

**DESARROLLO DE UNA PLATAFORMA DE SIMULACIÓN Y DISEÑO DE UN
LABORATORIO DE PRUEBAS PARA GRUPOS ELECTRÓGENOS DE BAJA
POTENCIA PARA APLICACIONES EN ZONAS REMOTAS**

JUAN CAMILO PRADILLA RODRÍGUEZ



**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2012**

**DESARROLLO DE UNA PLATAFORMA DE SIMULACIÓN Y DISEÑO DE UN
LABORATORIO DE PRUEBAS PARA GRUPOS ELECTRÓGENOS DE BAJA
POTENCIA PARA APLICACIONES EN ZONAS REMOTAS**

JUAN CAMILO PRADILLA RODRÍGUEZ

**Trabajo de grado para optar el título de
INGENIERO ELECTRÓNICO**

Director:
Dr. JORGE LUIS CHACÓN

Codirector:
Dr. GABRIEL ORDOÑEZ PLATA



**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA ELECTRÓNICA Y
TELECOMUNICACIONES
BUCARAMANGA
2012**

Dedico este trabajo de grado a aquellas personas que inspiraron, ayudaron, animaron y permitieron la realización del mismo:

Al buen Padre Dios,

A mi madre

Rosalba Rodríguez Herrera,

A mi esposa

Yerly Margarita Quiroz Pertuz,

A mi hijo

Andrés Camilo Pradilla Quiroz,

A mi padre

Francisco Ignacio Pradilla Villaveces,

Al resto de mi familia y amigos.

AGRADECIMIENTOS

A Dios, por su paciencia, amor incondicional y fortaleza en medio de los momentos difíciles durante mis años de estudio.

A mi madre por su perseverancia y apoyo incondicional.

A mi esposa por acompañarme en la última etapa de mi carrera.

A mi hijo por ser mi inspiración, motivación e impulso final para hacerme ingeniero.

Al Dr. Jorge Luis Chacón y al Dr. Gabriel Ordóñez director y codirector (respectivamente), por su amabilidad, su paciente asesoría y respaldo.

A mis calificadores Dr. Julio César Chacón y Ms(c). ING. José Gabriel Plata, por aportar su conocimiento y tiempo al presente proyecto.

A mis líderes cristianos Dr. Néstor Angarita y Dr. Carlos Rico por su apoyo moral, sus oraciones, consejo y exhortaciones.

Y, a todos los que de una u otra forma aportaron al presente proyecto, gracias de corazón.

TABLA DE CONTENIDO

	pág.
INTRODUCCIÓN	14
1. TRANSFORMACIONES DE ENERGÍA EN UN GRUPO ELECTRÓGENO	15
1.1 MOTOR DE COMBUSTIÓN INTERNA	15
1.1.1 Medición de la eficiencia de un MCI de ciclo Otto	16
1.2 FUNCIONAMIENTO DE LAS MÁQUINAS ELÉCTRICAS COMO GENERADORES	16
1.2.1 Generador de inducción (GI)	16
1.2.1.1 Eficiencia de un GI	17
2. DESCRIPCIÓN DE LOS MODELOS MATEMÁTICOS DEL GRUPO ELECTRÓGENO	17
2.1 MODELO DE LA CARGA ELÉCTRICA	17
2.2 MODELOS DE MEDICIÓN DE LA POTENCIA ELÉCTRICA	18
2.2.1 Modelo de la medición de potencia en sistemas monofásicos sinusoidales	18
2.2.2 Modelo de la medición de potencia en sistemas trifásicos sinusoidales balanceados	18
2.2.3 Modelo de la medición de potencia en sistemas trifásicos sinusoidales desbalanceados	18
2.3 MODELOS DEL GI	20
2.3.1 Modelo del GI trifásico en régimen permanente	20
2.3.2 Modelo del GI trifásico en régimen transitorio	21
2.3.3 Modelo del GI monofásico en régimen permanente	21
2.3.4 Modelo del GI monofásico en régimen transitorio	21
2.4 MODELO DEL MCI DE CICLO OTTO	21
2.4.1 Flujo másico de aire	21
2.4.2 Modelo de pérdidas mecánicas	22
2.4.3 Potencia teórica y modelo de pérdidas térmicas	22
2.4.4 Flujo másico de combustible	22
3. ELABORACIÓN DE LA PLATAFORMA DE SIMULACIÓN	22
3.1 REQUISITOS	22
3.2 DISEÑO	23
3.2.1 Creación de usuario	23
3.2.2 Realizar prácticas	24
3.2.3 Modelo de simulación	25
3.2.4 Interfaz con el usuario	25
3.2.4.1 Ventana del menú principal	25
3.2.4.2 Ventanas de los submenús	25

3.2.4.3 Ventana de las prácticas de simulación	25
3.2.4.4 Ventanas de ayuda	26
3.3 IMPLEMENTACIÓN	26
3.3.1 Crear escenario	26
3.3.2 Simular escenario	26
3.3.3 Presentar resultados	26
3.3.4 Editar, crear y guardar modelos	26
3.4 PRUEBAS	26
3.5 PROPUESTA PARA EL MANUAL DE LABORATORIO	26
3.5.1 Explicación de las medidas de seguridad al manipular un grupo generador	27
3.5.2 Operación del grupo generador	27
3.5.3 Cálculo de las salidas del grupo generador	27
4. DISEÑO DEL LABORATORIO DE PRUEBAS PARA EL GRUPO ELECTRÓGENO DEL LABORATORIO DE MÁQUINAS TÉRMICAS ALTERNATIVAS DE INGENIERÍA MECÁNICA	27
4.1 MEDIDORES DE FLUJO MÁSICO	28
4.1.1 Medidor de flujo másico de combustible (gasolina)	28
4.1.2 Medidor de flujo volumétrico de combustible (biogás)	28
4.1.3 Medidor de flujo másico de aire	29
4.2 TERMÓMETROS	29
4.2.1 Para medir la temperatura ambiental	29
4.2.2 Para medir la temperatura en la admisión	29
4.2.3 para medir la temperatura de los gases de escape	29
4.3 BARÓMETRO	29
4.4 SISTEMA PARA MEDIR EL TORQUE MECÁNICO Y LA VELOCIDAD ANGULAR	29
4.5 SISTEMA PARA MEDIR LAS EMISIONES DEL MCI	30
4.6 ANALIZADOR DE REDES DE BAJA POTENCIA	30
4.7 TARJETA DE ADQUISICIÓN DE DATOS Y COMPUTADOR	30
CONCLUSIONES	30
RECOMENDACIONES	30
REFERENCIAS	30
BIOGRAFÍAS	31
ANEXOS	32

LISTA DE FIGURAS

	pág.
Figura 1. Esquema de un grupo generador	15
Figura 2. Modelo general del grupo generador	17
Figura 3. Modelo del grupo generador que discrimina las máquinas utilizadas	17
Figura 4. Potencias de dimensionamiento vectorial y aritmética, caso sinusoidal, no balanceado	19
Figura 5. Circuito equivalente por fase del GI trifásico autoexcitado en régimen permanente.	20
Figura 6. Diagrama de flujo general de la plataforma de simulación (usuario cliente)	23
Figura 7. Diagrama de flujo general de la plataforma de simulación (usuario administrador)	23
Figura 8. Diagrama de flujo para la creación de usuario	24
Figura 9. Diagrama de flujo para las prácticas	24
Figura 10. Diagrama de flujo de la práctica teórica 1	24
Figura 11. Diagrama de flujo de la práctica teórica 2	24
Figura 12. Diagrama de flujo para las prácticas de simulación	24
Figura 13. Ventana del menú principal	25
Figura 14. Ejemplo de una ventana de submenú	25
Figura 15. Ventana del menú principal	25
Figura 16. Ejemplo de una ventana de tutoría	26
Figura 17. Ejemplo de una ventana de error	26
Figura 18. Ejemplo de salida: Potencia de la carga eléctrica vs. carga eléctrica	26
Figura 19. Ejemplo de salida: Torque mecánico vs. carga eléctrica	26
Figura 20. Estado actual del Grupo electrógeno del Laboratorio de Máquinas Térmicas Alternativas de la Escuela de Ingeniería Mecánica	28
Figura 21. Esquema de la propuesta para el laboratorio de pruebas	28

LISTA DE TABLAS

	pág.
Tabla 1. Rango y precisión para el medidor de flujo másico de combustible	28
Tabla 2. Rango y precisión para el medidor de flujo volumétrico de combustible	29
Tabla 3. Rango y precisión para el medidor de flujo másico de aire	29
Tabla 4. Rango y precisión para termómetro que medirá la temperatura ambiental	29
Tabla 5. Rango y precisión para termómetro que medirá la temperatura en la admisión	29
Tabla 6. Rango y precisión para termómetro que medirá la temperatura de los gases de escape	29
Tabla 7. Rango y precisión para barómetro	29
Tabla 8. Rango y precisión para el sistema de medición del torque Mecánico	29

ANEXOS

	pág.
ANEXO A. SOLUCIÓN AL SISTEMA DE ECUACIONES DEL ANÁLISIS TRANSITORIO DEL GI TRIFÁSICO	33
ANEXO B. ALGORITMOS DE LOS MODELOS IMPLEMENTADOS	48
ANEXO C. MANUAL DE LA PLATAFORMA DE SIMULACIÓN	73

RESUMEN

TÍTULO: DESARROLLO DE UNA PLATAFORMA DE SIMULACIÓN Y DISEÑO DE UN LABORATORIO DE PRUEBAS PARA GRUPOS ELECTRÓGENOS DE BAJA POTENCIA PARA APLICACIONES EN ZONAS REMOTAS *

AUTOR: PRADILLA RODRÍGUEZ, Juan Camilo**

PALABRAS CLAVE: motor de combustión interna, máquina de inducción, generación de potencia eléctrica, medición de potencia (mecánica y eléctrica), eficiencia de potencia, simulación.

DESCRIPCIÓN

El objetivo fundamental del presente proyecto es simular las conversiones de energía que ocurren dentro de un grupo electrógeno de baja potencia. Otro objetivo importante de este trabajo de grado es presentar una propuesta para la implementación del sistema de medición del grupo electrógeno del laboratorio de Máquinas Térmicas Alternativas de Ingeniería Mecánica.

En el desarrollo de la plataforma de simulación se enfatizó en la conversión de energía para desarrollar los modelos de las máquinas involucradas (motor de combustión interna y máquina eléctrica) y se analizaron cargas similares a las encontradas en el LMTA (Laboratorio de Máquinas Térmicas Alternativas), las cuales son resistivas (con comportamiento lineal) y corresponden con las que comúnmente se pueden encontrar en zonas remotas.

Además, la plataforma de simulación que se expone en el presente trabajo de grado, fue desarrollada con el fin de que los estudiantes de ingenierías Mecánica, Eléctrica y Electrónica, puedan estudiar los conceptos básicos que tienen que ver con las conversiones de energía que ocurren en un grupo electrógeno y también los conceptos fundamentales de la medición de potencia eléctrica; los cuales actualmente son de gran relevancia en la industria.

También, el presente proyecto presenta un manual de la plataforma de simulación en el cual se proponen dos prácticas teóricas y se explica el procedimiento para realizar prácticas de simulación.

* Trabajo De Grado.

** Facultad de Ingenierías Fisicomecánicas, Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Universidad Industrial de Santander. Director: Dr. Jorge Luis Chacón. Codirector: Dr. Gabriel Ordóñez Plata.

ABSTRACT

TITLE: DEVELOPMENT OF A PLATFORM OF SIMULATION AND DESIGN OF A RESEARCH LABORATORY FOR GENERATOR SETS OF LOW POWER FOR APPLICATIONS IN REMOTE ZONES*

AUTHOR: PRADILLA RODRÍGUEZ, Juan Camilo**

KEYWORDS: internal combustion engine, induction machine, electrical power generation, power measurement (electrical and mechanic), power efficiency, simulation.

DESCRIPTION

The main objective of the present project is to simulate the energy conversions that happen within a generator set of low power. Another important objective of this work of degree is to present a proposal for the implementation of the system of measurement of the generator set of the laboratory of Alternative Thermal Machines of Mechanical Engineering.

In the development of the simulation platform it was emphasized in the energy conversion to develop the models of the involved machines (motor of internal combustion and electrical machine) and was analyzed similar charges to those found in the LMTA (Laboratory of Alternative Thermal Machines, for its initials in Spanish), which are resistive (with behavior linear) and correspond to what is commonly be found in remote areas.

Also, the simulation platform that is exposed in the present work of degree, was developed in order that the students of engineering Mechanical, Electrical and Electronic, can study the basic concepts that have to do with the energy conversions also happen in a generator set and the fundamental concepts of the measurement of electric power; concepts that at the moment are of great relevance in the industry.

In addition, this project presents a manual of the simulation platform in which are proposed two theoretical practices and is explained the procedure for conducting simulation exercises.

* Project of Degree

** Faculty of Physics-Mechanical Engineerings. School of Electrical, Electronical and Telecommunications Engineerings. Santander's Industrial University. Director: PhD. Jorge Luis Chacón. Co-director: PhD. Gabriel Ordóñez Plata.

Desarrollo de una Plataforma de Simulación y Diseño de un Laboratorio de Pruebas para Grupos Electrónicos de Baja Potencia para Aplicaciones en Zonas Remotas

Juan Camilo Pradilla Rodríguez

Resumen—El objetivo fundamental del presente artículo es presentar la plataforma de simulación desarrollada para ser utilizada en el Laboratorio de Máquinas Térmicas Alternativas de la escuela de Ingeniería Mecánica, mediante la cual se busca observar y analizar las conversiones de energía que ocurren dentro de un grupo electrógeno de baja potencia. Otro objetivo importante de este artículo es presentar una propuesta para la implementación del sistema de medición del grupo electrógeno del laboratorio mencionado.

Palabras clave—motor de combustión interna, máquina de inducción, generación de potencia eléctrica, medición de potencia (mecánica y eléctrica), eficiencia de potencia, simulación.

NOMENCLATURA

a	Radio de la manivela de un MCI
C_{pf}	Calor específico de un combustible
C_{pm}	Calor específico de una mezcla gaseosa
E_i	Energía interna
F	Fuerza tangencial aplicada a la manivela del MCI
f_m	Frecuencia de giro de la manivela del MCI
GI	Generador de Inducción
I	Intensidad de corriente eléctrica (o simplemente corriente eléctrica)
I_1	Corriente eléctrica del estator de un GI por fase
I_2	Corriente eléctrica del rotor de un GI por fase
I_c	Corriente eléctrica del capacitor externo conectado a un GI en cada fase
I_h	Corriente eléctrica de pérdidas en el entrehierro de un GI por fase
I_m	Corriente eléctrica de magnetización de un GI por fase
$LMTA$	Laboratorio de Máquinas Térmicas Alternativas
MCI	Motor de Combustión Interna
m_a	Masa del aire que entra a un MCI por ciclo
\dot{m}_a	Flujo másico del aire en un MCI
m_f	Masa del combustible que entra a un MCI por ciclo
\dot{m}_f	Flujo másico de combustible en un MCI
$m_f Q_f$	Poder calórico del combustible

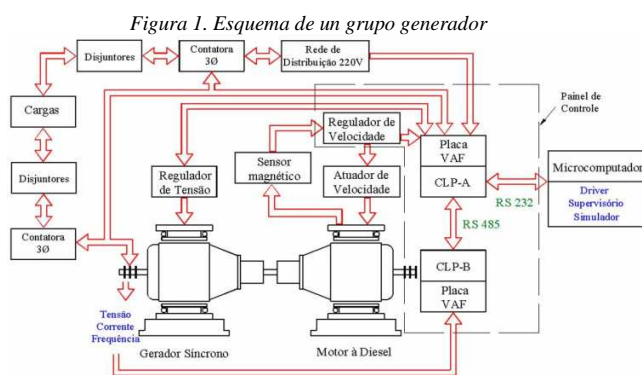
RPM	Revoluciones por minuto de una máquina giratoria
η_m	Eficiencia mecánica del MCI
η_t	Eficiencia termodinámica del MCI
η_T	Eficiencia térmica del MCI
P	Potencia eléctrica activa (valor medio)
$p(t)$	Potencia eléctrica instantánea
P_e	Potencia equivalente (potencia a la salida de un MCI)
P_f	Pérdidas de potencia mecánicas de un MCI
P_i	Potencia indicada del MCI, corresponde a la Potencia teórica menos la potencia de pérdidas térmicas.
P_T	Potencia teórica del MCI (la que entregaría si no tuviera pérdidas)
Q	Potencia eléctrica reactiva (valor medio)
Q_{chem}	Flujo térmico de combustión del MCI
Q_f	Calor de combustión por unidad de masa
\dot{Q}_{ht}	Pérdidas térmicas de potencia del MCI
R_1	Resistencia del estator de un GI por fase
R_2	Resistencia del rotor de un GI por fase
R_{gas}	Constante universal de los gases ideales
R_h	Resistencia de pérdidas en el entrehierro de un GI por fase
s	Deslizamiento del GI
S	Potencia eléctrica de dimensionamiento
V	Tensión eléctrica
V_d	Cilindraje del MCI
X_1	Reactancia del estator de un GI por fase
X_2	Reactancia del rotor de un GI por fase
X_m	Reactancia de magnetización de un GI por fase

INTRODUCCIÓN

Un grupo electrógeno es un sistema que se utiliza para la generación de energía eléctrica; consiste en el acople de un MCI con una máquina eléctrica. Las máquinas eléctricas comúnmente utilizadas para la generación de potencia eléctrica son las síncronas, debido a diferentes factores que no se estudian en el presente trabajo de grado [CHAPMAN, 2005]. Para sistemas de generación de bajas potencias se utilizan comúnmente las

máquinas de inducción, debido a su precio y a su sencillo diseño; además, el MCI utilizado normalmente para este tipo de sistemas es el de encendido por chispa o de ciclo Otto, la razón principal para esto es su bajo precio comparado con otros MCI.

En la figura 1 se puede observar el esquema de un grupo electrógeno. La máquina de la derecha es un MCI (en este caso uno Diesel) y la de la izquierda es una máquina eléctrica (para este ejemplo, se trata de una máquina síncrona). El grupo electrógeno incluye también comúnmente otros elementos como son: un sistema de alimentación con combustible para el MCI y sistemas de control para las dos máquinas. Estos elementos y otros más se pueden observar en la figura 1. Allí puede verse que las dos máquinas están unidas por su eje, esto con el fin de reducir las pérdidas por fricción; también se puede ver que las variables de salida de la máquina eléctrica que funciona como generador son tres: tensión eléctrica (llamada voltaje por los diseñadores del esquema), corriente y frecuencia; de este modo el cálculo de la potencias eléctricas del sistema se realiza mediante fórmulas que contienen estos tres parámetros.



