge-Kutta de cuarto orden y/o la Regla Trapezoidal durante un tiempo de nT periodos, por lo que se puede decidir cuánto tiempo se desea observar el

transitorio. En este proyecto de investigación se resolvieron todos los sistemas por medio de estos dos métodos utilizando un paso de integración de $h \approx 3.2552 \times 10^{-05}$, todos los resultados mostrados en este trabajo son por medio del Runge-Kutta de cuarto orden. A continuación se aplica la técnica de aceleración para obtener el estado estable del sistema, donde las condiciones finales del estado transitorio pasan a ser las condiciones iniciales del proceso de aceleración.

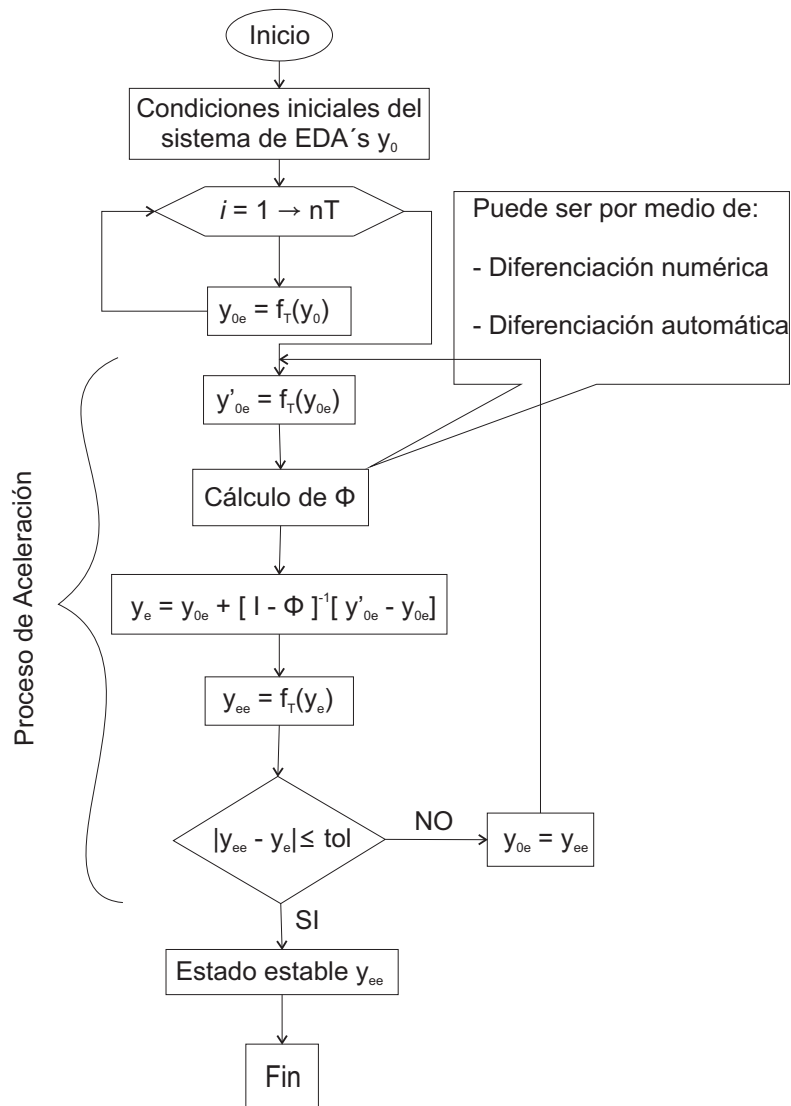


Figura 4.1: Diagrama de flujo, aceleración al estado estable.

Este paso consiste en calcular un ciclo base [Chua75] y obtener una solución previa y'_{oe} , posteriormente se calcula la matriz de transición de estados Φ por medio de cualesquiera de los métodos ya antes mencionados (DN, EHA, EHD, RT), para así poder aplicar el método de Newton para aproximar la solución al estado estable. Con esto último se evalúa otro ciclo y el resultado se compara con la tolerancia proporcionada por el usuario, si se cumple, entonces $y_e(n) = y_e(nT)$ con lo que se tendrá el estado estable, de lo contrario se tiene que realizar el proceso nuevamente tomando como condición inicial y_{ee} , ver Figura 4.1. Obsérvese que la función $f_T(y)$ indica la evaluación de un ciclo del sistema.

4.3. Comparación de la matriz de transición de estados

Se resolverá el ejemplo 5.12 ilustrado en [Rugh93] con el objetivo de comparar la matriz de transición de estados Φ obtenida por medio de la DN, DA y mediante la solución analítica.

El sistema a resolver es

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ -\cos t & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4.23)$$

Por lo tanto de [Rugh93], Φ en forma analítica está dada por

$$\Phi(t, 0) = \begin{bmatrix} e^{-t} & 0 \\ -1/2 + 1/2e^{-t}(\cos t - \sin t) & 1 \end{bmatrix} \quad (4.24)$$

Para $t = 2\pi$ se obtiene Φ por medio de DN, DA y de forma analítica tal como se muestra en la Tabla 4.1

Se observa que para el elemento (1,1), Φ es igual en los primeros diez dígitos para los tres casos, siendo igual en los primeros 11 dígitos para la solución analítica y DN. Para el elemento (2,1) de Φ , éstos son iguales en los primeros 8 dígitos en los tres casos, mientras que para la solución analítica y DN son iguales en los primeros 9 dígitos. Para los otros elementos de Φ son los mismos para los tres casos.

Tabla 4.1: Comparación de Φ .

Solución analítica	DN
$\begin{bmatrix} 0.00186744273170799 & 0.0 \\ -0.499066278634146 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 0.0018674427339484 & 0.0 \\ -0.499066278597952 & 1.0 \end{bmatrix}$
RT	
$\begin{bmatrix} 0.00186744272271352 & 0.0 \\ -0.499066279401455 & 1.0 \end{bmatrix}$	

4.4. Estudio de energización del transformador considerando aceleración al estado estacionario

Se analizará el transformador trifásico de cinco columnas considerando aceleración numérica al estado estable. El modelo matemático es el presentado en la Sección 3.4. Los parámetros del transformador se muestran en el Apéndice A, los cuales fueron tomados de [Pérez04], el tipo de conexión del transformador es Delta-Estrella Aterrizada. La permeabilidad es calculada de la característica de saturación representada por la función arcotangente como sigue

$$B(H) = 0.7 \tan^{-1}(754H) + 0.95H \quad (4.25)$$

El valor de las constantes en (4.25) se obtuvieron de [Pérez04], por lo tanto conociendo H , (3.54) a (3.57), se puede obtener B y así de esta forma se conoce la permeabilidad en cada segmento del núcleo del transformador por medio de (3.58).

La simulación se realizará de la siguiente forma; se energizará el transformador considerando a R_L como la resistencia de carga ($R_L = 20$ p.u. casi en vacío, $R_L = 1$ p.u. plena carga), después de 0.1 segundos (6 ciclos) se aplica la aceleración numérica para obtener el estado estable, analizándolo durante 0.05 segundos (3 ciclos). La Figura 4.2 muestra las corrientes en el lado primario, cuando el transformador se encuentra a plena carga, es decir, con $R_L = 1$ p.u. La Figura 4.3, también presenta las corrientes en los devanados primario, pero esta vez con el transformador energizado con muy poca carga ($R_L = 20$ p.u.).

Se puede observar en las Figuras 4.2 y 4.3 el comportamiento transitorio que presenta el transformador al ser energizado, ya sea a plena carga o en vacío, observándose los

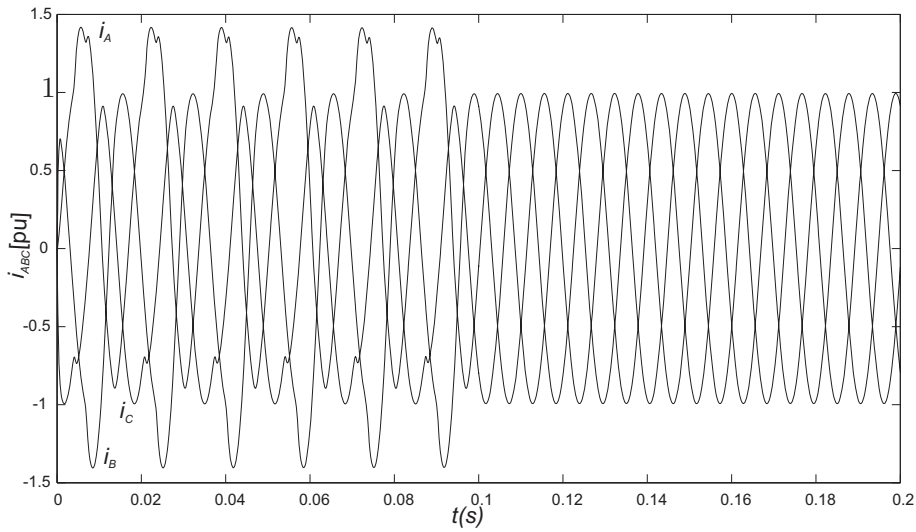


Figura 4.2: Energización del transformador con $R_L = 1$ p.u.