Fuente: <http://www.ece.ufrgs.br/~gmasp/espanhol/plataforma3.jpg>

El principio de funcionamiento de un grupo electrógeno desde el punto de vista de la transformación de energía (en el que se centra el presente proyecto) es el siguiente: la energía química del combustible es liberada en forma de calor mediante su combustión, a continuación parte de esta energía liberada es transformada en energía mecánica a través de la expansión de la mezcla aire – combustible en los cilindros del MCI y luego parte de esta última energía es transformada en energía eléctrica. La razón por la cual en cada proceso no se transforma toda la energía química en energía eléctrica es debido a que en cada transformación se producen pérdidas de energía originadas por diferentes fenómenos; las pérdidas principales son:

- En el MCI: pérdidas de calor por escape, conducción y radiación, pérdidas por fricción.
- En la máquina eléctrica: pérdidas en los conductores y pérdidas por corrientes parásitas en el hierro y entrehierro.

Mediante la simulación de un grupo electrógeno de baja potencia se busca proveer (al laboratorio de Máquinas Térmicas Alternativas de la Escuela de Ingeniería Mecánica) de una herramienta didáctica para desarrollar conocimientos y competencias que tienen que ver con la interacción de máquinas térmicas con máquinas eléctricas; conocimientos y habilidades

específicamente relacionados con la conversión de energía que ocurre en estos sistemas. Además se propone un sistema de medición que permita calcular la eficiencia del grupo electrógeno del mencionado laboratorio con una precisión adecuada a los propósitos y recursos de dicho laboratorio. También, es importante enfatizar que el estudio de estos sistemas es muy importante debido a su aplicación en zonas remotas en las cuales los combustibles son fácilmente accesibles e incluso se pueden producir in situ, como es el caso del biogas.

En el numeral 1 se presentan los conceptos fundamentales que sustentan el desarrollo de este proyecto, los cuales son: motores de combustión interna (se hace énfasis en el cálculo de la eficiencia energética de los MCI de ciclo Otto), el funcionamiento de las máquinas eléctricas como generadores (se enfatiza el generador de inducción (GI) y la medición de su eficiencia energética).

En el numeral 2 se exponen los modelos utilizados para desarrollar la plataforma de simulación: el modelo de la carga eléctrica (lineal), el modelo de la medición de la potencia eléctrica, el modelo del GI y el modelo del MCI de ciclo Otto.

En el numeral 3 se explica el proceso seguido para el desarrollo de la plataforma de simulación, es decir, los requisitos, el diseño, la implementación y las pruebas realizadas a esta. Además se describe el manual de laboratorio propuesto para ser utilizado con la plataforma de simulación, el cual se presenta en el anexo C.

El numeral 4 es el informe de la investigación realizada para entregar una propuesta de acondicionamiento del sistema de medición de la eficiencia energética del grupo electrógeno del Laboratorio de Máquinas Térmicas Alternativas de la Escuela de Ingeniería Mecánica.

1. TRANSFORMACIONES DE ENERGÍA EN UN GRUPO ELECTRÓGENO

A continuación se explica el funcionamiento de los motores de combustión interna desde el punto de vista de la potencia, para luego pasar a explicar la forma de medir la eficiencia de dichos motores. Luego se explica el funcionamiento de los generadores eléctricos y el cálculo de su eficiencia, para pasar entonces a exponer los conceptos claves de la medición de la potencia eléctrica.

1.1 MOTOR DE COMBUSTIÓN INTERNA

Los motores de combustión interna (MCI) son muy utilizados, principalmente en los campos del transporte y la generación de potencia eléctrica, debido a su simplicidad, robustez y alta relación potencia/peso.

Básicamente, un MCI es una máquina que es construida con el propósito de producir energía mecánica a partir de la energía química almacenada en un combustible. Esto ocurre mediante la ignición del combustible que ha sido ubicado dentro del motor el proceso de ignición libera la energía química en forma de energía calorífica; entonces, el calor expande la mezcla aire combustible dentro del motor, transformando parte de dicha energía en energía mecánica.

1.1.1 Medición de la eficiencia de un MCI de ciclo Otto

En orden de prioridad, la eficiencia de un MCI depende de las siguientes características: el método de ignición, el combustible utilizado, la forma de preparación de la mezcla, el diseño de la cámara de combustión, el control de carga, los detalles del proceso de combustión, la emisión, las características de operación y el ciclo de trabajo.

El método de ignición que se utiliza en los MCI utilizados en generación eléctrica de baja potencia es, comúnmente, el de chispa (ciclo Otto), debido a que estos motores son más pequeños que los de ignición por compresión, a pesar de que estos últimos son más eficientes que los de ignición por chispa.

Ahora, los combustibles que se emplean comúnmente en los MCI utilizados para la generación eléctrica de baja potencia son el gas natural, la gasolina y el biogas.

Para comenzar el análisis de eficiencia de un MCI de ciclo Otto, se obtiene la potencia teórica del mismo, la cual se define como la potencia que el combustible entregaría al motor si no existieran pérdidas y se expresa de la siguiente manera:

$$P_T = Q_{HT} \dot{m}_f \quad (1)$$

Donde:

Q_{HT} = poder calórico del combustible en J/kg

\dot{m}_f = consumo de combustible del MCI en kg/s.

Ahora, para el análisis que se realiza a continuación se lleva el MCI de ciclo Otto real a un MCI equivalente de un solo cilindro y un solo pistón.

De este modo, la potencia que queda disponible luego de que el sistema sufre las pérdidas por disipación del calor en el calentamiento del refrigerante y a través de los gases de escape, se conoce como potencia indicada y se expresa de la siguiente manera:

$$P_{ig} = \frac{mepVN}{n_R} \quad (2)$$

Donde:

mep = presión efectiva media en el pistón del MCI en N/m²

V = volumen del cilindro en m³

N = velocidad rotacional del cigüeñal en rev/s

n_R = constante, 1 si el MCI es de 2 tiempos, o, 2 si el MCI es de 4 tiempos

A partir de las dos potencias anteriores se puede definir la eficiencia termodinámica del MCI de ciclo Otto, la cual se expresa de la siguiente manera:

$$\eta_t = \frac{P_{ig}}{P_T} \quad (3)$$

Además de las pérdidas térmicas, existen también pérdidas mecánicas en el MCI de ciclo Otto debidas al rozamiento en los pistones, en las bielas, en el cigüeñal, en la bomba de refrigerante, en el alternador, al trabajo de bombeo, etc. La

potencia que finalmente es entregada al eje de rotación del MCI es conocida entonces como potencia equivalente y su expresión aparece en la fórmula 4.

$$P_b = 2\pi aFN \quad (4)$$

Donde:

a es el radio de la manivela

F es la fuerza tangencial aplicada a la manivela

De las potencias definidas en las fórmulas 2 y 4 podemos obtener la expresión de la eficiencia mecánica del MCI la cual aparece en la fórmula 5.

$$\eta_m = \frac{P_b}{P_{ig}} \quad (5)$$

Ahora bien, la eficiencia total será entonces el producto de las eficiencias definidas en las fórmulas 3 y 5. De modo que la expresión de la eficiencia total será como sigue:

$$\eta_T = \eta_t \eta_m = \frac{P_b}{P_T} \quad (6)$$

Una explicación más profunda del funcionamiento del MCI se puede encontrar en [BARBOSA, SALCEDO, 2007] y en [Heywood John B, 1988].

1.2 FUNCIONAMIENTO DE LAS MÁQUINAS ELÉCTRICAS COMO GENERADORES.

Cuando una máquina eléctrica es utilizada como generador, ocurre que (bajo ciertas condiciones bien definidas) al ser la máquina impulsada por una fuerza externa transforma la energía mecánica de su movimiento en energía eléctrica [CHAPMAN, 2005]. Toda máquina eléctrica está compuesta por dos partes principales; la parte móvil llamada rotor y la parte fija conocida como estator. La presencia de un campo magnético variable alrededor del bobinado del rotor o el movimiento de este bobinado en presencia de un campo magnético constante induce una corriente en dicho bobinado, de modo tal que la energía mecánica se transforma en energía eléctrica.

Existen varios tipos de máquinas eléctricas dependiendo de la forma en que se realizan las interacciones entre los campos. Para este proyecto se seleccionó la máquina de inducción (que se conoce como generador de inducción cuando funciona como tal), debido a que es la que utiliza el grupo electrógeno del Laboratorio de Máquinas Térmicas Alternativas de la Escuela de Ingeniería Mecánica y también porque es el tipo de motor utilizado comúnmente, para aplicaciones de baja potencia, en la generación eléctrica de corriente alterna.

1.2.1 Generador de inducción (GI)

La máquina de inducción se diseñó originalmente para ser utilizada como motor, debido a que requiere poco mantenimiento, es muy robusto y tiene bajo costo en comparación con otro tipo de motores. Estas ventajas también pueden ser aprovechadas para utilizar esta máquina como generador, pero se deben tener en cuenta ciertas características extra en el diseño en comparación con el diseño como motor. La característica principal de la máquina de inducción cuando se

diseña como motor es que no necesita de un campo magnético externo a las bobinas para funcionar, puesto que las bobinas generan el campo magnético que, junto con la corriente que pasa por estas, produce la fuerza necesaria para mover el rotor.

Pero, es precisamente esta característica la que hace necesaria la inclusión de elementos extra cuando se diseña la máquina de inducción como generador eléctrico, puesto que si se intenta utilizar un motor de inducción como generador, al no existir un campo magnético presente, no se produce electricidad al mover dicha máquina.

Entonces, para que la máquina de inducción pueda utilizarse como generador de energía eléctrica, es necesario adicionarle un circuito que produzca el campo magnético necesario para que pueda inducir electricidad [SIMÕES, 2004]. Esto se consigue adicionando un banco de condensadores, los cuales producen la corriente magnetizante. También es preciso que haya algo de remanencia en el hierro del rotor, es decir, algo de magnetismo restante, cuando se ponga en marcha la turbina; cuando el GI ha perdido el magnetismo remanente es necesario producirlo, esto se logra mediante un sistema de arranque que puede ser una batería conectada a un circuito de electrónica de potencia o un pequeño generador. El sistema de arranque puede ser desconectado cuando el magnetismo producido por la corriente de arranque comienza a incrementarse gracias a las corrientes provenientes del banco de condensadores; el tiempo entre la conexión y la desconexión del sistema de arranque es calculado en el diseño, mediante pruebas realizadas al sistema. Cabe aclarar que es común que el sistema de arranque sea desconectado del sistema luego de realizar su función, en cambio, el banco de capacitores comúnmente permanece conectado todo el tiempo.

1.2.1.1 Eficiencia de un GI.

En un GI existen ciertas pérdidas que tienen su origen en la disipación del calor por efecto Joule de las corrientes en los conductores de las bobinas, en el hierro y entrehierro además de efectos mecánicos (principalmente, el rozamiento). Estas pérdidas pueden ser calculadas de forma experimental conectando el generador como motor y realizando una prueba de vacío (sin carga), de tal manera que la potencia consumida por el motor en ese estado será la debida a las pérdidas.

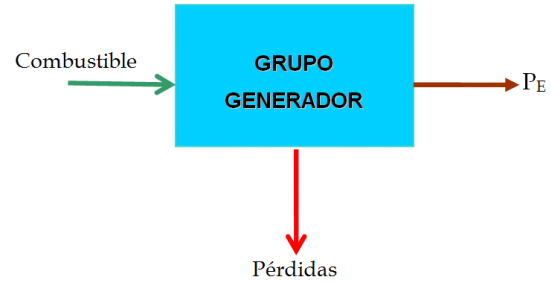
2. DESCRIPCIÓN DE LOS MODELOS MATEMÁTICOS DEL GRUPO ELECTRÓGENO

Partiendo del modelo más general (en el que se considera el grupo generador como una caja negra) es posible ir aumentando en detalle y complejidad hasta lograr un modelo robusto. El modelo más general de un grupo generador formado por un MCI y un GI, desde el punto de vista de la transformación de energía es un sistema cuya entrada es el combustible del MCI (no necesariamente se define un límite para la cantidad de combustible inicial del modelo, aunque esto puede ser útil para ciertos cálculos) y cuyas salidas son la potencia eléctrica y las pérdidas del sistema, como aparece en la figura 2.

La segunda aproximación realizada para modelar el grupo electrógeno es dividir este en sus componentes motores: el MCI y el GI, de esta manera, cada máquina puede verse como una

caja negra que interacciona con la otra como puede verse en la figura 3. En este caso aparece una variable intermedia que es la potencia equivalente (P_e) y se discriminan las pérdidas de cada motor.

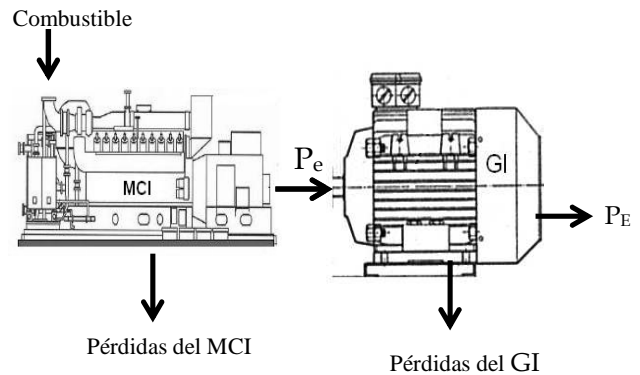
Figura 2. Modelo general del grupo generador



Fuente: el autor

Ahora, cada uno de los bloques de este modelo será analizado de forma independiente. El orden siguiente viene dado debido a que en un grupo generador, la potencia total depende de la carga eléctrica, de modo tal que se parte de la suposición de que la tensión eléctrica en la carga tiene un valor RMS constante (el sistema asegura esta condición mediante algún método cuyo estudio se sale de los alcances del presente proyecto), además también se considera que la velocidad del motor se mantiene en un valor constante y algo superior a la velocidad síncrona del GI.

Figura 3. Modelo del grupo generador que discrimina las máquinas utilizadas



Fuente: el autor

2.1 MODELO DE LA CARGA ELÉCTRICA

Debido a que la carga considerada es lineal y resistiva, el modelo consiste en definir unos valores de resistencia que definen directamente los valores de V_L y de I_L según la ley de Ohm:

$$V_L = I_L R_L \quad (7)$$

2.2 MODELOS DE MEDICIÓN DE LA POTENCIA ELÉCTRICA

En este caso, se implementaron algoritmos para realizar el cálculo de las diferentes potencias, utilizando las fórmulas propuestas por la norma IEEE [IEEE 1459 – 2000] [BAUTISTA, 2007], tomando los casos de estudio de la norma aplicados a cargas lineales y para modelos lineales del sistema: sistemas monofásicos, sistemas trifásicos balanceados y los sistemas trifásicos desbalanceados; en los tres casos sin armónicos, es decir, las formas de onda analizadas son sinusoidales. A continuación se presentan las fórmulas aplicadas para la simulación, tanto para sistemas monofásicos como para sistemas trifásicos.

2.2.1 Modelo de la medición de potencia en sistemas monofásicos sinusoidales

Para este modelo se definen las siguientes magnitudes:

Tensión:

$$v = \sqrt{2}V_{RMS} \text{sen}(\omega t + \alpha) \quad (8)$$

Corriente:

$$i = \sqrt{2}I_{RMS} \text{sen}(\omega t + \beta) \quad (9)$$

Potencia instantánea:

$$p = v i \quad (10)$$

Potencia activa promedio:

$$P = V I \cos \theta ; \theta = \alpha + \beta \quad (11)$$

Potencia reactiva promedio:

$$Q = V I \text{sen} \theta \quad (12)$$

Potencia de dimensionamiento:

$$S = V I \quad (13)$$

2.2.2 Modelo de la medición de potencia en sistemas trifásicos sinusoidales balanceados

Tensiones de fase:

$$v_a = \sqrt{2}V_{atn} \text{sen}(\omega t + \alpha_a) \quad (14)$$

$$v_b = \sqrt{2}V_{btn} \text{sen}(\omega t + \alpha_b - 120^\circ) \quad (15)$$

$$v_c = \sqrt{2}V_{ctn} \text{sen}(\omega t + \alpha_c + 120^\circ) \quad (16)$$

Donde V_{atn} , V_{btn} , V_{ctn} son los valores RMS de las tensiones de fase.

Para el caso de sistemas trifásicos trifilares, las tensiones de línea a neutro se definen asumiendo un nodo neutro artificial.

Las corrientes de línea se expresan de forma semejante:

$$i_a = \sqrt{2}I_a \text{sen}(\omega t + \beta_a) \quad (17)$$

$$i_b = \sqrt{2}I_b \text{sen}(\omega t + \beta_b - 120^\circ) \quad (18)$$

$$i_c = \sqrt{2}I_c \text{sen}(\omega t + \beta_c + 120^\circ) \quad (19)$$

Por lo tanto la potencia instantánea es:

$$p = v_a i_a + v_b i_b + v_c i_c \quad (20)$$

De la misma manera, la potencia activa promedio trifásica se define como:

$$P = P_a + P_b + P_c \quad (21)$$

Donde P_a , P_b , y P_c son las potencias activas de cada fase y se definen de la siguiente manera:

$$P_a = V_{atn} I_a \cos \theta_a ; \theta_a = \alpha_a + \beta_a \quad (22)$$

$$P_b = V_{btn} I_b \cos \theta_b ; \theta_b = \alpha_b + \beta_b \quad (23)$$

$$P_c = V_{ctn} I_c \cos \theta_c ; \theta_c = \alpha_c + \beta_c \quad (24)$$

Ahora, las potencias reactivas por fase se expresan así:

$$Q_a = V_{atn} I_a \text{sen} \theta_a \quad (25)$$

$$Q_b = V_{btn} I_b \text{sen} \theta_b \quad (26)$$

$$Q_c = V_{ctn} I_c \text{sen} \theta_c \quad (27)$$

Ahora la potencia de dimensionamiento por fase es como sigue:

$$S_a = V_{atn} I_a \quad (28a) \quad S_b = V_{btn} I_b \quad (28b) \quad S_c = V_{ctn} I_c \quad (28c)$$

La potencia de dimensionamiento del sistema trifásico es:

$$S = S_a + S_b + S_c \quad (29)$$

2.2.3 Modelo de la medición de potencia en sistemas trifásicos sinusoidales desbalanceados

Tensiones de fase:

$$v_a = \sqrt{2}V_{atn} \text{sen}(\omega t + \alpha_a) \quad (30)$$

$$v_b = \sqrt{2}V_{btn} \text{sen}(\omega t + \alpha_b - 120^\circ) \quad (31)$$

$$v_c = \sqrt{2}V_{ctn} \text{sen}(\omega t + \alpha_c + 120^\circ) \quad (32)$$

Las corrientes de línea se expresan de forma semejante:

$$i_a = \sqrt{2}I_a \text{sen}(\omega t + \beta_a) \quad (33)$$

$$i_b = \sqrt{2}I_b \text{sen}(\omega t + \beta_b - 120^\circ) \quad (34)$$

$$i_c = \sqrt{2}I_c \text{sen}(\omega t + \beta_c + 120^\circ) \quad (35)$$

Por lo tanto la potencia instantánea es:

$$p = v_a i_a + v_b i_b + v_c i_c \quad (36)$$

De la misma manera, la potencia activa promedio trifásica se define como:

$$P = P_a + P_b + P_c \quad (37)$$

Donde P_a , P_b , y P_c son las potencias activas de cada fase y se definen de la siguiente manera:

$$P_a = V_{atn} I_a \cos \theta_a ; \theta_a = \alpha_a + \beta_a \quad (38)$$

$$P_b = V_{btn} I_b \cos \theta_b ; \theta_b = \alpha_b + \beta_b \quad (39)$$

$$P_c = V_{ctn} I_c \cos \theta_c ; \theta_c = \alpha_c + \beta_c \quad (40)$$

Ahora, las potencias reactivas por fase se expresan así:

$$Q_a = V_{atn} I_a \sin \theta_a \quad (41)$$

$$Q_b = V_{btn} I_b \sin \theta_b \quad (42)$$

$$Q_c = V_{ctn} I_c \sin \theta_c \quad (43)$$

Para hallar la potencia de dimensionamiento vectorial (ver la fórmula 92), la potencia reactiva total Q es como sigue:

$$Q = Q_a + Q_b + Q_c \quad (44)$$

Como en el caso de la potencia activa, la utilización de las componentes simétricas puede ser útil en el análisis de la potencia reactiva. Las componentes simétricas de la potencia reactiva son las siguientes:

Ahora la potencia de dimensionamiento por fase es como sigue:

$$S_a = V_{atn} I_a \quad (45a) \quad S_b = V_{btn} I_b \quad (45b) \quad S_c = V_{ctn} I_c \quad (45c)$$

La potencia de dimensionamiento aritmética del sistema trifásico es:

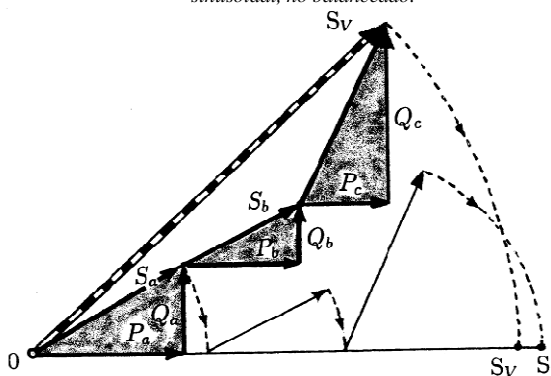
$$S_A = S_a + S_b + S_c \quad (46)$$

En cambio, la potencia de dimensionamiento vectorial se define como sigue:

$$S_V = \sqrt{P^2 + Q^2} \quad (47)$$

En la figura 4 se ve una comparación gráfica entre la potencia de dimensionamiento vectorial y la aritmética.

Figura 4 Potencias de dimensionamiento vectorial y aritmética, caso sinusoidal, no balanceado.



Fuente: Norma IEEE 1459 de 2000

Otro concepto importante que aparece en la norma IEEE 1459 es el de potencia de dimensionamiento efectiva. Este concepto asume un circuito balanceado virtual el cual tiene exactamente las mismas pérdidas de potencia que el circuito desbalanceado. Es claro que el sistema balanceado virtual tendrá una corriente de línea equivalente I_e y una tensión de fase equivalente V_e . De este modo la potencia de dimensionamiento efectiva es definida de la siguiente manera:

$$S_e = 3V_e I_e \quad (48)$$

Las expresiones para V_e e I_e son diferentes dependiendo de si el sistema trifásico es tetrafilar o trifilar, así que es necesario exponer cada caso:

Sistema tetrafilar:

En este caso, la tensión equivalente se halla mediante la siguiente expresión:

$$V_e = \sqrt{\frac{1}{18} [3(V_a^2 + V_b^2 + V_c^2) + V_{ab}^2 + V_{bc}^2 + V_{ca}^2]} \quad (49)$$

Ahora, para este tipo de sistemas la corriente equivalente se halla de la siguiente manera:

$$I_e = \sqrt{\frac{I_a^2 + I_b^2 + I_c^2 + I_n^2}{3}} \quad (50)$$

Sistema trifilar:

Para estos sistemas, la tensión equivalente se expresa de la siguiente manera:

$$V_e = \sqrt{\frac{V_{ab}^2 + V_{bc}^2 + V_{ca}^2}{9}} \quad (51)$$

Y la corriente equivalente es definida como sigue:

$$I_e = \sqrt{\frac{I_a^2 + I_b^2 + I_c^2}{3}} \quad (52)$$

De esta forma, aparece otro concepto el cual es el de factor de potencia efectivo, que se expresa así:

$$P_{Fe} = \frac{P}{S_e} \quad (53)$$

Se define la potencia de desbalance de la siguiente manera:

$$S_U = \sqrt{S_e^2 - (S^+)^2} \quad (54)$$

Donde S^+ es la potencia de dimensionamiento de secuencia positiva, la cual está definida de la siguiente manera:

$$S^+ = V^+ I^+ \quad (55)$$

2.3 MODELOS DEL GI

Para esta máquina, se analizaron los modelos para generadores trifásicos y monofásicos en régimen permanente y transitorio.

Lo que se pretende con este modelo es determinar la evolución temporal de las tensiones y corrientes del GI en los dos intervalos de tiempo que aparecen a continuación:

1. Conexión, es decir, desde que el GI se pone en movimiento, hasta que éste alcanza el funcionamiento en régimen permanente. Esta simulación se puede realizar bajo dos condiciones iniciales diferentes:

a. A partir de condiciones iniciales nulas (excepto la velocidad de la máquina que debe ser superior a la de sincronismo). En este caso se supone la aplicación de una tensión en el estator dada por un sistema de arranque auxiliar (que en algunos grupos generadores puede ser una línea de transmisión, en otros un pequeño generador y en varios consiste en un sistema formado por una batería y un inversor).

b. Con la máquina energizada (una reconexión luego de haber funcionado en régimen permanente durante cierto tiempo). Es decir, las condiciones iniciales no son nulas. Por lo tanto no se utiliza el sistema de arranque auxiliar.

2. Desconexión, es decir desde que se apaga el MCI que lo impulsa, hasta que la tensión residual alcanza un valor nulo. El modelo para esta condición consiste en suponer que al apagarse el MCI el torque mecánico entregado por este al GI cae a cero instantáneamente y el torque debido al rozamiento en el eje compartido por el MCI y el GI se desprecia, aunque su valor puede ser agregado al modelo si se desea aumentar la precisión del mismo.

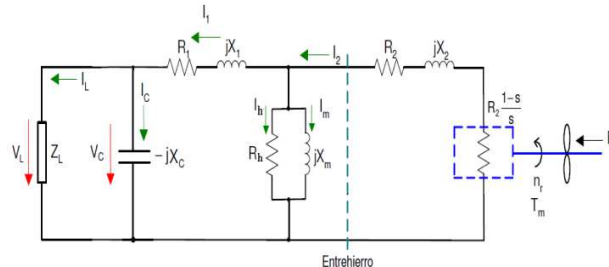
Ahora bien, para determinar los transitorios de tensión y corriente en los GI analizados en el presente proyecto, se hallaron las ecuaciones diferenciales de cada lazo. Para simplificar el análisis, estas ecuaciones diferenciales se expresan en el dominio de s .

A continuación se presenta el análisis de cada modelo.

2.3.1 Modelo del GI trifásico en régimen permanente

Este primer caso analizado, corresponde al estado del GI en el cual el arranque ya se ha realizado y la corriente de magnetización ya se ha producido. Este modelo puede ser expresado mediante un circuito equivalente [GUERRERO CUESTA, 2006]. El circuito equivalente por fase del GI trifásico autoexcitado, con todos los parámetros referidos al estator y a la frecuencia eléctrica puede ser observado en la figura 5.

Figura 5. Circuito equivalente por fase del GI trifásico autoexcitado en régimen permanente.



Fuente: GUERRERO CUESTA, William Edixón. Estudio de la variación del voltaje y la frecuencia de generación de una máquina de inducción funcionando como generador. Capítulo 1, Páginas 2 y 15.

Donde:

- I_1 = corriente del estator por fase
- R_1 = resistencia del estator por fase
- X_1 = reactancia del estator por fase (representa el flujo de salida del estator)
- I_m = corriente de magnetización por fase (crea el flujo en el entrehierro)
- I_h = corriente de pérdidas en el entrehierro por fase
- I_c = corriente del capacitor externo de cada fase
- R_h = resistencia de pérdidas en el entrehierro por fase
- X_m = reactancia de magnetización por fase
- I_2 = corriente del rotor por fase
- R_2 = resistencia del rotor por fase
- X_2 = reactancia del rotor por fase (representa el flujo de salida del rotor)
- Z_L = impedancia de la carga por fase
- P_e = potencia equivalente por fase, es decir, 1/3 de la potencia equivalente del MCI.
- T_m es el torque mecánico de entrada por fase
- $s = \frac{f_s - f_r}{f_s}$ = deslizamiento del GI. Donde:
- f_s = frecuencia sincrónica
- f_r = frecuencia del rotor.

El valor de la frecuencia del rotor depende del número de polos del GI, según la siguiente expresión:

$$f_r = \frac{2f_e}{p} k_f \quad (56)$$

Donde:

- p = número de polos del GI.
- k_f = factor arbitrario, pero que debe ser mayor que uno para asegurar que se produzca la generación eléctrica.

En la figura 5, aparece una expresión con unidades de resistencia (R_T), la cual representa la transformación de la energía mecánica en energía eléctrica. Esta expresión es definida a continuación:

$$R_T = \frac{R_2(1-s)}{s} \quad (57)$$

De esta forma se determina la corriente I_2 y la tensión en R_T (V_T), a partir de las siguientes expresiones:

$$P_e = I_2^2 R_T \quad (58)$$

$$P_e = \frac{V_T^2}{R_T} \quad (59)$$

De modo tal que, las magnitudes de la corriente I_2 y la tensión V_T se expresan así:

$$|I_2| = \sqrt{\frac{P_e}{R_T}} \quad (60)$$

$$|V_T| = \sqrt{R_T P_e} \quad (61)$$

Además, luego de un análisis de nodos se puede notar que la corriente de la carga es:

$$I_L = I_2 - I_h - I_m - I_c \quad (62)$$

Donde:

$$I_h = \left| \frac{V_T - I_2(R_2 + jX_2)}{R_h} \right| \quad (63)$$

$$I_m = \left| \frac{V_T - I_2(R_2 + jX_2)}{jX_m} \right| \quad (64)$$

$$I_c = \left| \frac{V_L}{-jX_c} \right| \quad (65)$$

La expresión para V_L es la siguiente:

$$V_L = I_L R_L \quad (66)$$

Luego, reemplazando V_L de la fórmula 66 en la fórmula 65 y si la expresión resultante la aplicamos en la fórmula 62, se obtiene la siguiente expresión para I_L :

$$I_L = \frac{I_2 - I_h - I_m}{1 + j \frac{R_L}{X_c}} \quad (67)$$

2.3.2. Modelo del GI trifásico en régimen transitorio

Al hallar las ecuaciones diferenciales de los lazos de la figura 5, se obtiene el modelo por fase en régimen transitorio para un GI trifásico. A continuación se exponen las ecuaciones (en el dominio de s) de este modelo:

$$V_T = R_2 I_h + R_2 I_m + R_2 I_c + R_2 I_L + sL_2 I_h + sL_2 I_m + sL_2 I_c + sL_2 I_L + R_h I_h \quad (68)$$

$$V_T = R_2 I_h + R_2 I_m + R_2 I_c + R_2 I_L + sL_2 I_h + sL_2 I_m + sL_2 I_c + sL_2 I_L + sL_m I_m \quad (69)$$

$$V_T = R_2 I_h + R_2 I_m + R_2 I_c + R_2 I_L + sL_2 I_h + sL_2 I_m + sL_2 I_c + sL_2 I_L + R_1 I_c + R_1 I_L + sL_1 I_c + sL_1 I_L + \frac{I_c}{sC} \quad (70)$$

$$V_T = R_2 I_h + R_2 I_m + R_2 I_c + R_2 I_L + sL_2 I_h + sL_2 I_m + sL_2 I_c + sL_2 I_L + R_1 I_c + R_1 I_L + sL_1 I_c + sL_1 I_L + R_L I_L \quad (71)$$

Restando la fórmula 69 de la fórmula 68:

$$R_h I_h - sL_m I_m = 0 \quad (72)$$

Restando la fórmula 71 de la fórmula 70 y reagrupando los términos se obtiene:

$$\frac{I_c}{sC} - R_L I_L = 0 \quad (73)$$

Restando la fórmula 70 de la fórmula 68 se obtiene:

$$R_1 I_c + sR_1 L_1 I_L I_c + sL_1 I_L + R_L I_L - sL_m I_m = 0 \quad (74)$$

La solución de este sistema de ecuaciones aparece en el anexo A.

2.3.3 Modelo del GI monofásico en régimen permanente

El circuito equivalente del GI monofásico es idéntico al circuito equivalente del GI trifásico, excepto por la fuente de potencia, en la cual la corriente (producida por la interacción del movimiento del rotor con el campo magnético) es de la siguiente forma [ALLER 2008]:

$$i = I_m \cos(\omega t) \cos \theta \quad (75)$$

Donde:

θ es el ángulo de la bobina del rotor respecto a la bobina del estator.

2.3.4 Modelo del GI monofásico en régimen transitorio

De la misma manera que con el modelo en régimen permanente, en el régimen transitorio del GI monofásico se aplicarán las mismas fórmulas que para el GI trifásico, teniendo en cuenta que la corriente de la fuente es como aparece en la ecuación 75.

2.4 MODELO DEL MCI DE CICLO OTTO

Aunque para calcular los datos correspondientes a este bloque de la simulación de un grupo electrógeno se pueden utilizar los datos de salida del simulador SIMTA 1.0 [BARBOSA, SALCEDO, 2007], también se consideró necesario implementar un modelo propio para la plataforma de simulación, manteniendo así su independencia de otros sistemas desarrollados, con el fin de facilitar su utilización por parte de los usuarios.

Desde el punto de vista de la conversión de potencia, el modelo del MCI consiste en cuatro bloques: el modelo térmico de combustión, el modelo de las pérdidas térmicas, el modelo de las pérdidas mecánicas y el modelo de la potencia indicada.

A continuación se exponen cada uno de los bloques del modelo del MCI.

2.4.1 Flujo másico de aire

La fórmula utilizada para determinar este parámetro es la siguiente:

$$\dot{m}_a = \eta_v \frac{V_d}{n_R} \rho_{a,i} \frac{N}{60} \quad (76)$$

Donde:

η_v = eficiencia volumétrica del MCI, se asume un valor de 0,8 para este parámetro, pero puede editarse en la plataforma de simulación

V_d = cilindrada del MCI en cm^3

n_R = constante que vale 1 si el MCI es de 2 tiempos y 2 si es de 4 tiempos

$\rho_{a,i}$ = densidad del aire en el ambiente

N = velocidad del MCI en rpm.

2.4.2 Modelo de pérdidas mecánicas

Este bloque tiene como entrada la potencia equivalente y como salida la potencia indicada.

Se sabe que las pérdidas mecánicas del MCI tienen varias fuentes, las principales son: el rozamiento del pistón con el cilindro (disminuido por el lubricante del motor), el rozamiento en las bielas, en el cigüeñal, en la bomba de agua y en el alternador; además está la energía utilizada para el trabajo de bombeo.

Las pérdidas mecánicas del MCI son directamente proporcionales a la velocidad del mismo. Debido a la complejidad de los procesos que intervienen en la aparición de pérdidas mecánicas, para efectos prácticos, en este proyecto se utilizó una fórmula experimental [HEYWOOD. 1988], la cual puede ser corregida para adaptarse a las características de cada MCI que se desee simular.

De este modo la fórmula utilizada para hallar la potencia de pérdidas mecánicas (P_{fr}) es:

$$P_{fr} = V_d N \frac{0,97 + 0,15 \left(\frac{N}{1000} \right) + 0,05 \left(\frac{N}{1000} \right)^2}{10n_R} \quad (77)$$

Donde:

N = velocidad del MCI en revoluciones por minuto

n_R = vueltas por ciclo del cigüeñal, en el caso del MCI de 4 tiempos n_R es igual a 2, e igual a 1 para el motor de 2 tiempos.

V_d = cilindrada del MCI. .

2.4.3 Potencia teórica y modelo de pérdidas térmicas

La entrada de este bloque es la potencia indicada y la salida es la potencia teórica. La expresión que relaciona estas dos potencias es como sigue:

$$P_T = P_{ig} + \dot{Q}_{HT} = P_b + P_{fr} + \dot{Q}_{HT} \quad (78)$$

Donde:

\dot{Q}_{HT} = pérdidas térmicas.

Ahora, la potencia indicada se puede expresar en términos de la potencia teórica de la siguiente manera:

$$P_{ig} = P_T \eta_i \quad (79)$$

Donde:

$$\eta_i = \frac{1}{\text{sfc } Q_{HT}} \quad (80)$$

Es la eficiencia térmica del MCI.

Donde:

sfc = consumo específico de combustible, al cual se le da, en la plataforma, un valor por omisión de 200 g/kWh, siendo los valores típicos entre 180 y 300 g/kWh.

Q_{HT} = poder calórico del combustible. Para la gasolina este parámetro tiene un valor típico de 44 000 kJ/g y para el biogás (metano) tiene un valor típico de 50 000 kJ/g. Por supuesto, estos son los valores que la plataforma usa por omisión, pero pueden ser editados por el usuario.

Luego, aplicando la ecuación (79) en la ecuación (78) resulta:

$$\dot{Q}_{HT} = (P_b + P_{fr}) \left(\frac{1}{\eta_i} - 1 \right) \quad (81)$$

2.4.4 Flujo másico de combustible

La potencia teórica puede expresarse de la siguiente manera:

$$P_T = Q_{HT} \dot{m}_f \quad (82)$$

Donde:

\dot{m}_f = flujo másico de combustible.