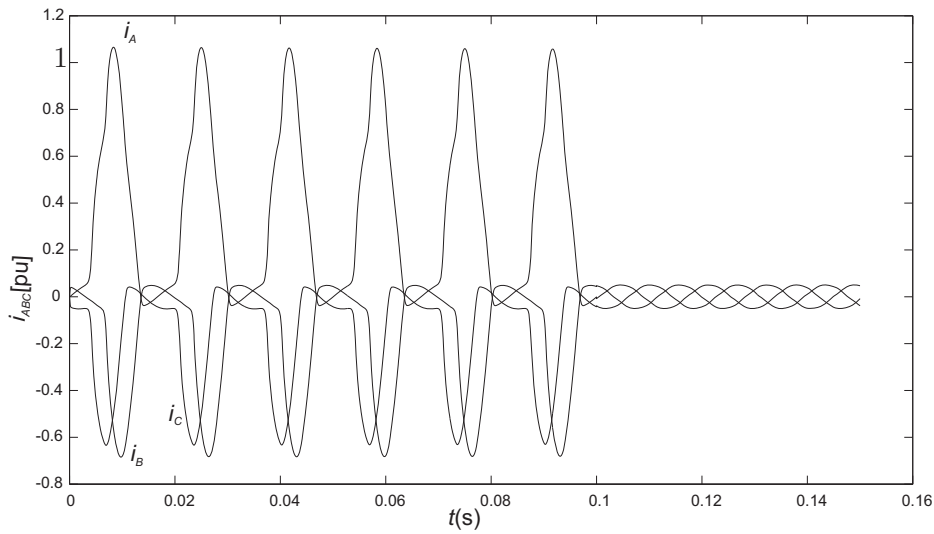


Figura 4.3: Energización del transformador con $R_L = 20$ p.u.

picos que puede alcanzar la corriente de inrush cuando el transformador está energizado con poca carga (en vacío). También se observa que este sistema es pobremente amortiguado, ya que para llegar al estado estacionario se llevaría mucho tiempo si se dejará seguir la simu-

lación, haciéndola computacionalmente costosa, por lo que utilizar la aceleración numérica es de gran utilidad, puesto que el estado estable se encuentra en forma rápida, existiendo para este caso.

La solución al estado estacionario es obtenida por medio de (4.7), donde la matriz de transición de estados Φ es calculada mediante la DN y la DA. La Tabla 4.2 muestra el número de iteraciones requeridas para llegar al estado estable y el tiempo total de simulación de ambas técnicas, se observa que para llegar al estado estable la DN requiere de 9 iteraciones, mientras que las fórmulas de Euler y la Regla Trapezoidal se llevaron más iteraciones. En cuanto a tiempo se refiere, el EHD fue el más rápido de todos. La Tabla 4.3 muestra que cambiando el parámetro k se puede reducir el tiempo de simulación, observándose que ahora el método más rápido, en cuanto a tiempo se refiere, fue el Euler Hacia Atrás, requiriendo de 12 iteraciones para converger al estado estable.

Tabla 4.2: DN vs DA para el transformador.

Aceleración al estado estable				
	DN	DA		
		EHA	EHD	RT
k	512	512	512	512
Iteraciones	9	13	12	11
Tiempo Total	2.1 seg	1.58 seg	1.45 seg	1.52 seg

Tabla 4.3: DN vs DA para el transformador reduciendo k .

Aceleración al estado estable				
	DN	DA		
		EHA	EHD	RT
k	512	64	512	64
Iteraciones	9	12	12	13
Tiempo Total	2.1 seg	1.13 seg	1.45 seg	1.22 seg

4.5. Análisis de estabilidad angular transitoria del generador síncrono considerando aceleración al estado estacionario

Se analizará el turbogenerador a barra infinita considerando la aceleración numérica al estado estacionario, éste estará modelado en sus coordenadas **abc** (Ver Sección 3.3). Se implementarán sus controles de voltaje y velocidad cuyos modelos se indican en las Secciones 3.3.2 y 3.3.3. Se implementará computacionalmente el modelo diferencial-algebraico representado por las ecuaciones (3.6), (3.7), (3.38) y (3.39), respaldándose en (3.8) utilizando Runge-Kutta de cuarto orden y/o Regla Trapezoidal. Los parámetros para el turbogenerador se proporcionan en el Apéndice A [Pérez04].

El caso de estudio se solucionará de la siguiente forma; se arranca la simulación con valores iniciales cercanos al estado estacionario solucionando el sistema de EDA's durante 4 periodos de tiempo, llegando así de la posición 0 a la posición 1 de la Figura 4.4, posteriormente se aplica la aceleración para alcanzar el estado estacionario y se evalúan 2 ciclos del estado estable (posición 1 a 2). En este instante se aplica una falla de la fase **a** a tierra en la barra infinita permaneciendo esta durante 3 ciclos (posición 2 a 3), después de liberar la falla se analiza el transitorio durante 3 ciclos (posición 3 a 4). Finalmente se vuelve a aplicar aceleración al estado estacionario, evaluándose éste durante 2 ciclos (posición 4 a 5).

La Figura 4.4 muestra las corrientes en los devanados del estator y rotor del generador síncrono, donde se puede observar los estados de prefalla, falla, posfalla y estado estable final, es decir, el generador ante esta falla es estable.

La Figura 4.5 ilustra las variables mecánicas del generador síncrono, ya que por ser las variables más sensibles se puede observar con mayor precisión el comportamiento que presenta el generador síncrono ante una falla asimétrica. Se observa que este sistema también es pobremente amortiguado y que sí existe el estado estable, antes y después de ser aplicada la falla.

La Figura 4.6 muestra el comportamiento de las variables que conforman el control de excitación. Se observa que el estado estacionario tardaría mucho en encontrarse si la simulación se dejará seguir por varios periodos de tiempo. La Figura 4.7 muestra el

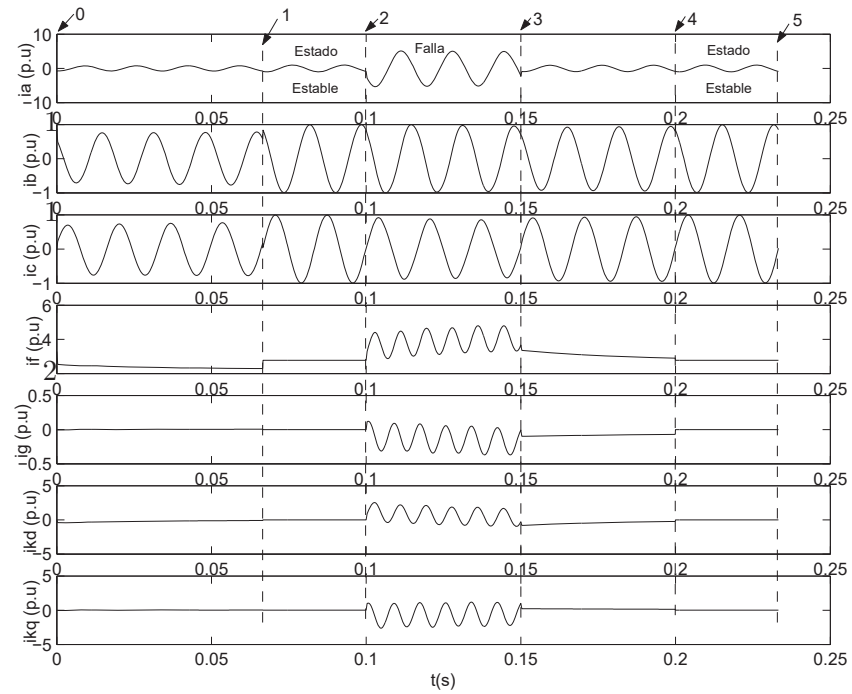


Figura 4.4: Corrientes en el estator y rotor.