De esta manera se puede determinar el flujo másico de combustible mediante la siguiente expresión:

$$\dot{m}_f = \frac{P_T}{Q_{HT}} \quad (83)$$

3. ELABORACIÓN DE LA PLATAFORMA DE SIMULACIÓN

En esta sección, se exponen los requisitos, el análisis, el diseño, la implementación y las pruebas realizadas a la plataforma de simulación; además de presentar la propuesta del autor respecto al manual de laboratorio que será utilizado junto con la plataforma de simulación implementada para realizar la práctica del Laboratorio de Máquinas Térmicas Alternativas de la Escuela de Ingeniería Mecánica.

3.1 REQUISITOS

El presente proyecto tiene como objetivo fundamental el desarrollo de una plataforma de simulación del funcionamiento de los grupos electrógenos de baja potencia desde el punto de vista de la conversión de energía en cada uno de sus componentes, de modo tal que las personas interesadas en este tipo de sistemas (primeramente los estudiantes de Ingeniería Mecánica que estén cursando la asignatura Máquinas Térmicas Alternativas y también aquellos estudiantes de la Escuela de Ingeniería Eléctrica, Electrónica y Telecomunicaciones, que deseen observar el efecto que tiene sobre estos sistemas la carga eléctrica lineal y no lineal) puedan analizar los procesos de transformación de energía y las diferentes pérdidas que se presentan en los grupos electrógenos para observar su eficiencia y poder analizar cómo pueden ser reducidas las pérdidas de energía.

Por lo tanto, los procedimientos fundamentales para el funcionamiento de la plataforma de simulación son los siguientes:

- 1) Simular la conversión de energía del grupo generador cuando este es conectado a una carga eléctrica lineal.
- 2) Simular la conversión de energía del grupo electrógeno bajo la condición de tener una carga eléctrica no lineal conectada a este.

Ahora bien, las funcionalidades de la plataforma de simulación son las siguientes:

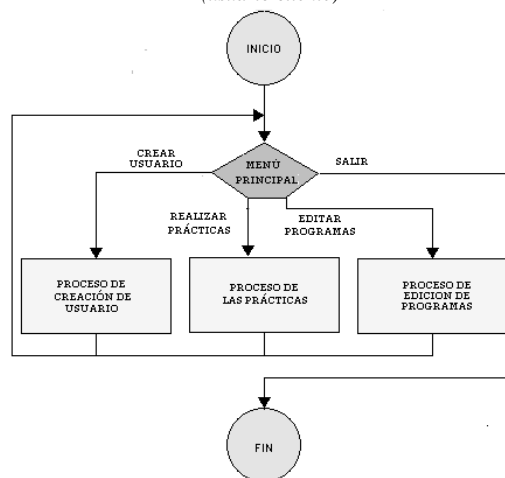
- Crear una cuenta de usuario.
- Entrar a una cuenta de usuario con su clave respectiva.
- Eliminar cuentas de usuario (sólo el usuario administrador tiene esta opción).
- Crear el escenario de simulación según las condiciones establecidas por el usuario.
- Editar los datos de entrada de un escenario de simulación.
- Realizar la simulación bajo las condiciones elegidas por el usuario.
- Graficar las variables de un escenario de simulación.
- Guardar en una carpeta exclusiva de un usuario los datos y gráficas de un escenario simulado por dicho usuario.
- Consultar los datos de entrada que definieron a un escenario que ya ha sido simulado por el mismo usuario que está accediendo al sistema.
- Consultar los datos de salida de un escenario que ya ha sido simulado por el mismo usuario que está accediendo al sistema.
- Consultar los datos de entrada que definieron a un escenario que ya ha sido simulado por otro usuario (esta funcionalidad puede activarse sólo para el administrador o para todos los usuarios según el criterio del usuario administrador).
- Consultar los datos de salida de un escenario que ya ha sido simulado por otro usuario (de igual manera que con la funcionalidad anterior, puede activarse esta opción sólo para el administrador o para todos los usuarios según el criterio del usuario administrador).
- Activar o desactivar opciones de acceso a datos (sólo permitido al usuario administrador).
- Editar los modelos para adaptarlos a los requerimientos de precisión deseados por el usuario.
- Agregar nuevos sistemas que complementen la plataforma de simulación.
- Utilizar modelos creados por el usuario.
- Volver al sistema original producido en el presente proyecto.

Otro aspecto importante a tener en cuenta entre los requisitos de la plataforma de simulación son los actores que intervienen en ella, los cuales son de dos tipos: un usuario administrador y un número indeterminado de usuarios básicos.

3.2 DISEÑO

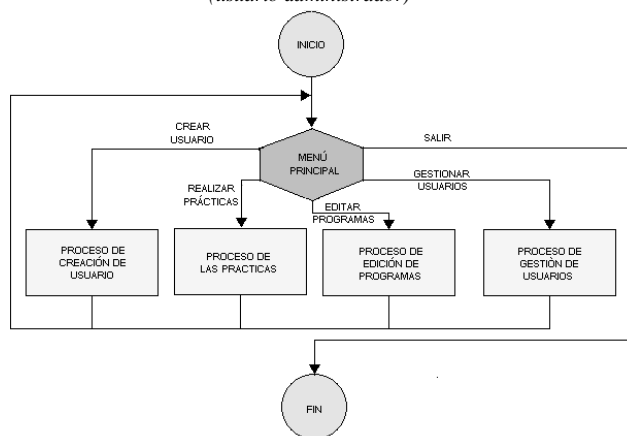
A continuación se muestran los diagramas de flujo generales de la plataforma de simulación (el primero es el correspondiente al usuario cliente y el segundo el correspondiente al usuario administrador):

Figura 6. Diagrama de flujo general de la plataforma de simulación (usuario cliente)



Fuente: El autor

Figura 7. Diagrama de flujo general de la plataforma de simulación (usuario administrador)



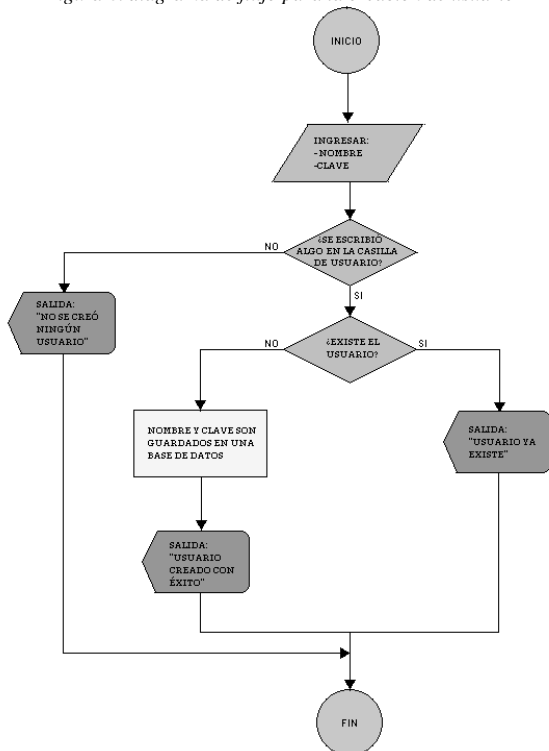
Fuente: El autor

A continuación, se exponen los procesos correspondientes a cada opción del menú principal:

3.2.1 Creación de usuario

El diagrama de flujo para este proceso aparece en la figura 8.

Figura 8. diagrama de flujo para la creación de usuario

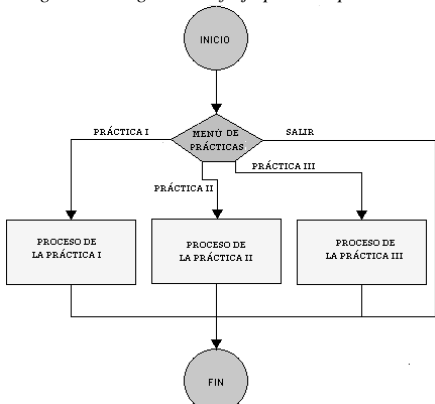


Fuente: el autor

3.2.2 Realizar prácticas.

Este proceso, tiene en realidad tres subprocesos, los cuales son las tres prácticas que se pueden realizar con la plataforma de simulación. En la figura 9 aparece el diagrama de flujo correspondiente a este proceso.

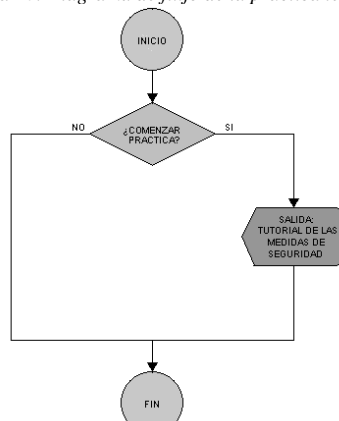
Figura 9. Diagrama de flujo para las prácticas



Fuente: El autor

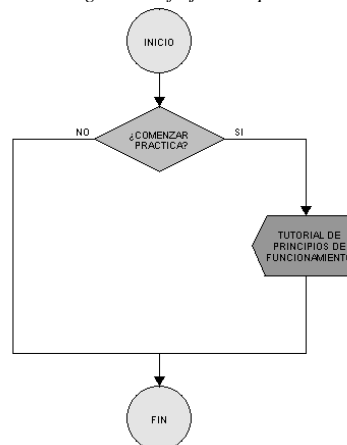
Luego, el proceso correspondiente a cada una de las prácticas aparece en las figuras 10 a 12.

Figura 10. Diagrama de flujo de la práctica teórica 1



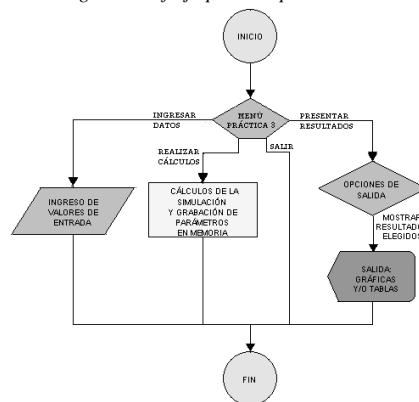
Fuente: El autor

Figura 11. Diagrama de flujo de la práctica teórica 2



Fuente: El autor

Figura 12. Diagrama de flujo para las prácticas de simulación



Fuente: El autor

Ahora, a continuación, se describe el proceso de simulación de la conversión de energía en un grupo electrógeno, tanto con carga eléctrica lineal como con carga eléctrica no lineal.

En primera instancia se toman los datos correspondientes al MCI y se aplican a los modelos térmico de combustión y de pérdidas

térmicas (expuestos en el apartado 2.4.3) y al modelo de pérdidas mecánicas (expuesto en el apartado 2.4.2). Entonces, con los datos de salida de los modelos mencionados, se obtienen las potencias indicada y equivalente.

A continuación, la potencia equivalente y los parámetros del GI se utilizan como entrada del modelo del GI (uno de los expuestos en el apartado 2.3). Luego, la potencia equivalente es aplicada para hallar los parámetros de transformación (tensión y corriente), con los cuales se hallan los valores de la corriente, la tensión y las potencias de la carga (las cuales se hallan mediante uno de los modelos del apartado 2.2). Esto es válido si la carga es lineal.

Luego de esto, se halla la eficiencia del grupo generador comparando el valor del flujo térmico de combustión con la potencia de dimensionamiento de la carga, para conocer el porcentaje de la potencia del combustible que es convertida en potencia eléctrica.

3.2.3 Modelo de simulación

El modelo de simulación se implementará en la plataforma mediante programas independientes (módulos) que interactúan entre sí y con la interfaz con el usuario. Los algoritmos correspondientes a cada módulo aparecen en el Anexo B.

3.2.4 Interfaz con el usuario

La interfaz con el usuario está compuesta de ventanas de menús (menú principal y submenús), ventanas de solicitud de datos, ventanas de entrega de datos (tablas y gráficos), ventanas de ayuda (teóricas y prácticas). A continuación, se muestra la apariencia de estas ventanas, y se explica el uso de cada una de ellas:

3.2.4.1 Ventana del menú principal

Figura 13. Ventana del menú principal



Fuente: El autor

En esta ventana aparecen las opciones principales de la plataforma de simulación, las cuales corresponden a las prácticas que se pueden realizar con la plataforma que se dividen en dos categorías: aquellas que se realizan dentro de la práctica de laboratorio de la asignatura Máquinas Térmicas Alternativas de la Escuela de Ingeniería Mecánica y aquellas que tiene que ver con el análisis de la medición de potencia eléctrica.

En cuanto a las opciones que tienen que ver con el laboratorio de la asignatura mencionada, estas corresponden a las tres prácticas explicadas en la sección 3.2.2 y expuestas en el Anexo C.

En cambio, las opciones que tienen que ver con el análisis de la medición de potencia son explicadas a continuación:

En primer lugar, aparece el análisis de sistemas monofásicos con armónicos. Este análisis parte de datos de la simulación o de

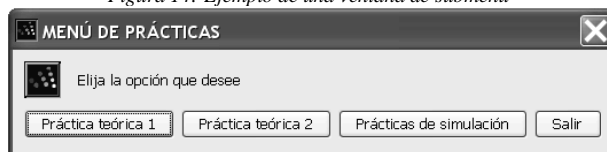
otros datos dados por el usuario para mostrar a este los conceptos que tienen que ver con la medición de potencia de este tipo de sistemas, con el fin de que el usuario pueda apreciar las dificultades que pueden presentarse al medir la potencia eléctrica y por lo tanto al determinar la eficiencia de un sistema tal como un grupo generador.

De la misma manera, la otra opción del análisis de medición de potencia permite al usuario apreciar los conceptos que tienen que ver con la medición de potencia en sistemas trifásicos con armónicos y/o desbalanceados.

3.2.4.2 Ventanas de los submenús

Los submenús de la plataforma de simulación son implementados con el propósito de presentar todas las opciones del sistema en el momento en que estas son requeridas por el usuario, ya sea de forma secuencial (dentro del flujo normal de la ejecución de los programas que componen la plataforma de simulación), o en los casos en que el usuario desee utilizarlos para realizar ajustes al escenario analizado.

Figura 14. Ejemplo de una ventana de submenú



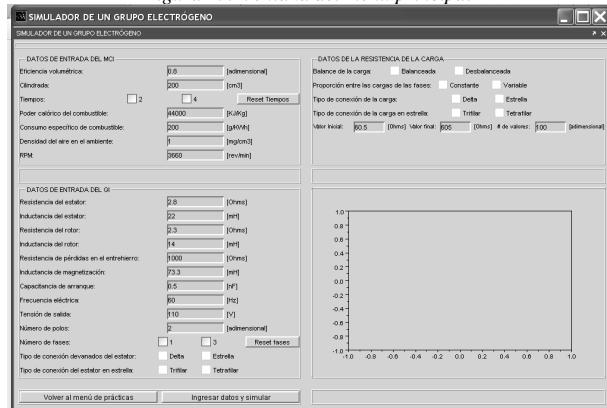
Fuente: El autor

3.2.4.3 Ventana de las prácticas de simulación

Esta ventana pide que el usuario ingrese los datos de entrada del MCI, del GI y de la carga eléctrica del GI, dando una corta explicación de los tipos de datos que se están requiriendo y generando una ventana de error (ver sección 3.2.4.4) luego de oprimir el botón ingresar datos y simular, en caso de que el valor ingresado no sea congruente con el valor esperado (si es el caso) o en caso de que no se haya ingresado algún valor.

También permite ver los resultados de la simulación de un escenario creado por el usuario (ver sección 3.4.1), en forma de tablas o gráficamente.

Figura 15. Ventana del menú principal



Fuente: El autor

3.2.4.4 Ventanas de ayuda

Estas ventanas tienen varios propósitos y por esto pueden ser:

- Ventanas de tutoría, las cuales explican conceptos teóricos fundamentales para entender el funcionamiento de la plataforma de simulación y el de un grupo electrógeno real. Además, también pueden hacer una exposición de los procesos que se van realizando durante el flujo del proceso de simulación.

Figura 16. Ejemplo de una ventana de tutoría



Fuente: El autor

- Ventanas de error, que han sido implementadas con el fin de avisar al usuario cuando ha cometido un error al ingresar un dato.

Figura 17. Ejemplo de una ventana de error



Fuente: El autor

3.3 IMPLEMENTACIÓN

Ahora se exponen algunos casos de uso que, a juicio del autor, ilustran claramente el funcionamiento básico de la plataforma de simulación.

3.3.1 Crear escenario

Crear un escenario consiste en dar valores a cada una de las variables de entrada. Estos valores definen las características del grupo electrógeno que se desea simular, desde el combustible utilizado, pasando por las características de diseño del MCI y del GI, hasta el comportamiento (lineal o no lineal) y el valor de la carga eléctrica. Un escenario puede ser guardado en algún tipo de memoria permanente (disco duro, CD, memoria extraíble, etc) para analizarlo en un uso posterior de la plataforma.

3.3.2 Simular escenario

La simulación es realizada una vez el usuario considera que el escenario ya está listo y escoge la opción del submenú de la práctica de simulación llamada simular. Este caso de uso es realizado automáticamente por la plataforma de simulación. Si falta algún dato fundamental para realizar la simulación, el sistema generará una ventana de error.

3.3.3 Presentar resultados

De la misma manera que el caso de uso anterior, y luego de que se ha realizado este, la presentación de resultados se realiza en el momento en que el usuario escoge la opción del submenú de simulación llamada resultados. En ese momento aparecerá otro menú, el cual presenta todos los resultados disponibles para que el usuario escoja aquellos que desea observar.

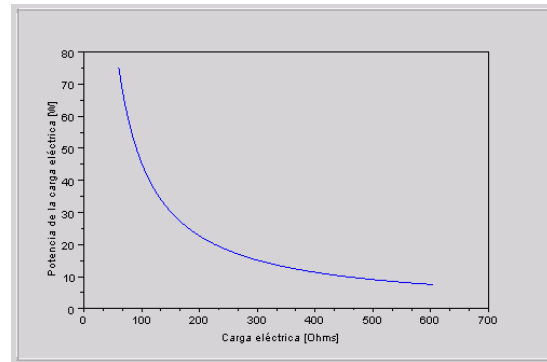
3.3.4 Editar, crear y guardar modelos

Estos casos de uso, permiten que el usuario pueda adaptar la plataforma de simulación para mejorar la precisión o velocidad de proceso de sus modelos ya sea mejorándolos o reemplazándolos por otros.

3.4 PRUEBAS

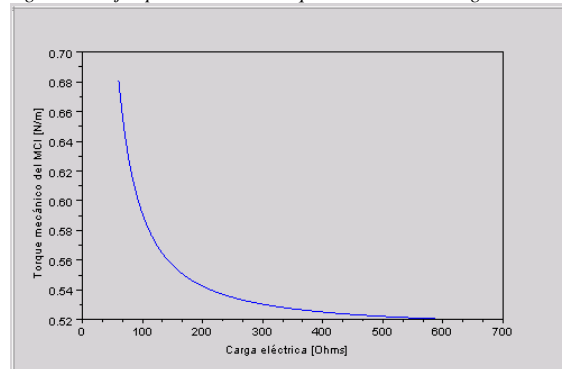
A continuación (en las figuras 18 y 19) se muestran un par de resultados gráficos de la simulación para un MCI de 4 tiempos y un GI monofásico, además se utilizaron los datos por defecto.

Figura 18. Ejemplo de salida: Potencia de la carga eléctrica vs. carga eléctrica



Fuente: El autor

Figura 19. Ejemplo de salida: Torque mecánico vs. carga eléctrica



Fuente: El autor

3.5 PROPUESTA PARA EL MANUAL DE LABORATORIO

A continuación, se exponen los puntos principales de la propuesta para el manual de laboratorio de la plataforma de simulación, para ser utilizado en la asignatura Máquinas Térmicas Alternativas de la Escuela de Ingeniería Mecánica. El manual completo aparece en el anexo C.

3.5.1 Explicación de las medidas de seguridad al manipular un grupo generador.

En esta primera práctica, se presentarán al usuario las precauciones que se deben tener en cuenta al manipular un grupo electrógeno real. Estas precauciones tienen que ver con:

- La manipulación de combustibles.
- El abastecimiento de combustible del MCI.
- El trabajo en un ambiente con aire contaminado.
- El trabajo cerca de sistemas rotativos moviéndose a altas velocidades.
- El trabajo en presencia de campos eléctricos, campos magnéticos y corrientes que pueden poner en riesgo la salud del estudiante.
- Las precauciones que deben tenerse en cuenta al realizar mediciones eléctricas
- Las precauciones que deben tenerse al estar cerca de una máquina eléctrica conectada a la red eléctrica o que está generando electricidad.
- Cuidados que deben realizarse al hacer las pruebas para el cálculo de los parámetros de una máquina de inducción.

3.5.2 Operación del grupo generador

Esta práctica comienza con la identificación de las partes principales de las máquinas que componen el grupo electrógeno: el MCI y el GI; luego, se presentan los principios de operación de estas máquinas; a lo cual sigue la exposición de los procesos de arranque y apagado del grupo generador; para seguidamente explicar los ajustes y mantenimiento necesarios para un buen desempeño del grupo electrógeno.

3.5.3 Cálculo de las salidas del grupo generador

En esta práctica de laboratorio se realiza la simulación del funcionamiento del grupo generador utilizando la plataforma desarrollada en el presente proyecto. De esta manera, se calculan las diferentes salidas del grupo generador, las cuales se pueden discriminar según la máquina de donde proceden de la siguiente manera:

Del MCI: la potencia equivalente, el torque, el rendimiento térmico, el consumo de aire, el consumo de combustible y la relación aire combustible.

Del GI: la potencia consumida por la carga.

Los resultados de esta práctica se mostrarán al usuario en forma de gráficas para cada salida del sistema mencionada en función de la carga eléctrica del grupo electrógeno.

Además, se propone que se implemente en un desarrollo futuro la simulación de las emisiones del MCI, el cual está fuera de los propósitos del presente proyecto.

4. DISEÑO DEL LABORATORIO DE PRUEBAS PARA EL GRUPO ELECTRÓGENO DEL LABORATORIO DE MÁQUINAS TÉRMICAS ALTERNATIVAS DE INGENIERÍA MECÁNICA

En la presente sección se expone la propuesta del autor referente a los instrumentos de medición que pueden ser utilizados para

implementar el banco de pruebas del grupo electrógeno del laboratorio de Máquinas Térmicas Alternativas de la Escuela de Ingeniería Mecánica.

Lo primero que se tuvo en cuenta al escoger los diferentes dispositivos de medición es el hecho de que sus salidas van a ser conectadas a un computador, razón por la cual se optó por escoger sensores y medidores cuya salida sea una señal eléctrica digital.

Ahora bien, las opciones del mercado para los sensores y analizadores de redes, varían en precisión, complejidad y precio, es por esto que el criterio para escoger el más adecuado para el laboratorio en cuestión debe cumplir con un compromiso entre las tres variables mencionadas es decir debe ser el más preciso posible dentro de un rango de precios que se pueda incluir en el presupuesto del laboratorio y con la mínima complejidad en cuanto a conexión y operación.

A continuación se realiza un análisis DOFA del banco de pruebas del grupo electrógeno del Laboratorio de Máquinas Térmicas Alternativas de la Escuela de Ingeniería Mecánica:

Debilidades:

- El grupo generador no está funcionando.
- Los aparatos de medición disponibles son obsoletos.
- No se dispone de un manual de operación y mantenimiento del grupo electrógeno.
- No existe un manual de pruebas de laboratorio

Oportunidades:

- La plataforma de simulación desarrollada en el presente proyecto es un complemento del banco de pruebas del grupo generador.
- La propuesta de adecuación del banco de pruebas que aparece en el presente documento es un punto de partida para llegar a poner en funcionamiento el grupo generador.
- La plataforma de simulación puede dar inicio a una línea de investigación conjunta entre la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones y la Escuela de Ingeniería Mecánica.
- Los estudiantes de las ingenierías Mecánica, Eléctrica y Electrónica; pueden hacer pruebas con la plataforma de simulación.
- A partir de este y otros trabajos dentro de esta línea de investigación, se pueden crear diplomados sobre grupos electrógenos para técnicos y tecnólogos.

Fortalezas:

- Hay un computador disponible para este banco de pruebas.
- El banco de pruebas está destinado para las prácticas de una asignatura específica: Máquinas Térmicas Alternativas.
- Existe una tradición de colaboración para el desarrollo de proyectos entre la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones y la Escuela de Ingeniería Mecánica.
- En la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones existen programas de postgrado a nivel

maestría y doctorado y en la Escuela de Ingeniería Mecánica a su vez, existe un postgrado a nivel maestría, lo cual permite que la rama de investigación del presente proyecto pueda tener un desarrollo elevado en un futuro próximo

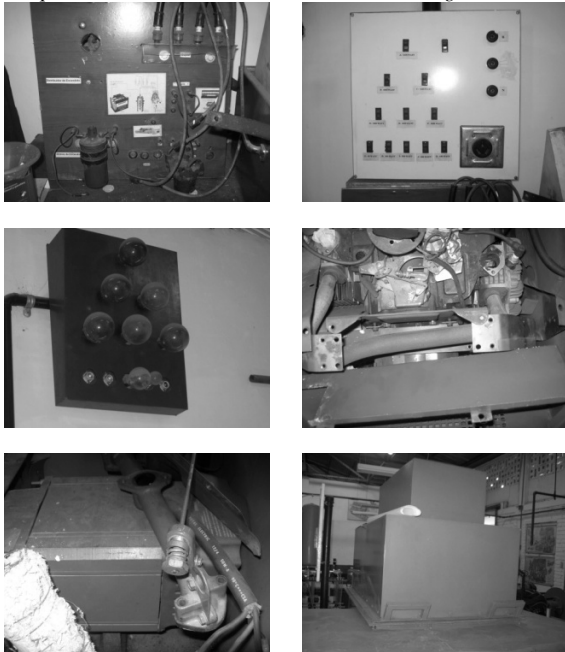
Amenazas:

- Hay pocos estudiantes interesados en aportar de su tiempo para la puesta a punto del banco de pruebas.
- Otras instituciones universitarias de la región al contar con mayores recursos, podrían tomar la delantera por encima de la UIS en esta rama de investigación.
- Existe poca inversión del gobierno destinada a esta rama de investigación.

Se concluye entonces que el presente proyecto es un punto de partida para que el banco de pruebas esté nuevamente en funcionamiento prontamente. Además, lo que se entrega en este documento sirve también como motivador para que muchos estudiantes de Ingeniería Mecánica, Ingeniería Eléctrica y de Ingeniería Electrónica se decidan a entregar de su tiempo y esfuerzo para conseguir un banco de pruebas del grupo generador funcional y óptimo.

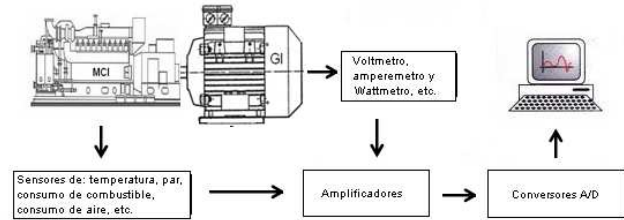
Enseguida, aparecen las propuestas para cada sistema de medición que es necesario adquirir para que el banco de pruebas pueda cumplir con los objetivos de la práctica de laboratorio con un grupo generador dentro del cronograma correspondiente a la asignatura Máquinas Térmicas Alternativas.

Figura 20. Estado actual del Grupo electrógeno del Laboratorio de Máquinas Térmicas Alternativas de la Escuela de Ingeniería Mecánica



Fuente: El autor

Figura 21. Esquema de la propuesta para el laboratorio de pruebas



Fuente: El autor

4.1 MEDIDORES DE FLUJO MÁSICO

4.1.1 Medidor de flujo másico de combustible (gasolina)

Se puede hallar el valor máximo del rango que se necesita para poder medir la magnitud del flujo másico de combustible (gasolina) en el MCI a partir de la siguiente expresión:

$$sfc = \frac{\dot{m}_{f,m\acute{a}x}}{P_{m\acute{a}x}} \quad (84)$$

Para el MCI del grupo electrógeno, el valor del consumo específico de combustible correspondiente es de aproximadamente 200 g / kW h [HEYWOOD, 1988, pag 58] , el valor de la potencia máxima ($P_{m\acute{a}x}$) es 7 kW, por lo tanto el valor máximo del flujo másico de combustible es:

$$\dot{m}_{f,m\acute{a}x} = 2,1 \frac{kg}{h} \quad (85)$$

En la tabla 1 se observan los valores del rango y precisión correspondientes al medidor de flujo másico de combustible:

Tabla 1. Rango y precisión para el medidor de flujo másico de combustible

Rango	Precisión
0 – 2,1 kg/h	+/- 0,02 kg/h

4.1.2 Medidor de flujo volumétrico de combustible (biogás)

El valor máximo del flujo volumétrico de combustible (biogás) se puede hallar

A partir de la ecuación y teniendo en cuenta la relación entre el flujo másico de combustible y el flujo volumétrico de combustible, la cual aparece a continuación:

$$\dot{m}_{f,m\acute{a}x} = \rho_{biogas} \dot{V}_{f,m\acute{a}x} \quad (86)$$

Donde:

densidad del biogás que podemos aproximarla al valor correspondiente al metano en estado gaseoso: 0,717 kg/m³ *.
 flujo volumétrico de combustible (biogás) en m³/s

De esta manera, es posible hallar el flujo volumétrico de combustible máximo reemplazando la ecuación en la ecuación y

* <http://es.wikipedia.org/wiki/Metano>

aplicando los valores correspondientes de las variables. Así el valor máximo del flujo volumétrico de combustible es:

$$\dot{v}_{f,máx} = 2,93 \text{ m}^3/\text{h} \quad (87)$$

En la tabla 2 se observan los valores del rango y precisión correspondientes al medidor de flujo volumétrico de combustible:

Tabla 2. Rango y precisión para el medidor de flujo volumétrico de combustible

Rango	Precisión
0 – 2,93 m ³ /h	+/- 0,03 m ³ /h

4.1.3 Medidor de flujo másico de aire

Para medir el flujo másico de aire, es necesario conocer las características geométricas del MCI, la densidad del aire en el ambiente y la relación de compresión de MCI (o la densidad del aire en el MCI). De esta manera, si se desea medir el flujo másico de aire durante un ciclo del MCI, solo se necesitaría un medidor de densidad de aire, el cual determinaría la precisión de la medición. En cambio, si lo que se desea es conocer el flujo de aire másico instantáneo, se requiere utilizar un medidor de flujo. Para las condiciones y requerimientos del grupo electrógeno del Laboratorio de Máquinas Térmicas Alternativas, la medición de la densidad del aire no es necesaria, pero si es interesante conocer el flujo de aire instantáneo.

La siguiente relación permite hallar el valor máximo del flujo másico de aire a partir del valor máximo del flujo másico de combustible:

$$12 < \dot{m}_a / \dot{m}_f < 18 \quad (88)$$

De modo que, el valor máximo del flujo másico de aire estaría dado por la siguiente expresión:

$$\dot{m}_{a,máx} = 18\dot{m}_{f,máx} = 37,8 \frac{\text{kg}}{\text{h}} \quad (89)$$

Ahora, se recomienda utilizar un anemómetro de hilo caliente para medir el flujo de aire. En la tabla 3 se observan los valores del rango y precisión para este sensor que cumplen con los requerimientos del laboratorio:

Tabla 3. Rango y precisión para el medidor de flujo másico de aire

Rango	Precisión
0 – 37,8 kg/h	+/- 0,4 kg/h

4.2 TERMÓMETROS

4.2.1 Para medir la temperatura ambiental

En este caso se presentan dos alternativas. La primera de ellas consiste en un dispositivo que permita una lectura en el instrumento, o uno que permita una conexión con un computador (se recomienda en este caso utilizar una termocupla). En ambos casos, el rango y la precisión están definidos en la tabla 4.

Tabla 4. Rango y precisión para termómetro que medirá la temperatura ambiental

Rango	Precisión
0-50 °C	+/- 0,1°C

4.2.2 Para medir la temperatura en la admisión

En este caso la recomendación es que se utilice una termocupla con las siguientes características de rango y precisión:

Tabla 5. Rango y precisión para termómetro que medirá la temperatura en la admisión

Rango	Precisión
0-100 °C	+/- 0,1°C

4.2.3 Para medir la temperatura de los gases de escape

En este caso la recomendación es que se utilice una termocupla con las siguientes características de rango y precisión:

Tabla 6. Rango y precisión para termómetro que medirá la temperatura de los gases de escape

Rango	Precisión
0 – 2000 °C	+/- 1°C

4.3 BARÓMETRO

De la misma manera que ocurre con el termómetro, también es posible utilizar dentro del sistema de medición un barómetro que entregue visualmente su medición o bien uno que produzca una señal que pueda ser enviada a un computador, e independientemente del dispositivo escogido, el rango y la precisión recomendadas para un correcto funcionamiento del sistema de medición del grupo electrógeno aparecen en la tabla 7.

Tabla 7. Rango y precisión para barómetro

Rango	Precisión
100 – 200 kPa	+/- 1 kPa

4.4 SISTEMA PARA MEDIR EL TORQUE MECÁNICO Y LA VELOCIDAD ANGULAR

Las opciones analizadas para realizar esta medición son: un aparato industrial de medición de par y velocidad, un sensor de par y velocidad con salida digital y un sensor de par y velocidad con salida analógica.

Debido al costo elevado de un dinamómetro industrial y a la dificultad para adaptar el sensor analógico al sistema de medición, se recomienda entonces utilizar un sensor de par y velocidad de salida digital. El sensor digital más utilizado para medir este parámetro en este tipo de aplicaciones es la celda de carga, por lo tanto se recomienda adquirir una con las siguientes características de rango y precisión (dados una potencia máxima de 7000 W y una velocidad de giro de trabajo de 1800 RPM):

Tabla 8. Rango y precisión para el sistema de medición del torque mecánico

Rango	Precisión
0 – 250 N/m	+/- 0,1 N

4.5. SISTEMA PARA MEDIR LAS EMISIONES DEL MCI

Debido a que el estudio de este sistema de medición está fuera de los alcances del presente proyecto, en este punto solo se realiza la recomendación de realizar una futura investigación para implementar un sistema de medición de emisiones para el grupo generador del Laboratorio de Maquinas Térmicas Alternativas. En este laboratorio se busca medir la emisión de los siguientes gases: CO, O₂, CO₂, HC- y metano.

4.6 ANALIZADOR DE REDES DE BAJA POTENCIA

En este caso, debido a que este tipo de dispositivos son costosos, lo que se recomienda es que este sistema sea diseñado a partir de dispositivos que realicen la medición de la corriente y tensión instantáneas, de este modo la salida de dichos dispositivos se conecta a un computador, en el cual se realizan los análisis para calcular las potencias del sistema.

4.7 TARJETA DE ADQUISICIÓN DE DATOS Y COMPUTADOR

Existen en el mercado una gran variedad de tarjetas de adquisición de datos, si los responsables del laboratorio deciden adquirir una de estas, se recomienda, como primera opción, una producida por National Instruments™, debido a la calidad, altamente reconocida, de los productos de este fabricante.

Debido a los costos de las tarjetas comerciales y a que estas no están hechas para una aplicación específica, se concluyó que la mejor recomendación para este sistema es que se diseñe una tarjeta para su uso exclusivo dentro del sistema de medición del grupo electrógeno del que trata este apartado. De este modo se debe comenzar el desarrollo de la tarjeta luego de decidir que computador se va a disponer para utilizarlo en la medición de las variables del grupo electrógeno, y luego de adquirir todos los sensores y sistemas de medición. Esto se debe a que la tarjeta debe ser diseñada para conectarse a un puerto de expansión específico, por lo que es necesario saber que computador está disponible para esta aplicación; además, las señales de los sensores y sistemas de medición definen el diseño de los circuitos de entrada de la tarjeta o, el diseño de los circuitos externos que deben conectarse a esta para cambiar los niveles de las señales y de esta forma sea posible procesar los datos en el computador. Cabe resaltar también que al decidir diseñar la tarjeta se tiene la ventaja de utilizar un computador que esté disponible inmediatamente para el laboratorio, de modo tal que se reducen los costos del sistema.

CONCLUSIONES

1. El modelo del MCI utilizado en el presente proyecto, es un primer acercamiento al estudio de este tipo de motores para estudiantes de la escuela de Ingeniería Eléctrica, Electrónica y de Telecomunicaciones y de Ingeniería Mecánica, orientado principalmente para comprender los conceptos básicos referentes a estas máquinas.
2. El modelo del GI y de la carga eléctrica es un modelo lineal, debido a que las cargas mas comunes para este tipo de generadores son típicamente resistivas, y además, fue escogido como un primer acercamiento a los conceptos de

este tipo de máquinas y cargas para estudiantes de Ingeniería Mecánica y de Ingeniería Eléctrica y Electrónica.

3. Las salidas del sistema, están orientadas a comprender el proceso de transformación de energía ocurrido dentro del grupo generador.
4. La ineficiencia del sistema es compensada por el bajo precio que tendría el biogás al ser producido con biomasa de desecho, de modo que se justifica el uso de este tipo de grupo generador en zonas remotas.