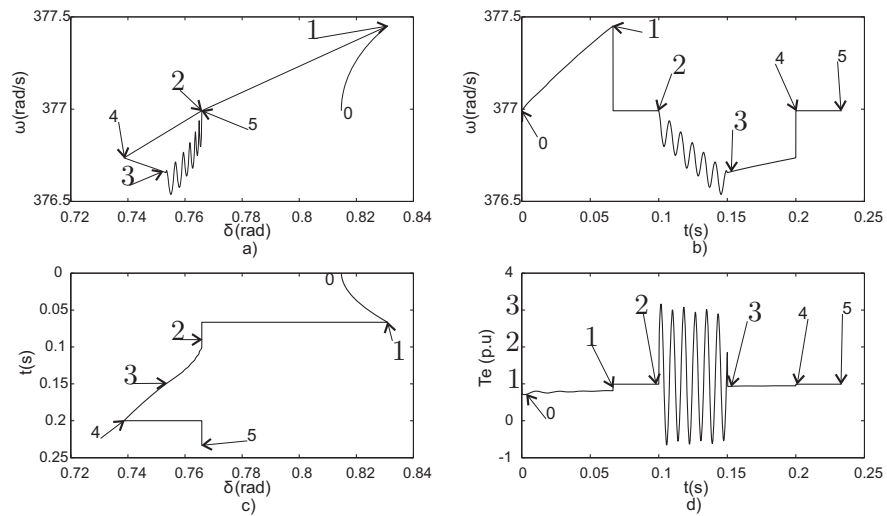


Figura 4.5: Variables mecánicas. a) plano de fase, b) velocidad angular, c) desplazamiento angular y d) par eléctrico.

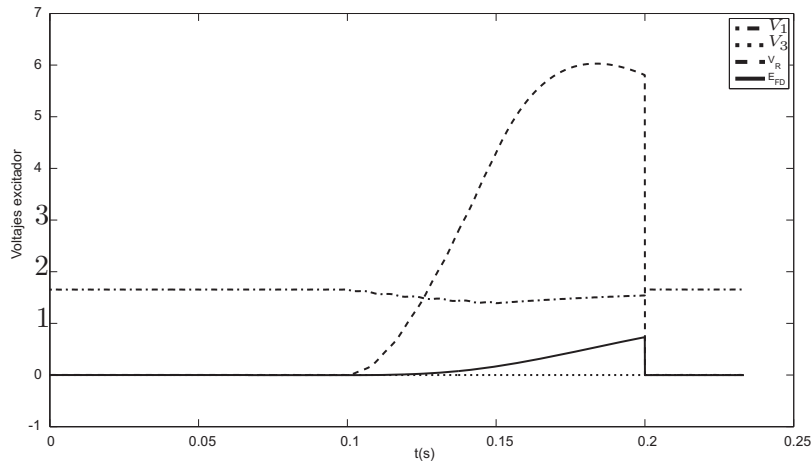


Figura 4.6: Control de excitación

comportamiento del control de velocidad.

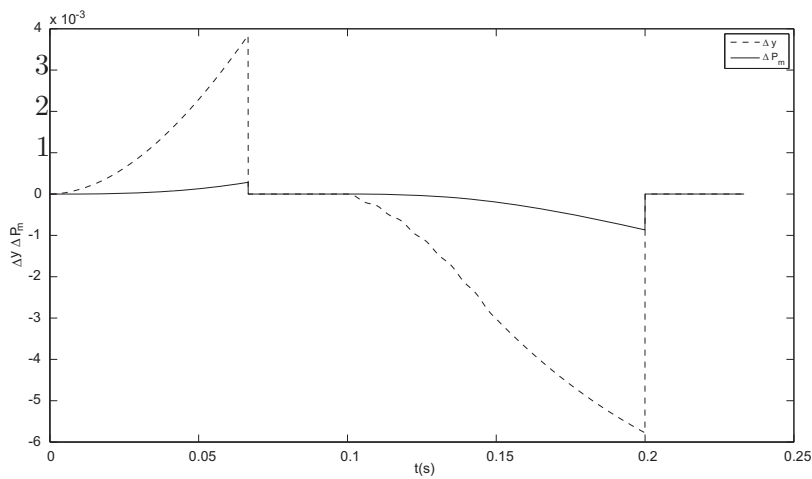


Figura 4.7: Control de velocidad

Al igual que en el transformador, la solución al estado estacionario es obtenida por medio de (4.7) y la Tabla 4.4 muestra el número de iteraciones requeridas para llegar al estado estable y el tiempo total de simulación de las técnicas presentadas en este proyecto. Se observa que para la primer aceleración, los primeros tres métodos convergen igual, en la segunda aceleración la DN tiene una iteración menos que los métodos de Euler, mientras

que la RT se llevó más iteraciones. En tiempo, resultó más eficiente el método de Euler Hacia Adelante. Se elige un valor de k diferente a 512 debido a que este valor genera una iteración más en la convergencia. Por lo tanto, k toma el valor de 519 en el EHA debido a que un valor menor genera una iteración más y similarmente para el EHD, en el cual k toma un valor de 518. La Tabla 4.5 muestra que cambiando el parámetro k se puede reducir el tiempo de simulación, observándose que nuevamente el método de EHD fue el más rápido.

Tabla 4.4: DN vs DA para el generador.

Primer acercamiento al estado estable				
	DN	DA		
		EHA	EHD	RT
k	512	512	512	512
Iteraciones	4	4	4	13
Segundo acercamiento al estado estable (Después de la falla)				
k	512	519	518	512
Iteraciones	3	4	4	13
Tiempo Total	2.9 seg	2.3 seg	2.2 seg	10 seg

Tabla 4.5: DN vs DA para el generador reduciendo k .

Primer acercamiento al estado estable				
	DN	DA		
		EHA	EHD	RT
k	512	64	32	4
Iteraciones	4	5	6	13
Segundo acercamiento al estado estable (Después de la falla)				
k	512	64	32	4
Iteraciones	3	5	5	12
Tiempo Total	2.9 seg	1.31 seg	1.25 seg	2 seg

4.6. Análisis de estabilidad angular transitoria del generador síncrono. Caso inestable

Se analizará el turbogenerador a barra infinita sin considerar sus elementos de control. El objetivo de este caso de estudio es analizar la estabilidad angular transitoria para ver el comportamiento ante una falla trifásica a tierra con el propósito de hacer inestable el sistema y observar el comportamiento que tiene el método Newton en el proceso de aceleración. Para esto se analizará el comportamiento del ángulo δ y la velocidad del rotor ω , Figura 4.8.

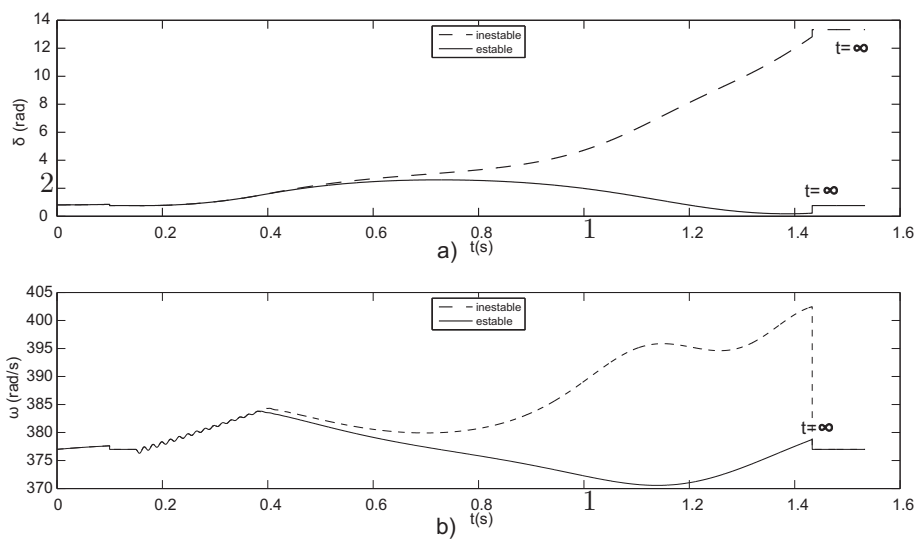


Figura 4.8: Caso inestable. a) ángulo, b) velocidad

Este sistema se solucionará de la siguiente forma, se arranca el sistema dejando 6 ciclos de transitorio (0.1 segundos), aplicando la aceleración en este momento analizando el estado estacionario durante 3 ciclos (0.05 segundos). En este instante se aplica la falla trifásica a tierra. Ya que el objetivo es analizar cuando el sistema se hace inestable, se hará una comparación con el caso estable para observar su comportamiento, por lo tanto para el caso estable la falla tendrá una duración de 14 ciclos (0.233 segundos) y para el caso inestable la falla durará 15 ciclos (0.25 segundos). Al ser liberada la falla se dejarán 63 ciclos (1.05 segundos) de transitorio de posfalla con la finalidad de observar como el sistema se hace

inestable cuando la falla dura 15 ciclos. Después de esto se aplica nuevamente la aceleración al estado estacionario analizándolo durante 3 ciclos (0.05 segundos). La Figura 4.8 muestra el comportamiento del ángulo y la velocidad, se observa como éstos se incrementan cuando la falla dura 15 ciclos haciendo al sistema inestable. Como se puede observar, el Newton en el proceso de aceleración converge en ambos casos, la diferencia está en que para el caso estable el ángulo llega al mismo estado estacionario al que se llegó antes de aplicar la falla. Para el caso inestable el ángulo ya no llega al mismo punto de operación.

Para el análisis de estabilidad angular transitoria se hará una explicación más detallada de como saber si el sistema es estable o inestable debido a que el Newton converge en ambos casos. La Figura 4.9 muestra el ángulo y la velocidad para el caso estable, para éste se hicieron pruebas dejando diferentes periodos de tiempo en el transitorio de posfalla antes de aplicar el proceso de aceleración (número 1, estado estable no factible, número 2, estado estable inicial).

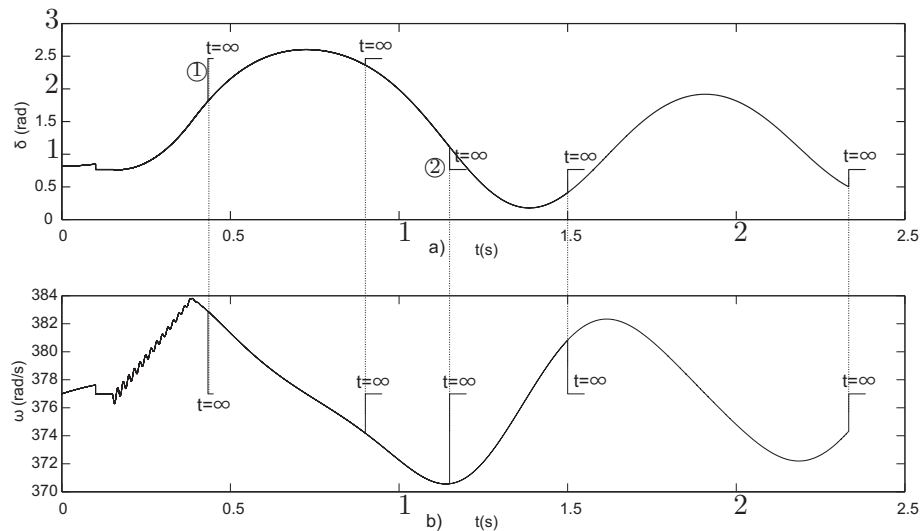


Figura 4.9: Caso estable. a) ángulo, b) velocidad

Se observa que al aplicar la falla, la velocidad se incrementa provocando una desaceleración al liberar ésta. De las pruebas realizadas se muestra que aplicar el proceso de aceleración al estado estacionario durante la etapa de desaceleración de la máquina ocasiona que el método de Newton llegue a converger a otra condición de operación en el ángulo δ ,

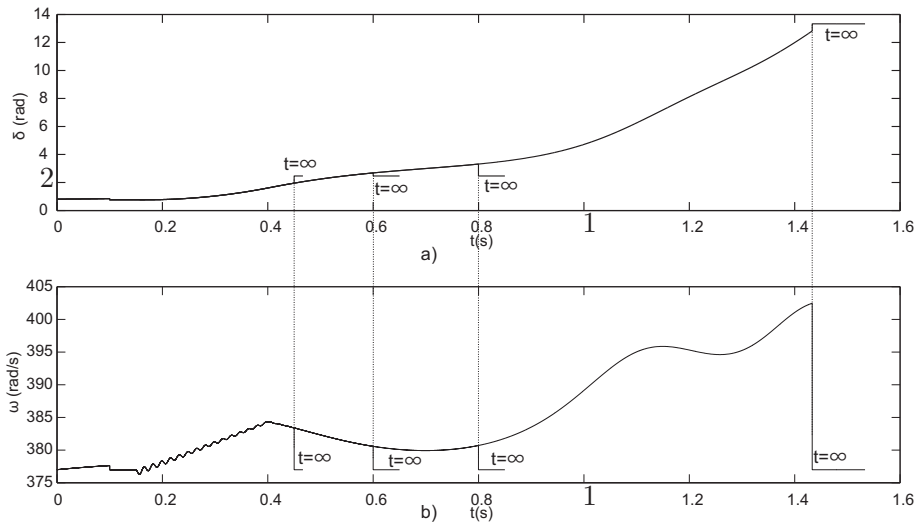


Figura 4.10: Caso inestable. a) ángulo, b) velocidad

indicando que el sistema es inestable lo cual no es cierto, 4.9. Por esta razón se observó que para obtener una respuesta correcta sobre la operación del sistema es aplicar la aceleración al estado estacionario durante la etapa de aceleración para este caso. Para corroborar esta explicación la Figura 4.10 muestra las mismas variables pero para el caso cuando el sistema se hace inestable, observándose que al aplicar el proceso de aceleración al estado estacionario durante la etapa de aceleración de la velocidad, éste converge hacia otro punto de operación en el ángulo δ , indicando que el sistema es inestable. De aquí la importancia de las condiciones iniciales del sistema al momento de aplicar el proceso de aceleración al estado estable. Para este caso el proceso de aceleración debe aplicarse cuando el $\Delta\omega$ sea positivo, es decir que la máquina esté acelerándose.

4.7. Análisis de estabilidad angular transitoria del sistema Generador-Transformador-Línea-Barra Infinita considerando aceleración al estado estacionario

En la Sección anterior se analizó el generador a barra infinita incluyendo sus controles. Esta Sección analizará el sistema Generador-Transformador-Línea-Barra Infinita, con

el objetivo de observar la interacción del modelo implementado para el transformador con el modelo del generador y línea a barra infinita, para este caso, el transformador trifásico estará conectado en delta-estrella aterrizada y el tipo de generador utilizado es el hidrogenador con sus respectivos controles de excitación y de velocidad. La Figura 4.11 muestra el sistema a solucionar y sus parámetros son proporcionados en el Apéndice A.

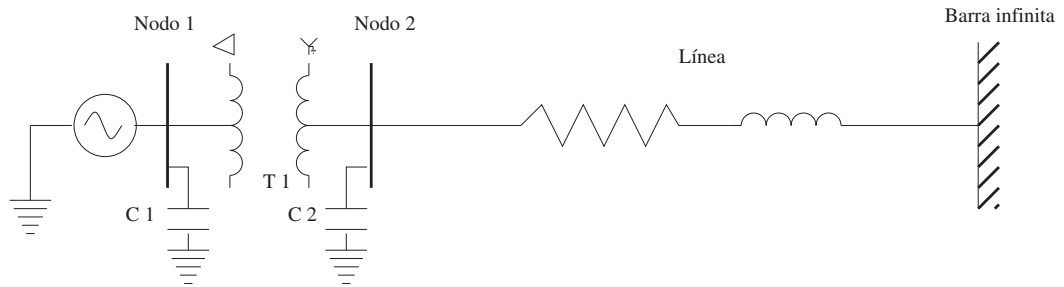


Figura 4.11: Sistema Generador-Transformador-Línea-Barra Infinita.