RECOMENDACIONES

1. Desarrollar un modelo para cargas no lineales.
2. Implementar un modelo para sub-armónicos generados debido al diseño distribuido de las bobinas de un GI.
3. Aplicar un análisis de los transitorios mecánicos.
4. Implementar un modelo del proceso de energizar una máquina de inducción para que pueda funcionar como generador de potencia eléctrica.
5. Implementar un modelo del sistema utilizado para garantizar una tensión constante a la salida del GI cuando éste no se encuentra conectado a una red eléctrica externa.

REFERENCIAS

ALLER, José Manuel. Máquinas Eléctricas Rotativas: Introducción a la Teoría General. Editorial Equinoccio. 2008.

BARBOSA MORA, José Luis; SALCEDO SANDOVAL, Juan Felipe. Herramienta Educativa en la Web Basada en la Simulación de los Procesos Termofluidodinámicos de Motores de Combustión Interna, SIMTA 1.0. Trabajo de grado, Escuela de Ingeniería de Sistemas, UIS. 2007.

BASANTA LOPEZ Samuel. Compensación de Huecos para Aerogeneradores: Estudio y Comparación de Diferentes Alternativas. Proyecto fin de carrera, Ingeniería Industrial, Escuela Técnica Superior de Ingeniería, Universidad Pontificia Comillas, Madrid, España. 2007.

BAUTISTA MORANTES, Adan de Jesús. Impacto de los Sistemas de Medición de Ingeniería Eléctrica en la Valoración de la Potencia No Activa y Su Efecto Sobre la Facturación. Trabajo de grado, Master en Ingeniería Eléctrica, Escuela de Ingenierías Eléctrica Electrónica y Telecomunicaciones, UIS. 2007.

CHAPMAN, Stephen J. Máquinas Eléctricas, 4ª Edición. Mc. Graw Hill. 2005.

FRAILE MORA, Jesús. Máquinas Eléctricas, 5ª Edición. Mc. Graw Hill. 2003

GUERRERO CUESTA, William Edixón. Estudio de la Variación del Voltaje y la Frecuencia de Generación de una Máquina de Inducción Funcionando como Generador. Tesis,

Ingeniería Eléctrica, Escuela Politécnica Nacional. Ecuador. 2006.

HEYWOOD, John B. Internal Combustion Engine Fundamentals. Mc Graw Hill. 1988.

IEEE 1459–2000, IEEE Trial-Use Standard Definitions for the Measurement of Electric Power Quantities Under Sinusoidal, Nonsinusoidal, Balanced, or Unbalanced Conditions. Power System Instrumentation and Measurements Committee of the IEEE Power Engineering Society. IEEE-SA Standards Board. 30 January 2000.

MARTYR, A. J.; PLINT, M. A. Engine Testing. 3rd Edition. Butterworth – Heinemann. 2007

MOSQUERA TOVAR, Fabián Ricardo; DUARTE ROA, William Eduardo. Análisis Comparativo de los Diferentes Sistemas de Medición de Gas Natural. Trabajo de grado. Escuela de Ingeniería Mecánica, UIS. 2002.

MORA, Héctor Manuel. Introducción a SCILAB. Universidad Nacional. Bogotá. 2005.

NIGRO Norberto, STORTI Mario A., AMBROGGI Leonardo. Modelización Numérica de un Motor de Combustión Interna Monocilíndrico Encendido por Chispa. Santa Fe, Argentina. 1999.

RAMAJO, Damián E.; NIGRO, Norberto. Modelización numérica de la Combustión en Motores de Combustión Interna. Centro Internacional de Mecánica Computacional CIMEC, Instituto de Tecnología Química, INTEC, CONICET, Argentina. 2004.

SIMÖES, Godoy. Renewable Energy System. Design and Analysis with Inductor Generators. CRC Press. 2004.

BIOGRAFÍAS



Juan Camilo Pradilla Rodríguez
Estudiante de Ingeniería Electrónica en la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones de la Universidad Industrial de Santander.
Áreas de interés: generación eléctrica, simulación de sistemas físicos, mediciones eléctricas y transformación de energía,

entre otras.



Dr. Jorge Luis Chacón Velasco
Ingeniero Mecánico Especialista En Automóviles y Tractores, IPB (URSS). Doctor en Procesos Termofluidodinámicos aplicados a MCIA, UPV (Valencia, España).

Actualmente Profesor Asociado de la Universidad Industrial de Santander en Bucaramanga Colombia, adscrito a la Escuela de Ingeniería Mecánica.



Dr. Gabriel Ordóñez Plata
Ingeniero Electricista, Universidad Industrial de Santander. Especialista Universitario en Técnicas de Investigación, UPCMadríd.

Doctor Ingeniero Industrial, Universidad Pontificia Comillas, Madrid.
Actualmente Profesor Asociado de la Universidad Industrial de Santander en Bucaramanga Colombia, adscrito a la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

ANEXOS

ANEXO A. SOLUCIÓN AL SISTEMA DE ECUACIONES DEL ANÁLISIS TRANSITORIO DEL GI TRIFÁSICO

En el presente anexo, se presentan las ecuaciones que permiten dar la solución al sistema de ecuaciones formado por las fórmulas 69 y de la 72 a la 74 del apartado 2.3.2 (Modelo del GI trifásico en régimen transitorio).

Entonces, aplicando el método de Gauss-Jordan al sistema de ecuaciones formado por las fórmulas 69 y de la 72 a la 74 se obtienen las expresiones para I_L , para I_c , para I_m y para I_h :

$$I_{Ltran} = \frac{sL_m}{[(R_1 + sL_1)(1 + sCR_L) + R_L] \left[(R_2 + sL_2) \left(1 + \frac{sL_m}{R_h} \right) + sL_m \right] + sL_m(R_2 + sL_2)(1 + sCR_L)} V_T \quad (A1)$$

$$I_{Ctran} = \frac{s^2 L_m C R_L}{[(R_1 + sL_1)(1 + sCR_L) + R_L] \left[(R_2 + sL_2) \left(1 + \frac{sL_m}{R_h} \right) + sL_m \right] + sL_m(R_2 + sL_2)(1 + sCR_L)} V_T \quad (A2)$$

$$I_{Mtran} = \frac{(R_1 + sL_1)(1 + sCR_L) + R_L}{[(R_1 + sL_1)(1 + sCR_L) + R_L] \left[(R_2 + sL_2) \left(1 + \frac{sL_m}{R_h} \right) + sL_m \right] + sL_m(R_2 + sL_2)(1 + sCR_L)} V_T \quad (A3)$$

$$I_{Htran} = \frac{\frac{sL_m}{R_h} [(R_1 + sL_1)(1 + sCR_L) + R_L]}{[(R_1 + sL_1)(1 + sCR_L) + R_L] \left[(R_2 + sL_2) \left(1 + \frac{sL_m}{R_h} \right) + sL_m \right] + sL_m(R_2 + sL_2)(1 + sCR_L)} V_T \quad (A4)$$

Donde:

$$V_T = \frac{s}{s^2 + \omega^2} V_{T\text{máx}} \quad (A5)$$

A.1 Cálculo de las Variables Transitorias de la Carga

Como los polos pueden ser complejos, se parte de la expresión de la corriente transitoria de carga correspondiente a dos polos conjugados complejos:

$$I_{Ltran_{1,2}} = \frac{Re(A_1) + jIm(A_1)}{s - Re(s_1) + jIm(s_1)} + \frac{Re(A_1) - jIm(A_1)}{s - Re(s_1) - jIm(s_1)} \quad (A6)$$

Ahora, utilizando un común múltiplo de los denominadores de la ecuación A6 (multiplicándolos) se halla una expresión equivalente agrupando los términos reales en el numerador de la izquierda y los imaginarios en el numerador de la derecha; se obtiene la siguiente expresión de $I_{Ltran_{1,2}}$:

$$I_{Ltran_{1,2}} = \frac{2Re(A_1)(s - Re(s_1))}{s^2 - 2sRe(s_1) + Re(s_1)^2 + Im(s_1)^2} + \frac{2Im(A_1)Im(s_1)}{s^2 - 2sRe(s_1) + Re(s_1)^2 + Im(s_1)^2} \quad (A7)$$

Ahora, a partir de las raíces del denominador de cada sumando del lado derecho de la ecuación A7, se obtienen los exponentes de los términos exponenciales y las expresiones de los términos sinusoidales de la expresión de $i_{Ltran_{1,2}}$ (en función del tiempo), la cual aparece a continuación:

$$i_{Ltran_{1,2}} = 2Re(A_1)e^{Re(s_1)t} \cos(Im(s_1)t) + 2Im(A_1)e^{Re(s_1)t} \sen(Im(s_1)t) \quad (A8)$$

Puede notarse entonces que la expresión del término n-ésimo de la corriente transitoria de carga es simplemente la mitad de la expresión anterior, es decir:

$$i_{Ltran_n} = Re(A_n)e^{Re(s_n)t} \cos(Im(s_n)t) + Im(A_n)e^{Re(s_n)t} \sen(Im(s_n)t) \quad (A9)$$

Por lo tanto, la expresión de la corriente de carga transitoria es como sigue:

$$i_{Ltran} = \sum_{n=1}^4 [Re(A_n)e^{Re(s_n)t} \cos(Im(s_n)t) + Im(A_n)e^{Re(s_n)t} \sen(Im(s_n)t)] \quad (A10)$$

Los valores A_n corresponden a los numeradores de la descomposición en fracciones parciales de la ecuación A1.

Entonces, teniendo en cuenta que después de un tiempo igual a 5τ (donde τ corresponde al valor máximo de los $\tau_i=1/s_i$) los valores de la corriente y la tensión son aproximadamente iguales a sus valores en régimen permanente, la expresión para la potencia del transitorio de la carga se obtiene de la siguiente manera:

$$P_{Ltran} = \frac{1}{5\tau} \int_0^{\infty} v_{Ltran} i_{Ltran} dt \quad (A11)$$

Ahora, la expresión de v_{Ltran} depende de la naturaleza de la carga de modo tal que, como en esta aplicación la carga es resistiva, la expresión para v_{Ltran} es como sigue:

$$v_{Ltran} = R_L \sum_{m=1}^4 [Re(A_m)e^{Re(s_m)t} \cos(Im(s_m)t) + Im(A_m)e^{Re(s_m)t} \sen(Im(s_m)t)] \quad (A12)$$

Por tanto, reemplazando en la ecuación A11 las expresiones de v_{Ltran} de la ecuación 12 y de i_{Ltran} de la ecuación A10, la expresión para la potencia del transitorio en la carga, en este caso, es como sigue:

$$P_{Ltran} = \frac{R_L}{5\tau} \int_0^{\infty} \left[\sum_{n=1}^4 Re(A_n) e^{Re(s_n)t} \cos(Im(s_n)t) + Im(A_n) e^{Re(s_n)t} \sen(Im(s_n)t) \right] * \left[\sum_{m=1}^4 Re(A_m) e^{Re(s_m)t} \cos(Im(s_m)t) + Im(A_m) e^{Re(s_m)t} \sen(Im(s_m)t) \right] dt \quad (A13)$$

Ahora, al multiplicar las expresiones que aparecen entre corchetes en la ecuación A13 y agrupar los términos semejantes resulta:

$$P_{Ltran} = \frac{R_L}{5\tau} \int_0^{\infty} \sum_{m=1}^4 \sum_{n=1}^4 [Re(A_m)Re(A_n) e^{[Re(s_m)+Re(s_n)]t} \cos(Im(s_m)t) \cos(Im(s_n)t) + Im(A_m)Im(A_n) e^{[Re(s_m)+Re(s_n)]t} \sen(Im(s_m)t) \sen(Im(s_n)t) + 2Re(A_m)Im(A_n) e^{[Re(s_m)+Re(s_n)]t} \sen(Im(s_m)t) \cos(Im(s_n)t)] dt \quad (A14)$$

Aplicando la propiedad de las integrales que dice que la integral de una suma es igual a la suma de las integrales de los sumandos, tenemos:

$$P_{Ltran} = \frac{R_L}{5\tau} \sum_{m=1}^4 \sum_{n=1}^4 \left[Re(A_m)Re(A_n) \int_0^{\infty} e^{[Re(s_m)+Re(s_n)]t} \cos(Im(s_m)t) \cos(Im(s_n)t) dt \right] + \left[Im(A_m)Im(A_n) \int_0^{\infty} e^{[Re(s_m)+Re(s_n)]t} \sen(Im(s_m)t) \sen(Im(s_n)t) dt \right] +$$

$$\left[2Re(A_m)Im(A_n) \int_0^{\infty} e^{[Re(s_m)+Re(s_n)]t} \text{sen}(Im(s_m)t) \cos(Im(s_n)t) dt \right] \quad (A15)$$

Aplicando las identidades trigonométricas producto de cosenos, producto de senos y producto de coseno por seno, se obtiene:

$$\begin{aligned} P_{Ltran} = \frac{R_L}{5\tau} \sum_{m=1}^4 \sum_{n=1}^4 & \left[\frac{Re(A_m)Re(A_n)}{2} \int_0^{\infty} [e^{[Re(s_m)+Re(s_n)]t} \cos(Im(s_m)t + Im(s_n)t) + e^{[Re(s_m)+Re(s_n)]t} \cos(Im(s_m)t - Im(s_n)t)] dt + \right. \\ & \frac{Im(A_m)Im(A_n)}{2} \int_0^{\infty} [e^{[Re(s_m)+Re(s_n)]t} \cos(Im(s_m)t - Im(s_n)t) - e^{[Re(s_m)+Re(s_n)]t} \cos(Im(s_m)t + Im(s_n)t)] dt + \\ & \left. Re(A_m)Im(A_n) \int_0^{\infty} [e^{[Re(s_m)+Re(s_n)]t} \text{sen}(Im(s_m)t + Im(s_n)t) + e^{[Re(s_m)+Re(s_n)]t} \text{sen}(Im(s_m)t - Im(s_n)t)] dt \right] \quad (A16) \end{aligned}$$

Aplicando las propiedades de la transformada de Laplace a la ecuación anterior, se obtiene:

$$P_{Ltran} = \frac{R_L}{5\tau} \sum_{m=1}^4 \sum_{n=1}^4 \left[\frac{Re(A_m)Re(A_n)}{2} \lim_{s \rightarrow 0} \left(\frac{s - Re(s_m) - Re(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} + \frac{s - Re(s_m) - Re(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} \right) + \right.$$

$$\begin{aligned}
& \frac{Im(A_m)Im(A_n)}{2} \lim_{s \rightarrow 0} \left(\frac{s - Re(s_m) - Re(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} - \right. \\
& \quad \left. \frac{s - Re(s_m) - Re(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} \right) + \\
& Re(A_m)Im(A_n) \lim_{s \rightarrow 0} \left(\frac{Im(s_m) + Im(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} + \right. \\
& \quad \left. \frac{Im(s_m)t - Im(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} \right) \tag{A17}
\end{aligned}$$

De este modo, luego de algunas simplificaciones, la expresión para la potencia activa de la carga cuando esta es resistiva aparece a continuación:

$$\begin{aligned}
P_{Ltran} = \frac{R_L}{10\tau} \sum_{m=1}^4 \sum_{n=1}^4 & \left[\frac{(Im(A_m)Im(A_n) - Re(A_m)Re(A_n))(Re(s_m) + Re(s_n)) + 2Re(A_m)Im(A_n)(Im(s_m) + Im(s_n))}{(Re(s_m) + Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} - \right. \\
& \left. \frac{(Re(A_m)Re(A_n) + Im(A_m)Im(A_n))(Re(s_m) + Re(s_n)) + 2Re(A_m)Im(A_n)(Im(s_m) - Im(s_n))}{(Re(s_m) + Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} \right] \tag{A18}
\end{aligned}$$

A.2 Cálculo de las Variables Transitorias de la Resistencia de Pérdidas en el Entrehierro

Del mismo modo que ocurre con la carga resistiva, la expresión para la tensión en la resistencia de pérdidas en el entrehierro es la siguiente:

$$v_{Htran} = R_L \sum_{m=1}^4 [Re(A_m)e^{Re(s_m)t} \cos(Im(s_m)t) + Im(A_m)e^{Re(s_m)t} \sen(Im(s_m)t)] \quad (A19)$$

Por lo anterior, la expresión para la potencia de pérdidas en el entrehierro es la siguiente:

$$P_{Htran} = \frac{R_H}{10\tau} \sum_{m=1}^4 \sum_{n=1}^4 \left[\frac{(Im(A_m)Im(A_n) - Re(A_m)Re(A_n))(Re(s_m) + Re(s_n)) + 2Re(A_m)Im(A_n)(Im(s_m) + Im(s_n))}{(Re(s_m) + Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} - \frac{(Re(A_m)Re(A_n) + Im(A_m)Im(A_n))(Re(s_m) + Re(s_n)) + 2Re(A_m)Im(A_n)(Im(s_m) - Im(s_n))}{(Re(s_m) + Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} \right] \quad (A20)$$

Cabe aclarar, que los valores de A_m y de A_n de las ecuaciones A19 y A20 son diferentes a los correspondientes valores de las ecuaciones A12 y A18. Al final del presente anexo se expone el método utilizado para hallar estos valores.

En cambio, las expresiones para Q_C y Q_M se calculan a partir de una definición de la tensión transitoria correspondiente diferente a la que aparece en la ecuación A12.