4.7.1. Caso de estudio

El sistema de la Figura 4.11 se solucionará de la siguiente manera. Arrancando el sistema con condiciones iniciales cercanas al estado estacionario, se dejará un tiempo de transitorio de 2 segundos (120 ciclos), aplicando la aceleración en este momento para analizar el estado estable durante 0.05 segundos (3 ciclos). En este instante se aplica una falla monofásica de la fase **c** a tierra en el nodo 2, la cual tendrá una duración de 0.1 segundos (6 ciclos), dejando el transitorio de posfalla durante 0.05 segundos (3 ciclos) para después aplicar nuevamente la aceleración al estado estable y analizarlo durante 0.05 segundos (3 ciclos).

La Figura 4.12 muestra las corrientes del rotor y estator de la máquina síncrona. Aquí se puede observar el comportamiento transitorio al iniciar la simulación, el primer instante en que se obtiene el estado estable, el momento en que se aplica la falla asimétrica, así como su transitorio de posfalla y el estado estable final. Los detalles se muestran para la corriente de la fase **c** del estator (i_c) y para la corriente en el devanado de campo i_f del rotor, esto con la finalidad de observar con más precisión los estados de la simulación, es

decir, el estado estable antes de aplicar la falla, así como su comportamiento cuando la falla es aplicada.

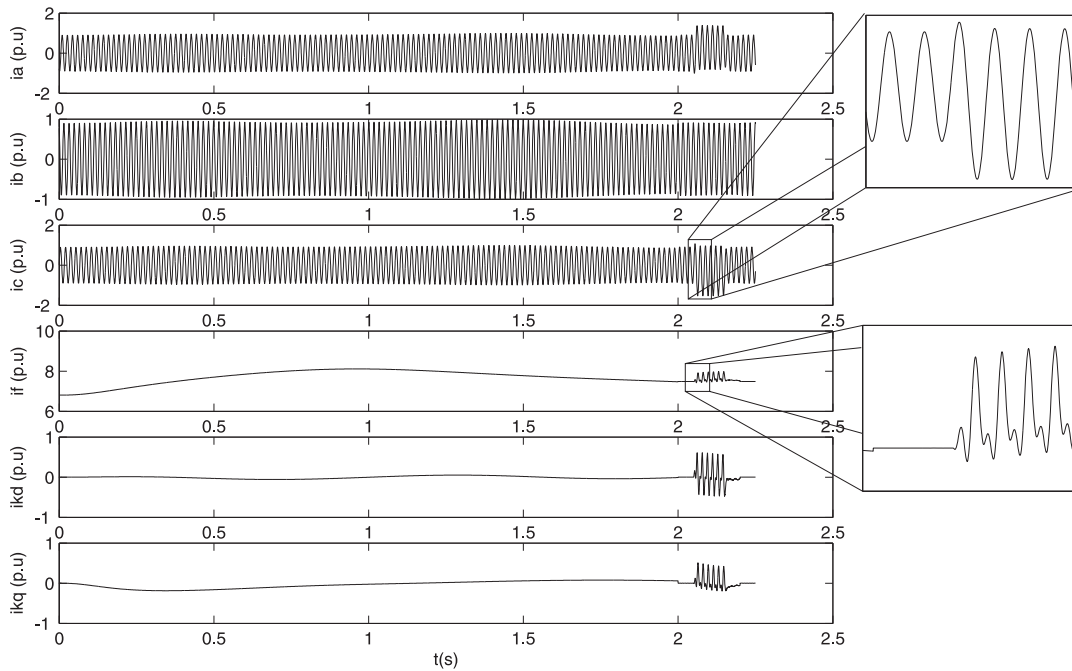


Figura 4.12: Corrientes en la máquina síncrona del sistema.

La Figura 4.13 ilustra las variables mecánicas del sistema, el inciso a) muestra el plano de fase donde los números indican los pasos que sigue la simulación; 1 transitorio, 2 estado estable, 3 falla monofásica, 4 transitorio de posfalla y 5 estado estable. El inciso b) indica la velocidad angular, el inciso c) presenta el desplazamiento angular y por último el inciso d) muestra el par eléctrico de la máquina. Se puede observar que si no se aplicara la aceleración numérica, el sistema tiende eventualmente a llegar a su estado estable, lo cual requeriría de mucho tiempo y esfuerzo computacional. De esta manera se corrobora la gran ayuda que ofrece el aplicar este tipo de técnicas de aceleración al estado estacionario y de esta forma poder realizar, para este caso de estudio, un análisis a detalle de la estabilidad angular transitoria. Por lo tanto es fácil obtener el estado estable antes y después de aplicar la falla asimétrica al sistema y poder decir que este sistema es estable ante esta falla.

La Figura 4.14 muestra las corrientes en los devanados primario y secundario del

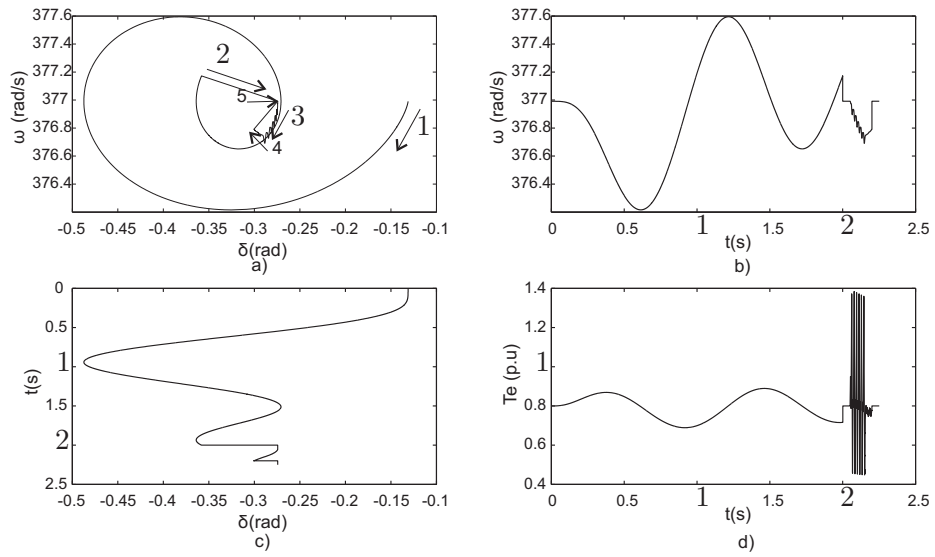


Figura 4.13: Variables mecánicas y par eléctrico del sistema.

transformador conectado en delta-estrella aterrizada, observándose en las ampliaciones que éstas están defasadas 180° .

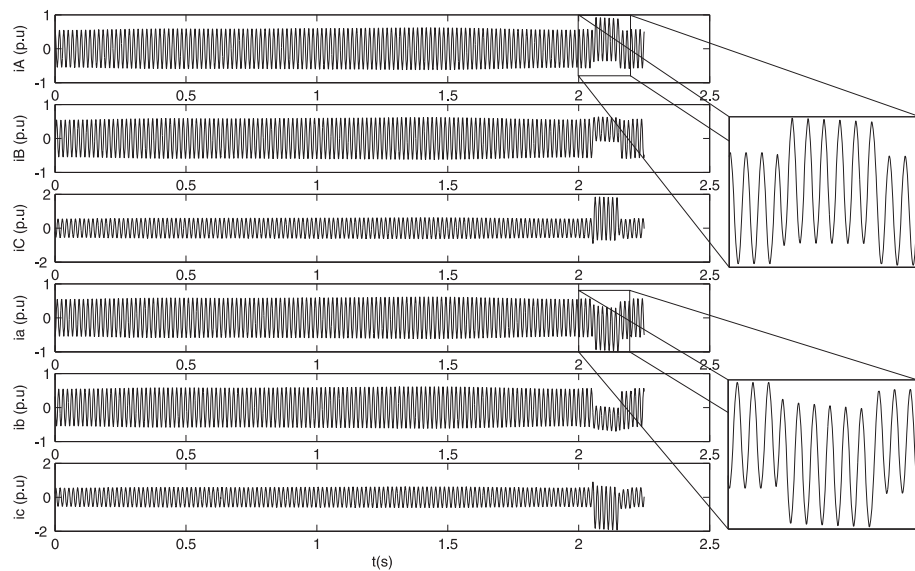


Figura 4.14: Corrientes en el transformador delta-estrella aterrizada del sistema.

Para este caso, solo los métodos de la diferenciación numérica y Euler hacia atrás son presentados, esto debido a que fueron los que mejores resultados presentaron. De esta forma, la Tabla 4.6 muestra el tiempo total de simulación, así como el número de iteraciones requeridas por cada método para obtener el estado estacionario del sistema. Se observa que la DN tiene mejores características de convergencia que el EHA, en lo que respecta al tiempo el Euler Hacia Atrás fue más rápido.

Tabla 4.6: DN vs DA para el sistema

Primer acercamiento al estado estable		
	DN	DA EHA
k	512	40
Iteraciones	3	23
Segundo acercamiento al estado estable (Después de la falla)		
k	512	40
Iteraciones	3	24
Tiempo Total	19.23 seg	17.6 seg

4.8. Resumen

En este Capítulo se presenta una forma rápida de obtener la respuesta en estado estable para sistemas pobremente amortiguados cuyas entradas son periódicas. Esta técnica está basada en el método de Newton y requiere del cálculo de la matriz de transición de estados Φ , por lo que varios algoritmos fueron presentados para su obtención. De los algoritmos presentados, las formulaciones de Euler y la Regla Trapezoidal requieren de la información del Jacobiano, por lo que la diferenciación automática es utilizada para su cálculo, ya que ésta entrega derivadas exactas en una forma rápida y eficiente.

Se presentó un análisis dinámico del transformador trifásico de cinco columnas en sus coordenadas de fase **abc** conectado en delta-estrella aterrizada. Se muestra su comportamiento cuando está conectado a plena carga y en vacío, además de su estado estable obtenido mediante la aceleración numérica. De los cuatro métodos presentados, el método de Euler hacia atrás fue el más rápido en cuanto a tiempo se refiere, requiriendo doce

iteraciones para converger al estado estable.

Se presentó un análisis dinámico y en estado estable del generador síncrono modelado en sus coordenadas de fase **abc** a barra infinita incluyendo sus controles de velocidad y excitación. La aceleración numérica al estado estable es utilizada debido a la gran ayuda que otorga en el análisis de estabilidad angular transitoria, ya que el estado estable puede encontrarse rápidamente antes y después de aplicar una falla monofásica. Para este caso, el método de Euler hacia adelante fue el más rápido en cuanto a tiempo se refiere, requiriendo cinco iteraciones en cada acercamiento al estado estable para converger.

También fue presentado el análisis de estabilidad angular transitoria del sistema Generador-Transformador-Línea-Barra Infinita, con lo cual se logra la interacción del modelo del generador síncrono con el modelo del transformador, ambos en sus coordenadas **abc**. De igual forma que en el análisis del generador, se utiliza la aceleración numérica para obtener el estado estable del sistema, antes y después de que una falla monofásica es aplicada, por lo que es fácil estudiar la estabilidad angular transitoria y el estado estacionario de este sistema. Para este caso, la DN tiene mejores características de convergencia, ya que sólo requirió de 3 iteraciones en cada aplicación de la aceleración numérica. En cuanto a tiempo se refiere, la fórmula de Euler Hacia Atrás fue más rápida.

Capítulo 5

Conclusiones

Un análisis de estabilidad angular transitoria en el dominio del tiempo en el marco de referencia de coordenadas **abc** fue presentado. Se eligió este marco de referencia debido a que se presentaron grandes ventajas para poder hacer un análisis y estudios en detalle de los SEP's. Se presentó una herramienta para el cálculo de derivadas (DA), la cual fue aplicada para la obtención rápida del estado estable en SEP's, presentando buenos resultados y abriéndose la posibilidad de explotarla en otras áreas y aplicaciones de la ingeniería eléctrica.

5.1. Conclusiones

Del análisis desarrollado durante el trabajo de esta Tesis, se tienen las siguientes conclusiones.

Se presentaron varias formas de calcular derivadas, destacando entre éstas la llamada diferenciación automática, esto debido a las desventajas que las demás poseen, por lo que las hacen imprácticas para muchas aplicaciones grandes. Esta herramienta es de gran ayuda en la solución de cualquier problema donde se requiera el cálculo de la derivada de una función(es), ya sea en forma de gradiente, Jacobiano o Hessiano, esto se debe a que nos entrega derivadas en una forma exacta y rápida. Se describió cómo funcionan los modos hacia adelante y hacia atrás de la diferenciación automática y la combinación de estos dos. Se explicó el funcionamiento de la herramienta de DA llamada ADIFOR, la cual genera código eficiente para el cálculo de derivadas para funciones de programas escritos en Fortran 77.

Se utilizó un método tipo Newton para obtener el estado estable en forma rápida, lo cual es de gran ayuda en el análisis de estabilidad angular transitoria, ya que el estado estable puede encontrarse rápidamente. Esto debido a que en sistemas pobremente amortiguados, utilizar técnicas de punto a punto sería computacionalmente costoso. Dado que el algoritmo de Newton requiere del cálculo de la matriz de transición de estados Φ , fueron presentados cuatro métodos para su obtención. Se observó que los métodos de Euler y la Regla Trapezoidal requieren del cálculo de la matriz Jacobiano, por lo que los algoritmos de la diferenciación automática son utilizados para su cálculo, esto debido a las ventajas que presenta esta herramienta.

Se presentó un modelo completo de la máquina síncrona en sus coordenadas **abc** considerando los elementos de control, tales como el sistema de excitación y el control de velocidad del gobernador. Se realizó un análisis dinámico y en estado estable del generador, este último aplicando la aceleración numérica al estado estacionario incluyendo DA.

Se presentó el modelo del transformador trifásico de cinco columnas en base a su circuito magnético. Se utilizó la función arcotangente para representar la saturación del material en el núcleo laminado, esto debido a su rapidez y fácil implementación.

También fue presentado el análisis de estabilidad angular transitoria del sistema Generador-Transformador-Línea-Barra Infinita, con lo cual se logra la interacción del modelo del generador síncrono con el modelo del transformador. Se utilizó la aceleración numérica con DA para obtener el estado estable de este sistema, lo cual fue de gran ayuda para el estudio de estabilidad angular transitoria, ya que éste se puede analizar durante el tiempo que se desee, para posteriormente aplicar aceleración y de esta forma llegar a su estado estable.

Los modelos mostrados en este trabajo, todos éstos en sus coordenadas **abc**, presentan grandes ventajas en los estudios de estabilidad angular transitoria ante fallas asimétricas. Los métodos numéricos utilizados para el estudio y análisis de estabilidad angular transitoria, fueron la Regla Trapezoidal (implícito) y el Runge-Kutta de Cuarto Orden (explícito), concluyendo que los resultados numéricos obtenidos son virtualmente iguales.

De los resultados obtenidos para el cálculo de Φ , se tiene que los métodos de Euler y la Regla Trapezoidal funcionaron adecuadamente, ya que en los análisis presentados fueron

más rápidos que la diferenciación numérica, en cuanto a tiempo se refiere. Siendo ésta última más estable y exacta, debido a que se requieren menos iteraciones para llegar al estado estable, sin saber el por qué de sus características de convergencia.

Se usó la diferenciación automática para obtener Φ de manera alterna y usarla directamente en el algoritmo de aceleración, proporcionando mejores resultados en la velocidad del cálculo. En cuanto a precisión se refiere, en esta aplicación tal diferencia no es notoria y se espera que mejorando el algoritmo para obtener Φ a partir del Jacobiano esta diferencia sea de magnitudes mayores. Por supuesto esto último queda fuera del alcance de la presente investigación.

Se mostró la aplicación de la diferenciación automática al análisis de la estabilidad angular transitoria. Esperando se vea el amplio horizonte que tiene esta herramienta en otros estudios y desarrollos de los sistemas eléctricos de potencia.

El análisis de estabilidad angular transitoria se realiza mediante la convergencia del Newton en el proceso de aceleración al estado estacionario, debido a que dependiendo de la condición de estado estable al cual llega éste se define si el sistema es estable o inestable. Para el caso en que el disturbio provocado al sistema ocasione un incremento en la velocidad de la máquina, se debe aplicar la aceleración al estado estable cuando el $\Delta\omega$ sea positivo, es decir, cuando la máquina se acelera, si la condición de estado estable es la misma a la que convergió antes de aplicar la falla el sistema es estable, de lo contrario es inestable. De igual manera, si el disturbio ocasiona un decremento en la velocidad, el proceso de aceleración se debe aplicar cuando exista una desaceleración en la máquina.