A.3 Cálculo de las Variables Transitorias de la Capacitancia de Arranque

La tensión transitoria del capacitor de arranque del GI se expresa así:

$$v_{Ctran} = \frac{1}{C} \int_0^t \left[\sum_{n=1}^4 [Re(A_n)e^{Re(s_n)t} \cos(Im(s_n)t) + Im(A_n)e^{Re(s_n)t} \sen(Im(s_n)t)] \right] dt \quad (A21)$$

Resolviendo la integral se tiene:

$$v_{Ctran} = \frac{1}{C(Re(A_n)^2 + Im(s_n)^2)} \sum_{m=1}^4 [(Re(A_m)Re(s_m) - Im(A_m)Im(s_m))e^{Re(s_m)t} \cos(Im(s_m)t) + (Re(A_m)Im(s_m) + Im(A_m)Re(s_m))e^{Re(s_m)t} \sen(Im(s_m)t)] \quad (A22)$$

Entonces, la expresión de la potencia del transitorio en la capacitancia, es:

$$Q_{Ctran} = \frac{1}{C(Re(A_n)^2 + Im(s_n)^2)5\tau} \int_0^{\infty} \sum_{n=1}^4 [Re(A_n)e^{Re(s_n)t} \cos(Im(s_n)t) + Im(A_n)e^{Re(s_n)t} \sen(Im(s_n)t)] * \sum_{m=1}^4 [Re(A_m)Re(s_m) - Im(A_m)Im(s_m)]e^{Re(s_m)t} \cos(Im(s_m)t) + [Re(A_m)Im(s_m) + Im(A_m)Re(s_m)]e^{Re(s_m)t} \sen(Im(s_m)t) dt \quad (A23)$$

Ahora, al multiplicar las sumatorias de la ecuación A23, resulta:

$$\begin{aligned}
Q_{ctran} = & \frac{1}{C(Re(A_n)^2 + Im(s_n)^2)5\tau} \int_0^\infty \sum_{m=1}^4 \sum_{n=1}^4 (Re(A_m)Re(s_m) - Im(A_m)Im(s_m))Re(A_n)e^{[Re(s_m)+Re(s_n)]t} \cos(Im(s_m)t) \cos(Im(s_n)t) + \\
& (Re(A_m)Im(s_m) + Im(A_m)Re(s_m))Im(A_n)e^{[Re(s_m)+Re(s_n)]t} \sen(Im(s_m)t) \sen(Im(s_n)t) + \\
& (Re(A_m)Re(s_m) - Im(A_m)Im(s_m))Im(A_n)e^{[Re(s_m)+Re(s_n)]t} \cos(Im(s_m)t) \sen(Im(s_n)t) + \\
& (Re(A_m)Re(s_m) + Im(A_m)Im(s_m))Re(A_n)e^{[Re(s_m)+Re(s_n)]t} \sen(Im(s_m)t) \cos(Im(s_n)t) dt \quad (A24)
\end{aligned}$$

Aplicando las identidades trigonométricas producto de cosenos, producto de senos y producto de coseno por seno y las propiedades de la transformada de Laplace a la ecuación anterior, se obtiene:

$$\begin{aligned}
Q_{ctran} = & \frac{1}{C(Re(A_n)^2 + Im(s_n)^2)5\tau} * \\
& \sum_{m=1}^4 \sum_{n=1}^4 \left[\frac{(Re(A_m)Re(s_m) - Im(A_m)Im(s_m))Re(A_n)}{2} * \right. \\
& \lim_{s \rightarrow 0} \left(\frac{s - Re(s_m) - Re(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} + \frac{s - Re(s_m) - Re(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} \right) + \\
& \frac{(Re(A_m)Im(s_m) + Im(A_m)Re(s_m))Im(A_n)}{2} * \\
& \lim_{s \rightarrow 0} \left(\frac{s - Re(s_m) - Re(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} - \frac{s - Re(s_m) - Re(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} \right) + \\
& \left. \frac{(Re(A_m)Re(s_m) - Im(A_m)Im(s_m))Im(A_n)}{2} * \right.
\end{aligned}$$

$$\begin{aligned}
& \lim_{s \rightarrow 0} \left(\frac{Im(s_m) + Im(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} - \frac{Im(s_m)t - Im(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} \right) + \\
& \frac{(Re(A_m)Re(s_m) + Im(A_m)Im(s_m))Re(A_n)}{2} * \\
& \lim_{s \rightarrow 0} \left(\frac{Im(s_m) + Im(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} + \frac{Im(s_m)t - Im(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} \right) \quad (A25)
\end{aligned}$$

De este modo, luego de algunas simplificaciones, la expresión para la potencia transitoria del capacitor de arranque del GI aparece a continuación:

$$\begin{aligned}
Q_{ctran} &= \frac{1}{C(Re(A_n)^2 + Im(s_n)^2)10\tau} * \\
& \sum_{m=1}^4 \sum_{n=1}^4 \left[\frac{\left((Re(A_m)Im(s_m) + Im(A_m)Re(s_m))Im(A_n) - (Re(A_m)Re(s_m) - Im(A_m)Im(s_m))Re(A_n)) \right) (Re(s_m) + Re(s_n))}{(Re(s_m) + Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} + \right. \\
& \left. \frac{\left((Re(A_m)Re(s_m) - Im(A_m)Im(s_m))Im(A_n) + (Re(A_m)Re(s_m) + Im(A_m)Im(s_m))Re(A_n)) \right) (Im(s_m) + Im(s_n))}{(Re(s_m) + Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} - \right. \\
& \left. \frac{\left((Re(A_m)Re(s_m) - Im(A_m)Im(s_m))Re(A_n) + (Re(A_m)Im(s_m) + Im(A_m)Re(s_m))Im(A_n)) \right) (Re(s_m) + Re(s_n)) + \right. \\
& \left. \frac{\left((Re(A_m)Re(s_m) + Im(A_m)Im(s_m))Re(A_n) - (Re(A_m)Re(s_m) - Im(A_m)Im(s_m))Im(A_n)) \right) (Im(s_m) - Im(s_n))}{(Re(s_m) + Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} \right) \quad (A26)
\end{aligned}$$

A.4 Cálculo de las Variables Transitorias de la Inductancia de Magnetización

Ahora, la tensión transitoria de la inductancia de magnetización del GI se expresa así:

$$v_{Mtran} = L_M \frac{d}{dt} \left[\sum_{m=1}^4 [Re(A_m)e^{Re(s_m)t} \cos(Im(s_m)t) + Im(A_m)e^{Re(s_m)t} \sen(Im(s_m)t)] \right] \quad (A27)$$

Derivando la ecuación A26 se obtiene:

$$v_{Mtran} = L_M \sum_{n=1}^4 [Re(A_m)(e^{Re(s_m)t} \cos(Im(s_m)t) - e^{Re(s_m)t} \sen(Im(s_m)t)) + Im(A_m)(e^{Re(s_m)t} \sen(Im(s_m)t) + e^{Re(s_m)t} \cos(Im(s_m)t))] \quad (A28)$$

Entonces, la potencia del transitorio de magnetización del GI se expresa de la siguiente manera:

$$Q_{Mtran} = \frac{L_M}{5\tau} \int_0^{\infty} \left[\sum_{n=1}^4 [Re(A_n)e^{Re(s_n)t} \cos(Im(s_n)t) + Im(A_n)e^{Re(s_n)t} \sen(Im(s_n)t)] \right] * \left[\sum_{m=1}^4 [(Re(A_m) + Im(A_m))e^{Re(s_m)t} \cos(Im(s_m)t) + (Im(A_m) - Re(A_m))e^{Re(s_m)t} \sen(Im(s_m)t)] \right] dt \quad (A29)$$

Ahora, al multiplicar las sumatorias de la ecuación A29, resulta:

$$\begin{aligned}
Q_{Mtran} = & \frac{L_M}{5\tau} \int_0^{\infty} \sum_{m=1}^4 \sum_{n=1}^4 (Re(A_m) + Im(A_m)) Re(A_n) e^{[Re(s_m)+Re(s_n)]t} \cos(Im(s_m)t) \cos(Im(s_n)t) + \\
& (Im(A_m) - Re(A_m)) Im(A_n) e^{[Re(s_m)+Re(s_n)]t} \sen(Im(s_m)t) \sen(Im(s_n)t) + \\
& (Re(A_m) + Im(A_m)) Im(A_n) e^{[Re(s_m)+Re(s_n)]t} \cos(Im(s_m)t) \sen(Im(s_n)t) + \\
& (Im(A_m) - Re(A_m)) Re(A_n) e^{[Re(s_m)+Re(s_n)]t} \sen(Im(s_m)t) \cos(Im(s_n)t) dt \tag{A30}
\end{aligned}$$

Aplicando las identidades trigonométricas producto de cosenos, producto de senos y producto de coseno por seno y las propiedades de la transformada de Laplace a la ecuación anterior, se obtiene:

$$\begin{aligned}
Q_{Mtran} = & \frac{L_M}{5\tau} \sum_{m=1}^4 \sum_{n=1}^4 \left[\frac{(Re(A_m) + Im(A_m)) Re(A_n)}{2} * \right. \\
& \lim_{s \rightarrow 0} \left(\frac{s - Re(s_m) - Re(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) + Im(s_n))^2} + \frac{s - Re(s_m) - Re(s_n)}{(s - Re(s_m) - Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} \right) + \\
& \left. \frac{(Im(A_m) - Re(A_m)) Im(A_n)}{2} * \right]
\end{aligned}$$

$$\begin{aligned}
& \lim_{s \rightarrow 0} \left(\frac{s - \operatorname{Re}(s_m) - \operatorname{Re}(s_n)}{(s - \operatorname{Re}(s_m) - \operatorname{Re}(s_n))^2 + (\operatorname{Im}(s_m) - \operatorname{Im}(s_n))^2} - \frac{s - \operatorname{Re}(s_m) - \operatorname{Re}(s_n)}{(s - \operatorname{Re}(s_m) - \operatorname{Re}(s_n))^2 + (\operatorname{Im}(s_m) + \operatorname{Im}(s_n))^2} \right) + \\
& \frac{(\operatorname{Re}(A_m) + \operatorname{Im}(A_m))\operatorname{Im}(A_n)}{2} * \\
& \lim_{s \rightarrow 0} \left(\frac{\operatorname{Im}(s_m) + \operatorname{Im}(s_n)}{(s - \operatorname{Re}(s_m) - \operatorname{Re}(s_n))^2 + (\operatorname{Im}(s_m) + \operatorname{Im}(s_n))^2} - \frac{\operatorname{Im}(s_m)t - \operatorname{Im}(s_n)}{(s - \operatorname{Re}(s_m) - \operatorname{Re}(s_n))^2 + (\operatorname{Im}(s_m) - \operatorname{Im}(s_n))^2} \right) + \\
& \frac{(\operatorname{Im}(A_m) - \operatorname{Re}(A_m))\operatorname{Re}(A_n)}{2} * \\
& \lim_{s \rightarrow 0} \left(\frac{\operatorname{Im}(s_m) + \operatorname{Im}(s_n)}{(s - \operatorname{Re}(s_m) - \operatorname{Re}(s_n))^2 + (\operatorname{Im}(s_m) + \operatorname{Im}(s_n))^2} + \frac{\operatorname{Im}(s_m)t - \operatorname{Im}(s_n)}{(s - \operatorname{Re}(s_m) - \operatorname{Re}(s_n))^2 + (\operatorname{Im}(s_m) - \operatorname{Im}(s_n))^2} \right) \quad (A31)
\end{aligned}$$

De este modo, luego de algunas simplificaciones, la expresión para la potencia transitoria del capacitor de arranque del GI aparece a continuación:

$$Q_{ctran} = \frac{1}{L_M 10\tau} *$$

$$\sum_{m=1}^4 \sum_{n=1}^4 \left[\frac{\left((\operatorname{Re}(A_m)\operatorname{Im}(s_m) + \operatorname{Im}(A_m)\operatorname{Re}(s_m))\operatorname{Im}(A_n) - (\operatorname{Re}(A_m)\operatorname{Re}(s_m) - \operatorname{Im}(A_m)\operatorname{Im}(s_m))\operatorname{Re}(A_n) \right) (\operatorname{Re}(s_m) + \operatorname{Re}(s_n)) + \left((\operatorname{Re}(A_m)\operatorname{Re}(s_m) - \operatorname{Im}(A_m)\operatorname{Im}(s_m))\operatorname{Im}(A_n) + (\operatorname{Re}(A_m)\operatorname{Re}(s_m) + \operatorname{Im}(A_m)\operatorname{Im}(s_m))\operatorname{Re}(A_n) \right) (\operatorname{Im}(s_m) + \operatorname{Im}(s_n))}{(\operatorname{Re}(s_m) + \operatorname{Re}(s_n))^2 + (\operatorname{Im}(s_m) + \operatorname{Im}(s_n))^2} - \right.$$

$$\frac{\left((Re(A_m)Re(s_m) - Im(A_m)Im(s_m))Re(A_n) + (Re(A_m)Im(s_m) + Im(A_m)Re(s_m))Im(A_n) \right) (Re(s_m) + Re(s_n)) + \left((Re(A_m)Re(s_m) + Im(A_m)Im(s_m))Re(A_n) - (Re(A_m)Re(s_m) - Im(A_m)Im(s_m))Im(A_n) \right) (Im(s_m) - Im(s_n))}{(Re(s_m) + Re(s_n))^2 + (Im(s_m) - Im(s_n))^2} \quad (A32)$$

A.5 Cálculo de las Variables Transitorias para las resistencias e Inductancias del Estator y del Rotor

Las expresiones para la potencia en la resistencia y en la inductancia del estator del GI se calcularon teniendo en cuenta que la corriente en el estator I_1 se define como la suma de la corriente en la capacitancia de arranque del GI más la corriente de carga. De la misma manera, las expresiones para la potencia en la resistencia y en la inductancia del rotor del GI se calcularon teniendo en cuenta que la corriente en el rotor I_2 se define como la suma de la corriente en el estator (I_1) más la corriente de magnetización del GI más la corriente de pérdidas en el entrehierro. Las expresiones así halladas se muestran en el Anexo B como parte del algoritmo utilizado para las variables transitorias del GI.

A.6 Método Utilizado para Calcular los Numeradores de las Fracciones Parciales de las Funciones de Transferencia

El método utilizado para calcular los numeradores de las fracciones parciales de las ecuaciones de la A1 a la A4 es el siguiente:

1. Se calculan las raíces del denominador (que es el mismo para las 4 funciones de transferencia halladas) y se agregan las raíces de la respuesta forzada.
2. Se definen los vectores ("h1", "h2", "h3", "h4", "h5" y "h6") de coeficientes de los polinomios, ordenados de menor a mayor potencia, obtenidos al quitar una raíz (diferente cada vez) al denominador factorizado (aumentado con la respuesta forzada).
3. Con los vectores anteriores, utilizándolos como columnas, se construye una matriz cuadrada "A".

4. La matriz "A" se descompone en sus valores y vectores singulares, obteniendo las matrices "U", "S" y "V"..
5. Se define la matriz "U1" como la traspuesta de la matriz "U".
6. Se define la matriz "S1" cuyos componentes son los recíprocos de los componentes de la matriz "S".
7. Se halla la matriz inversa de "A" mediante la siguiente ecuación: $A_{inv} = V * S1 * U1$ (A33)
8. Se definen los vectores fila ("BL", "BC", "BM" y "BH") de los coeficientes, ordenados de menor a mayor potencia, de cada numerador correspondiente a cada función de transferencia expuestas en las ecuaciones de la A1 a la A4 (aumentada con la respuesta forzada).
9. Finalmente, se calculan los numeradores de las fracciones parciales y se guardan estos valores en los vectores "XL", "XC", "XM" y "XH" mediante las siguientes ecuaciones:

$$XL = V_{Tmáx} A_{inv} * BL \quad (A34)$$

$$XC = V_{Tmáx} A_{inv} * BC \quad (A35)$$

$$XM = V_{Tmáx} A_{inv} * BM \quad (A36)$$

$$XH = V_{Tmáx} A_{inv} * BH \quad (A37)$$

ANEXO B. ALGORITMOS DE LOS MODELOS IMPLEMENTADOS

En el presente anexo se presentan los diferentes algoritmos de los modelos implementados para hallar las variables relacionadas con el proceso de transformación de energía en un grupo generador y las respectivas pérdidas de potencia ocurridas durante dicho proceso.

Figura B.1 Algoritmo para el cálculo de la relación aire – combustible del MCI

```
1 function AvsF // función para realizar el cálculo de la relación aire/combustible en un MCI.
2   execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
   '_' + fechayhora(5) + '_' + fechayhora(6) + '\Flujo_Combustible.bin', 'Mf'); // Desde el directorio creado
   para la presente simulación, se cargan los valores de las salidas totales del GI, desde el vector "Mf" guardado en el
   archivo "Flujo_Combustible.bin".
3   execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
   '_' + fechayhora(5) + '_' + fechayhora(6) + '\Flujodeaire.bin', 'Ma'); // Desde el directorio creado para la
   presente simulación, se cargan los valores de las salidas totales del GI, desde el vector "Ma" guardado en el archivo
   "Flujodeaire.bin".
4   for varAvsF = 1:nRL, // Rutina de iteración para calcular cada uno de la eficiencia del grupo generador
   correspondientes a cada valor de la carga eléctrica del GI.
5     execstr('AvsF(varAvsF) = Ma/Mf(varAvsF)'); // A partir de cada par de valores del flujo de aire y del flujo de
   combustible del MCI se calcula cada valor de la relación aire combustible, y se guardan estos valores en el vector
   "AvsF".
6   end; // Fin de la rutina de iteración que comienza en la línea 3.
7   execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
   '_' + fechayhora(5) + '_' + fechayhora(6) + '\AvsF.bin', AvsF); // Se guarda el vector "AvsF" en el archivo
   "AvsF.bin" y en el directorio creado para la presente simulación.
8   return; // Se termina la ejecución de esta función.
9 endfunction
```

Fuente: El autor

Figura B.2 Algoritmo para el cálculo de la eficiencia del GI

```

1 function Eficiencia_GI // Función para calcular la eficiencia del GI
    execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
    '_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGITotales.bin', 'salidasGITotales'); // Se carga el
matriz "salidasGITotales" desde el archivo "salidasGITotales.bin" ubicado en la carpeta de la fecha y la hora de la
simulación.
for varEfGI = 1:nRL, // Lazo de iteración para calcular cada uno de los valores de la eficiencia del GI:
    varEfGiaux = sci2exp(varEfGI);
    execstr('PLtotal' + varEfGiaux + '= salidasGITotales(varEfGI,1)'); // Se asigna a cada variable PLtotal cada valor
correspondiente de la columna 1 de la matriz "salidasGITotales".
    execstr('STtotal' + varEfGiaux + '= salidasGITotales(varEfGI,4)'); // Se asigna a cada variable PLtotal cada valor
correspondiente de la columna 4 de la matriz "salidasGITotales".
    execstr('NGI(varEfGI)= PLtotal' + varEfGiaux + '/STtotal' + varEfGiaux); // Se calcula cada valor de la eficiencia del
GI utilizando cada valor correspondiente de las variables creadas en las líneas 5 y 6, y se guarda este valor en cada
posición respectiva del vector "NGI".
end; // Fin del lazo de iteración que comienza en la línea 3
execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) + '_'
+ fechayhora(5) + '_' + fechayhora(6) + '\EficienciaGI.bin', NGI)'); // Se guarda el vector "NGI" en el archivo
"EficienciaGI.bin" ubicado en la carpeta de la fecha y la hora de la simulación.
endfunction

```

Fuente: El autor.