Se desarrollaron rutinas programadas en Fortran 77 para los estudios de este proyecto de Tesis, comparando los resultados obtenidos con los presentados por [Pérez04] cuyas rutinas están programadas en Matlab[®]. Estos no se muestran en este trabajo debido a que son los mismos, la diferencia se encuentra en el tiempo de cálculo siendo mucho más rápido en Fortran 77.

5.2. Trabajos futuros

De los resultados obtenidos en la realización de esta investigación se tienen las siguientes recomendaciones para trabajos futuros.

- Investigar y desarrollar una forma más eficiente para el cálculo de la matriz de transición de estados, la cual dependa del Jacobiano, para de esta forma explotar las herramientas de la diferenciación automática y así hacer más rápida la convergencia al estado estable.
- Explotar la diferenciación automática en otros estudios de los sistemas eléctricos de potencia como son:
 - Optimización
 - Análisis de sensitividades
 - Análisis de mercados y financieros
 - Estabilidad transitoria en sistemas multimáquina
 - Estimación de parámetros de elementos y/o sistemas
 - Colocación eficiente de dispositivos FACTS, etc

En otras palabras, teniendo una herramienta que proporcione el Jacobiano y/o Hessiano exactos de un sistema se pueden realizar cualquier tipo de estudios al mismo de manera rápida y eficiente, ya que se conoce la parte vital del sistema, la que indica cómo va a reaccionar ante alguna perturbación interna o externa.

Referencias

- [Anderson94] Anderson, P. M. y Fouad, A. A. *Power System Control and Stability*. IEEE Press, 1994.
- [Aprille72] Aprille, T. J. y Trick, T. Steady-state analysis of nonlinear circuits with periodic inputs. *Proceedings of the IEEE*, 60(1):108–114, January 1972.
- [Arrillaga90] Arrillaga, J. y Arnold, C. P. *Computer Analysis of Power Systems*. John Wiley & Sons, 1990.
- [Averick94] Averick, B. M., Moré, J. J., Bischof, C. H., Carle, C., y Griewank, A. Computing large sparse jacobian matrices using automatic differentiation. *SIAM Journal on Scientific Computing*, 15(2):285–294, 1994.
- [Bergen86] Bergen, A. R. *Power System Analysis*. Prentice-Hall, 1986.
- [Bischof91a] Bischof, C., Carle, A., Corliss, G., Griewank, A., y Hovland, P. Adifor: Fortran source translation for efficient derivatives. Inf. Téc. MCS–P278–1291, Center for Research on Parallel Computation, Argonne, Ill., 1991.
- [Bischof91b] Bischof, C., Carle, A., Corliss, G., Griewank, A., y Hovland, P. Adifor: Generating derivative codes from fortran programs. *Technical Report CRPC-TR91185*, Center for Research on Parallel Computation, December 1991.
- [Bischof92a] Bischof, C., Carle, A., Corliss, G., Hovland, P., y Griewank, A. Getting started with adifor. Inf. Téc. ANL/MCS–TM–164, Mathematics and Computer Science Division, Argonne, Ill., 1992.

- [Bischof92b] Bischof, C. H., Carle, A., Corliss, G. F., y Griewank, A. Adifor: Automatic differentiation in a source translation environment. *En* P. S. Wang, ed., *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, págs. 294–302. ACM Press, New York, 1992.
- [Bischof93] Bischof, C. H., Griewank, A., y Khademi, P. M. First theory institute on computational differentiation. Inf. Téc. ANL/MCS–TM–183, Mathematics and Computer Science Division, Argonne, Ill., 1993.
- [Bischof94a] Bischof, C., Carle, A., Hovland, P., Khademi, P., y Mauer, A. Adifor 2.0 users' guide (revision d). *Technical Report ANL/MCS-TM-192, Mathematics and Computer Science Division, Argonne National Laboratory*, 1994.
- [Bischof94b] Bischof, C., Carle, A., Khademi, P., y Mauer, A. The adifor 2.0 system for the automatic differentiation of fortran 77 programs. *Technical Report ANL-MCS-P481-1194, Mathematics and Computer Science Division, Argonne National Laboratory*, 1994.
- [Bischof94c] Bischof, C., Carle, A., Khademi, P., y Pusch, G. Automatic differentiation: Obtaining fast and reliable derivatives - fast. *Technical Report CRPC-TR94534, Center for Research on Parallel Computation*, December 1994.
- [Bischof02] Bischof, C. H., Hovland, P. D., y Boyana, N. Implementation of automatic differentiation tools. *In Proceedings of the 2002 ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation (Portland, Oregon, January 14 - 15). PEPM '02. ACM Press, New York, NY*, págs. 98–107, 2002.
- [ChainRule99] ChainRule. <http://mathworld.wolfram.com/chainrule.html>. The Chain Rule, 1999.
- [Chapra99] Chapra, S. C. y Canale, R. P. *Numerical Methods for Engineers*. McGraw-Hill, 1999.

- [Chua75] Chua, L. O. y Lin, P.-M. *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Prentice-Hall, 1975.
- [Coleman98] Coleman, T. F. y Arun, V. Admat: An automatic differentiation toolbox for matlab. *Technical report, Computer Science Department, Cornell University*, 1998.
- [Coleman00] Coleman, T. F. y Arun, V. Admit-1: Automatic differentiation and matlab interface toolbox - the matlab ad tool version. *ACM Transactions on Mathematical Software*, 26(1):150–175, 2000.
- [García00a] García, S. *Modelado del Transformador Trifásico en el Dominio del Tiempo y Análisis de su Operación en Estado Transitorio y Estado Estacionario Periódico*. Tesis de Maestría, Universidad Michoacana de San Nicolás de Hidalgo, 2000.
- [García00b] García, S., Medina, A., y Pérez, C. A state space single-phase transformer model incorporating nonlinear phenomena of magnetic saturation and hysteresis for transient and periodic steady-state analysis. *IEEE Summer Meeting, Seattle, Washington, EEUU*, págs. 2417–2421, July 2000.
- [Gear71] Gear, C. W. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, 1971.
- [Griewank89] Griewank, A. On automatic differentiation. En M. Iri y K. Tanabe, eds., *Mathematical Programming: Recent Developments and Applications*, págs. 83–108. Kluwer Academic Publishers, 1989.
- [Griewank90] Griewank, A., Juedes, D., y Srinivasan, J. Adol-c, a package for the automatic differentiation of algorithms written in c/c++. *Preprint MCS-P180-1190, Mathematics and Computer Science Division, Argonne National Laboratory*, 1990.
- [Griewank91] Griewank, A. The chain rule revisited in scientific computing. *Preprint*

-
- MCS-P227-0491, Mathematics and Computer Science Division, Argonne National Laboratory, 1991.*
- [Griewank00] Griewank, A. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. Frontiers in Applied Mathematics, SIAM, 2000.
- [Ibsais97] Ibsais, A. y Ajjarapu, V. The role of automatic differentiation in power system analysis. *IEEE transactions on power systems*, 12(2):592–597, May 1997.
- [Jerolimsky94] Jerolimsky, M. y Levacher, L. A new method for fast calculation of jacobian matrices: Automatic differentiation for power system simulation. *IEEE Transactions on Power Systems*, 9(2):700–706, May 1994.
- [Juedes91] Juedes, D. W. A taxonomy of automatic differentiation tools. En A. Griewank y G. F. Corliss, eds., *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, págs. 315–329. SIAM, Philadelphia, Pnn., 1991.
- [Krause87] Krause, P. C. *Analysis of Electric Machinery*. McGraw-Hill, 1987.
- [Krause89] Krause, P. C. y Wasynczuk, O. *Electromechanical Motion Devices*. McGraw-Hill, 1989.
- [Kundur94] Kundur, P. *Power System Stability and Control*. McGraw-Hill, 1994.
- [Kundur04] Kundur, P., Paserba, J., Ajjarapu, V., Anderson, G., Bose, A., Canizares, C., Hatziargyriou, N., Hill, D., Stankovic, A., Taylor, C., Van Cutsem, T., y Vittal, V. Definitions and classification of power system stability, ieeec/cigre joint task force on stability terms and definitions report. *IEEE Transactions on Power Systems*, 19(2):1387–1401, May 2004.
- [Li00] Li, D. y Tymerski, R. Comparison of simulation algorithms for accelerated determination of periodic steady state of switched networks. *IEEE Transactions on Industrial Electronics*, 47(6):1278–1285, December 2000.

- [Liu02] Liu, X., Xu, S., Li, Z., y Chiang, H. A new methodology for small-signal stability analysis of facts system. *Power System Technology*, 3:1928–1932, 2002.
- [Orfanogianni00] Orfanogianni, T. *A flexible software environment for steady-state power flow optimization with series FACTS devices*. Tesis Doctoral, Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, 2000.
- [Pérez99] Pérez, C. Representación de la saturación e histéresis, aproximando mediante la función arcotangente. *RVP/IA-99, IEEE, Acapulco, Gro., México*, págs. 118–121, Julio 1999.
- [Pérez00] Pérez, C. Fitting saturation and hysteresis via arctangent functions. *IEEE, Power Engineering Review, EEUU*, págs. 55–57, November 2000.
- [Pérez04] Pérez, C. *Modelos Avanzados de Elementos del Sistema Eléctrico de Potencia en Coordenadas de Fase para Estudios de Estabilidad Dinámica*. Tesis Doctoral, Universidad Autónoma de Nuevo León, Monterrey, N.L., Abril 2004.
- [Rall96] Rall, L. B. y Corliss, G. F. An introduction to automatic differentiation. En M. Berz, C. H. Bischof, G. F. Corliss, y A. Griewank, eds., *Computational Differentiation: Techniques, Applications, and Tools*, págs. 1–17. SIAM, Philadelphia, PA, 1996.
- [Rodríguez04] Rodríguez, O. y Medina, A. Efficient methodology for the transient and periodic steady-state analysis of the synchronous machine using a phase coordinates model. *IEEE Transactions on Energy Conversion*, 19(2):464–466, June 2004.
- [Rugh93] Rugh, W. J. *Linear System Theory*. Prentice-Hall, 1993.
- [Saadat99] Saadat, H. *Power System Analysis*. McGraw-Hill, 1999.
- [Semlyen95] Semlyen, A. y Medina, A. Computation of the periodic steady state in systems with nonlinear components using a hybrid time and frequency

- domain methodology. *IEEE Transactions on Power Systems*, 10(3):1498–1504, August 1995.
- [Solodovnik01] Solodovnik, E., Cokkinides, G., Dougal, R., y Meliopoulos, A. P. S. Nonlinear power system component modeling using symbolically assisted computations. *Power Engineering Society Summer Meeting. IEEE*, 3:1439–1444, 2001.
- [TAPENADE03] TAPENADE. The tapenade tutorial. <http://www-sop.inria.fr/tropics/tapenade/tutorial.html>, 2003.
- [Usaola90] Usaola, J. *Régimen permanente de sistemas eléctricos de potencia con elementos no lineales mediante un procedimiento híbrido en los dominios del tiempo y la frecuencia*. Tesis Doctoral, ETS de II, Madrid, España, 1990.
- [Verma] Verma, A. Admat: Automatic differentiation in matlab using object oriented methods. Technical report, Computer Science Department, Cornell University.
- [Verma00] Verma, A. An introduction to automatic differentiation. *Current Science*, 78(7), April 2000.
- [Zamora06] Zamora, A., Perez, C., Garcia, S., y Villanueva, H. D. Automatic differentiation in power flow using adifor. *Electronics, Robotics and Automotive Mechanics Conference (CERMA'06)*, págs. 226–234, 2006.

Apéndice A

Parámetros de los elementos del SEP

Parámetros del turbogenerador, potencia base 100 MVA a 20 KV.

$$V_{bus\ infinito} = 1.0\ p.u.$$

$$V_f = 0.00172\ p.u.$$

$$\omega_{base} = 377\ rad/seg$$

$$T_m = 0.99\ p.u.$$

$$H = 5.6\ seg$$

$$L_l = 0.19\ p.u.$$

$$L_d = 1.8\ p.u.$$

$$L_q = 1.8\ p.u.$$

$$L_f = 0.1414\ p.u.$$

$$L_g = 0.8125\ p.u.$$

$$L_{kd} = 0.08125\ p.u.$$

$$L_{kq} = 0.0939\ p.u.$$

$$R_a = R_b = R_c = 0.003\ p.u.$$

$$R_f = 9.29e - 04\ p.u.$$

$$R_g = 0.00178\ p.u.$$

$$R_{kd} = 0.01334\ p.u.$$

$$R_{kq} = 0.00821\ p.u.$$

Parámetros del excitador

$$\begin{aligned}
T_R &= 0.06 \text{ seg} \\
T_F &= 0.715 \text{ seg} \\
T_A &= 0.05 \text{ seg} \\
T_E &= 0.5 \text{ seg} \\
K_R &= 1.0 \\
K_F &= 0.04 \\
K_A &= 40 \\
K_E &= -0.05
\end{aligned}$$

Parámetros del control de velocidad

$$\begin{aligned}
T_g &= 0.2 \text{ seg} \\
T_t &= 0.3 \text{ seg} \\
Rh &= 0.05
\end{aligned}$$

Parámetros del hidrogenerador, potencia base 100 MVA a 20 KV.

$$\begin{aligned}
V_{bus \text{ infinito}} &= 1.0 \text{ p.u.} \\
V_f &= 0.00059 \text{ p.u.} \\
\omega_{base} &= 377 \text{ rad/seg} \\
T_m &= 0.8 \text{ p.u.} \\
H &= 5.6 \text{ seg} \\
L_l &= 0.0369 \text{ p.u.} \\
L_d &= 0.2615 \text{ p.u.} \\
L_q &= 0.1477 \text{ p.u.} \\
L_f &= 0.06305 \text{ p.u.} \\
L_{kd} &= 0.04923 \text{ p.u.} \\
L_{kq} &= 0.03166 \text{ p.u.} \\
R_a = R_b = R_c &= 0.000585 \text{ p.u.} \\
R_f &= 1.3e - 04 \text{ p.u.} \\
R_{kd} &= 0.00418 \text{ p.u.} \\
R_{kq} &= 0.00434 \text{ p.u.}
\end{aligned}$$

Parámetros del excitador

$$\begin{aligned}
T_R &= 0.05 \text{ seg} \\
T_F &= 0.04 \text{ seg} \\
T_A &= 0.1 \text{ seg} \\
T_E &= 0.4 \text{ seg} \\
K_R &= 1.0 \\
K_F &= 2.0 \\
K_A &= 10 \\
K_E &= -1.0
\end{aligned}$$

Parámetros del control de velocidad

$$T_g = 0.2 \text{ seg}$$

$$T_w = 0.5 \text{ seg}$$

$$T_{RH} = 4.0 \text{ seg}$$

$$Rh = 0.05$$

$$Rt = 0.38$$

Parámetros del transformador

$$\text{Reluctancias de dispersión} = [15 \ 15 \ 15 \ 5.37 \ 5.37 \ 5.37]$$

$$\text{Longitudes} = [1 \ 1 \ 1 \ 1.16 \ 1.16 \ 1.9 \ 1.9]$$

$$\text{Sección transversal} = [1 \ 1 \ 1 \ 1.77 \ 1.77 \ 1.77 \ 1.77]$$

$$\text{Número de vueltas} = [1.732 \ 1.732 \ 1.732 \ 1 \ 1 \ 1]$$

$$\text{Constantes para la curva de saturación} = [0.7 \ 754 \ 0.95]$$

$$\text{Matriz de reisistencias} = \begin{bmatrix} 0.0015 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0015 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0015 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0015 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0015 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0015 \end{bmatrix}$$

Parámetros para la línea de transmisión.

$$\text{Resistencia} = \begin{bmatrix} 0.0064 & 0.0032 & 0.0032 \\ 0.0032 & 0.0064 & 0.0032 \\ 0.0032 & 0.0032 & 0.0064 \end{bmatrix}$$

$$\text{Inductancia} = \begin{bmatrix} 0.0152 & 0.0076 & 0.0076 \\ 0.0076 & 0.0152 & 0.0076 \\ 0.0076 & 0.0076 & 0.0152 \end{bmatrix}$$

$$C1 = \begin{bmatrix} 0.0039 & -0.0008 & -0.0008 \\ -0.0008 & 0.0039 & -0.0008 \\ -0.0008 & -0.0008 & 0.0039 \end{bmatrix}$$

$$C2 = \begin{bmatrix} 0.15 & 0 & 0 \\ 0 & 0.15 & 0 \\ 0 & 0 & 0.15 \end{bmatrix}$$