Figura B.3 Algoritmo para el cálculo de la eficiencia del grupo generador

```

1 function Eficiencia_grupo // función para realizar el cálculo de la eficiencia del grupo generador.
2     execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
    '_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGITotales.bin', 'salidasGITotales'); // Desde el
directorio creado para la presente simulación, se cargan los valores de las salidas totales del GI, desde el vector
"salidasGITotales" guardado en el archivo "salidasGITotales.bin".
3     execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
    '_' + fechayhora(5) + '_' + fechayhora(6) + '\Potencia_Total.bin', 'PTotal'); // Desde el directorio creado
para la presente simulación, se cargan los valores de la potencia activa del GI, desde el vector "salidasGITotales"
guardado en el archivo "salidasGITotales.bin".
4     for varNgrupo = 1:nRL, // Rutina de iteración para calcular cada uno de los valores de la eficiencia del grupo
generador correspondientes a cada valor de la carga eléctrica del GI.
5         varNgrupoaux = sci2exp(varNgrupo); // Se define la variable "varNgrupoaux" como la versión string de la variable
varNgrupo.
6         execstr('PLtotal' + varNgrupoaux + '= salidasGITotales(varNgrupo,1)'); // Se crean las variables "PLtotal1" hasta
"STtotal[nRL]" es decir los valores de la potencia activa total en la carga.
7         execstr('Ngrupo(varNgrupo) = PLtotal' + varNgrupoaux + '/PTotal(varNgrupo)'); // A partir de cada valor de la
potencia activa total en la carga se calcula cada valor de la eficiencia del grupo generador, y se guardan estos valores
en el vector "Ngrupo".
8     end; // Fin de la rutina de iteración que comienza en la línea 4.
9     execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
    '_' + fechayhora(5) + '_' + fechayhora(6) + '\Eficiencia_grupo.bin', Ngrupo)'); // Se guarda el vector "Ngrupo" en
el archivo "Eficiencia_grupo.bin" y en el directorio creado para la presente simulación.
10    return; // Se termina la ejecución de esta función.
11 endfunction

```

Fuente: El autor

Figura B.4 Algoritmo para el cálculo del flujo de combustible del MCI

```

1 function Flujo_combustible// función para realizar el cálculo del flujo de combustible en un MCI.
   execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
   '_' + fechayhora(5) + '_' + fechayhora(6) + '\Potencia_Total.bin', 'PTotal'); // Desde el directorio creado
   para la presente simulación, se cargan los valores de la potencia total del MCI, desde el vector "PTotal" guardado en el
   archivo "Potencia_Total.bin".
   for varMf = 1:nRL, // Rutina de iteración para calcular cada uno de los valores del flujo de combustible
   correspondientes a cada valor de la carga eléctrica del GI.
       execstr('Mf(varMf) = PTotal(varMf)/Qf'); // Se calcula cada valor del flujo de combustible a partir de cada valor de
   la potencia total del MCI y el valor del poder calórico del combustible.
   end; // Fin de la rutina de iteración que comienza en la línea 3.
   execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
   '_' + fechayhora(5) + '_' + fechayhora(6) + '\Flujo_Combustible.bin', Mf); // Se guarda el vector "Mf" en el
   archivo "Flujo_Combustible.bin" y en el directorio creado para la presente simulación.
   return; // Se termina la ejecución de esta función.
endfunction

```

Fuente: El autor

Figura B.5 Algoritmo para el cálculo del flujo de aire del MCI

```

1 function FlujodeAire // funcion para calcular el flujo de aire.
2 Ma = nv*(2*Vd/Tiempos)*Denaireamb*(RPM/60); // Se calcula el valor del flujo de aire del MCI a partir de los valores de
   su eficiencia volumétrica, su cilindrada, sus tiempos, la densidad de aire en el ambiente y sus revoluciones por minuto.
3 execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) + '_' +
   fechayhora(5) + '_' + fechayhora(6) + '\Flujodeaire.bin', Ma); // Se guarda el valor de la variable "Ma" en el
   archivo "Flujodeaire.bin" y en el directorio creado para la presente simulación.
4 return; // Se termina la ejecución de esta función.
5 endfunction

```

Fuente: El autor

Figura B.6 Algoritmo para el cálculo de las variables eléctricas del GI

```

1  function GIpermanente // Función para determinar la tensión, la corriente y la potencia de transformación y en la
   carga eléctrica del GI cuando este está en régimen permanente.
2  global salidasGIperm; // Se define el tipo de la variable "salidasGIperm" como "global" para poder utilizarla en
   otras funciones.
3  s1 = 1-REM/(120*fe/polos); // Se define el deslizamiento del GI (de la fase 1 si es trifásico y de la red de
   secuencia positiva (del modelo bifásico) si es monofásico).
4  s2 = s1; // Se define el deslizamiento de la fase 2 del GI trifásico.
5  s3 = s2; // Se define el deslizamiento de la fase 3 del GI trifásico.
6  VL = VLmax; // Se define la tensión en la carga para el GI trifásico.
7  if fases == 1 then // Si el GI es monofásico, entonces:
8  global balance; // Se define la variable "balance" como "global" para poder utilizarla en otras funciones.
9  balance = 2; // Se le da un valor de "2" a la variable "balance".
10 s2 = 2 - s1; // Se calcula el deslizamiento de la red de secuencia negativa del circuito bifásico equivalente al
   GI monofásico.
11 VL = VLmax/2; // Se define la tensión en la carga para las dos redes (de secuencia positiva y de secuencia
   negativa) del circuito bifásico equivalente al GI monofásico.
12 varcarga = abs((RLf-RLi)/(nRL-1)); // Se calcula el valor de la diferencia entre dos valores consecutivos de la
   resistencia de la carga del GI.
13 if RLf > RLi then // Si el valor final de la resistencia de la carga del GI es mayor que el valor inicial de
   esta, entonces:
14 signo = '+'; // Se le asigna el valor string "+" a la variable "signo".
15 elseif RLf < RLi then // En cambio, si el valor final de la resistencia de la carga del GI es menor que el valor
   inicial de esta, entonces:
16 signo = '-'; // Se le asigna el valor string "-" a la variable "signo".
17 else // En cambio, si el valor final de la resistencia de la carga del GI es igual a el valor inicial de esta,
   entonces:
18 signo = '+'; // Se le asigna el valor string "+" a la variable "signo".
19 end; // Fin de la rutina de bifurcación que comienza en la línea 13.
20 for valvar = 1:nRL, // Rutina de iteración para calcular cada uno de los valores de la resistencia de la carga
21 valvaraux = sci2exp(valvar); // Se define la variable "valvaraux" como la versión tipo string de la variable
   "valvar", para utilizarla en la función "GIpermcargal" que es llamada en las línea 29.
22 if valvar == 1 then // Si la variable "valvar" tiene un valor numérico de "1" (la iteración que comienza en la
   línea 20 está en su primer ciclo), entonces:
23 mcararm1 = RLi; // Se define el valor de la variable "mcararm1" como el valor inicial de la resistencia de la
   carga del GI.
24 elseif valvar == nRL then // En cambio, si la variable "valvar" tiene un valor numérico igual al de la variable
   "nRL" (la iteración que comienza en la línea 20 está en su último ciclo), entonces:
25 mcararm1 = RLf; // Se define el valor de la variable "mcararm1" como el valor final de la resistencia de la
   carga del GI.
26 else // En cambio, si la variable "valvar" tiene un valor numérico diferente a "1" y al de la variable "nRL"
   (la iteración que comienza en la línea 20 está en un ciclo entre el primero y el último), entonces:
27 execstr('mcararm1 = mcararm1 + signo + 'varcarga'); // Se calcula el valor de la variable "mcararm1" como el
   valor de esta variable en el ciclo anterior más (o menos dependiendo del valor de la variable "signo" definida entre
   las líneas 13 y 19) más el valor de la variable "varcarga" definida en la línea 12.
28 end; // Fin de la rutina de bifurcación que comienza en la línea 22.
29 GIpermcargal; // Se ejecuta la función "GIpermcargal" que calcula los valores de tensión, corriente y potencia
   (de la carga del GI y de transformación) en régimen permanente para una carga lineal y resistiva.
30 end; // Fin de la rutina de iteración que comienza en la línea 20.
31 execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4)
   + '_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGIperm.bin',salidasGIperm)'); // Se guarda la matriz
   "salidasGIperm" (que contiene los valores de tensión, corriente y potencia calculados con la función "GIpermcargal")
   en el archivo "salidasGIperm.bin" ubicado en la carpeta cuyo nombre es la hora y fecha en que se realiza la
   simulación.
32 else // En cambio, si el GI es trifásico, entonces:
33 if balance == 1 then // Si el GI trifásico está balanceado, se calculan los valores de la carga del GI y se
   cargan las variables correspondientes:
34 varcarga = abs((RLf-RLi)/(nRL-1)); // Se calcula el valor de la diferencia entre dos valores consecutivos de la
   resistencia de la carga del GI.

```

```

35     if RLf > RLi then // Si el valor final de la resistencia de la carga del GI es mayor que el valor inicial de
        esta, entonces:
36         signo = '+'; // Se le asigna el valor string "+" a la variable "signo".
37     elseif RLf < RLi then // En cambio, si el valor final de la resistencia de la carga del GI es menor que el
        valor inicial de esta, entonces:
38         signo = '-'; // Se le asigna el valor string "-" a la variable "signo".
39     else // En cambio, si el valor final de la resistencia de la carga del GI es igual a el valor inicial de esta,
        entonces:
40         signo = '+'; // Se le asigna el valor string "+" a la variable "signo".
41     end; // Fin de la rutina de bifurcación que comienza en la línea 35.
42     for valvar = 1:nRL, // Rutina de iteración para calcular cada uno de los valores de la resistencia de la carga
        del GI y los respectivos valores de tensión, corriente y potencia (de la carga y de transformación) en régimen
        permanente:
43         valvaraux = sci2exp(valvar); // Se define la variable "valvaraux" como la versión tipo string de la variable
        "valvar", para utilizarla en la función "Gipermcargal" que es llamada en la línea 51.
44         if valvar == 1 then // Si la variable "valvar" tiene un valor numérico de "1" (la iteración que comienza
        en la línea 42 está en su primer ciclo), entonces:
45             mcararm1 = RLi; // Se define el valor de la variable "mcararm1" como el valor inicial de la resistencia de
        la carga del GI.
46         elseif valvar == nRL then // En cambio, si la variable "valvar" tiene un valor numérico igual al de la
        variable "nRL" (la iteración que comienza en la línea 42 está en su último ciclo), entonces:
47             mcararm1 = RLf; // Se define el valor de la variable "mcararm1" como el valor final de la resistencia de la
        carga del GI.
48         else // En cambio, si la variable "valvar" tiene un valor numérico diferente a "1" y al de la variable "nRL"
        (la iteración que comienza en la línea 42 está en un ciclo entre el primero y el último, excluyéndolos), entonces:
49             execstr('mcararm1 = mcararm1' + signo + 'varcarga'); // Se calcula el valor de la variable "mcararm1" como
        el valor de esta variable en el ciclo anterior más (o menos dependiendo del valor de la variable "signo" definida
        entre las líneas 35 y 41) más el valor de la variable "varcarga" definida en la línea 34.
50         end; // Fin de la rutina de bifurcación que comienza en la línea 44.
51     Gipermcargal; // Se ejecuta la función "Gipermcargal" que calcula los valores de tensión, corriente y
        potencia (de la carga del GI y de transformación) en régimen permanente para una carga lineal y resistiva.
52     end; // Fin de la rutina de iteración que comienza en la línea 42.
53     execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' +
        fechayhora(4) + '_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGiperm.bin',salidasGiperm)'); // Se guarda
        la matriz "salidasGiperm" (que contiene los valores de tensión, corriente y potencia calculados con la función
        "Gipermcargal") en el archivo "salidasGiperm.bin" ubicado en la carpeta cuyo nombre es la hora y fecha en que se
        realiza la simulación.
54     elseif balance == 3 then // Si el GI trifásico está desbalanceado, se calculan los valores de la carga del GI y
        se cargan las variables correspondientes:
55         varcargal = abs((RLf1-RLi1)/(nRL-1)); // Se calcula el valor de la diferencia entre dos valores consecutivos de
        la resistencia de la carga de la fase 1 del GI.
56         varcarga2 = abs((RLf2-RLi2)/(nRL-1)); // Se calcula el valor de la diferencia entre dos valores consecutivos de
        la resistencia de la carga de la fase 2 del GI.
57         varcarga3 = abs((RLf3-RLi3)/(nRL-1)); // Se calcula el valor de la diferencia entre dos valores consecutivos de
        la resistencia de la carga de la fase 3 del GI.
58         if RLf1 > RLi1 then // Si el valor final de la resistencia de la carga de la fase 1 del GI es mayor que el
        valor inicial de esta, entonces:
59             signo1 = '+'; // Se le asigna el valor string "+" a la variable "signo1".
60         elseif RLf1 < RLi1 then // En cambio, si el valor final de la resistencia de la carga de la fase 1 del GI es
        menor que el valor inicial de esta, entonces:
61             signo1 = '-'; // Se le asigna el valor string "-" a la variable "signo1".
62         else // En cambio, si el valor final de la resistencia de la carga de la fase 1 del GI es igual a el valor
        inicial de esta, entonces:
63             signo1 = '+'; // Se le asigna el valor string "+" a la variable "signo1".
64         end; // Fin de la rutina de bifurcación que comienza en la línea 58.
65         if RLf2 > RLi2 then // Si el valor final de la resistencia de la carga de la fase 2 del GI es mayor que el
        valor inicial de esta, entonces:
66             signo2 = '+'; // Se le asigna el valor string "+" a la variable "signo2".

```

```

67     elseif RLf2 < RLl2 then // En cambio, si el valor final de la resistencia de la carga de la fase 2 del GI es
menor que el valor inicial de esta, entonces:
68         signo2 = '-'; // Se le asigna el valor string "-" a la variable "signo2".
69     else // En cambio, si el valor final de la resistencia de la carga de la fase 2 del GI es igual a el valor
inicial de esta, entonces:
70         signo2 = '+'; // Se le asigna el valor string "+" a la variable "signo2".
71     end; // Fin de la rutina de bifurcación que comienza en la línea 65.
72     if RLf3 > RLl3 then // Si el valor final de la resistencia de la carga de la fase 3 del GI es mayor que el
valor inicial de esta, entonces:
73         signo3 = '+'; // Se le asigna el valor string "+" a la variable "signo3".
74     elseif RLf3 < RLl3 then // En cambio, si el valor final de la resistencia de la carga de la fase 3 del GI es
menor que el valor inicial de esta, entonces:
75         signo3 = '-'; // Se le asigna el valor string "-" a la variable "signo3".
76     else // En cambio, si el valor final de la resistencia de la carga de la fase 3 del GI es igual a el valor
inicial de esta, entonces:
77         signo3 = '+'; // Se le asigna el valor string "+" a la variable "signo3".
78     end; // Fin de la rutina de bifurcación que comienza en la línea 72.
79     for valvar = 1:nRL, // Rutina de iteración para calcular cada uno de los valores de la resistencia de la carga
del GI y los respectivos valores de tensión, corriente y potencia (de la carga y de transformación) en régimen
permanente:
80         valvaraux = sci2exp(valvar); // Se define la variable "valvaraux" como la versión tipo string de la variable
"valvar", para utilizarla en la función "GIpermargal" que es llamada en la línea 94.
81         if valvar == 1 then // Si la variable "valvar" tiene un valor numérico de "1" (la iteración que comienza en
la línea 79 está en su primer ciclo), entonces:
82             mcararm1 = RLl1; // Se define el valor de la variable "mcararm1" como el valor inicial de la resistencia de
la carga de la fase 1 del GI.
83             mcararm2 = RLl2; // Se define el valor de la variable "mcararm1" como el valor inicial de la resistencia de
la carga de la fase 2 del GI.
84             mcararm3 = RLl3; // Se define el valor de la variable "mcararm1" como el valor inicial de la resistencia de
85     elseif valvar == nRL then // En cambio, si la variable "valvar" tiene un valor numérico igual al de la
variable "nRL" (la iteración que comienza en la línea 79 está en su último ciclo), entonces:
86         mcararm1 = RLf1; // Se define el valor de la variable "mcararm1" como el valor final de la resistencia de la
carga de la fase 1 del GI.
87         mcararm2 = RLf2; // Se define el valor de la variable "mcararm1" como el valor final de la resistencia de la
carga de la fase 2 del GI.
88         mcararm3 = RLf3; // Se define el valor de la variable "mcararm1" como el valor final de la resistencia de la
carga de la fase 3 del GI.
89     else // En cambio, si la variable "valvar" tiene un valor numérico diferente a "1" y al de la variable "nRL"
(la iteración que comienza en la línea 79 está en un ciclo entre el primero y el último, excluyéndolos), entonces:
90         execstr('mcararm1 = mcararm1' + signo1 + 'varcargal'); // Se calcula el valor de la variable "mcararm1"
como el valor de esta variable en el ciclo anterior más (o menos dependiendo del valor de la variable "signo1"
definida entre las líneas 58 y 64) más el valor de la variable "varcargal" definida en la línea 55.
91         execstr('mcararm2 = mcararm2' + signo2 + 'varcarga2'); // Se calcula el valor de la variable "mcararm2"
como el valor de esta variable en el ciclo anterior más (o menos dependiendo del valor de la variable "signo2"
definida entre las líneas 65 y 71) más el valor de la variable "varcarga2" definida en la línea 56.
92         execstr('mcararm3 = mcararm3' + signo3 + 'varcarga3'); // Se calcula el valor de la variable "mcararm3"
como el valor de esta variable en el ciclo anterior más (o menos dependiendo del valor de la variable "signo3"
definida entre las líneas 72 y 78) más el valor de la variable "varcarga3" definida en la línea 57.
93     end; // Fin de la rutina de bifurcación que comienza en la línea 81.
94     GIpermargal; // Se ejecuta la función "GIpermargal" que calcula los valores de tensión, corriente y
potencia (de la carga del GI y de transformación) en régimen permanente para una carga lineal y resistiva.
95     end; // Fin de la rutina de iteración que comienza en la línea 79.
96     execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' +
fechayhora(4) + '_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGIperm.bin',salidasGIperm)'); // Se guarda
la matriz "salidasGIperm" (que contiene los valores de tensión, corriente y potencia calculados con la función
"GIpermargal") en el archivo "salidasGIperm.bin" ubicado en la carpeta cuyo nombre es la hora y fecha en que se
realiza la simulación.
97     end; // Fin de la rutina de bifurcación que comienza en la línea 33.
98 end; // Fin de la rutina de bifurcación que comienza en la línea 7.
99 endfunction

```

Fuente: El autor

Figura B.7 Algoritmo para el cálculo de las variables eléctricas del GI con carga lineal

```

1 function GIPermcargal //Función para calcular los parametros correspondientes al GI en régimen permanente con carga
eléctrica lineal.
2 global VT; // Se define el tipo de la variable "VT" como "global" para poder utilizarla en otras funciones.
3 global salidasGIPerm; // Se define el tipo de la variable "salidasGIPerm" como "global" para poder utilizarla en
otras funciones.
4 if fases == 1 then // Si el GI es monofásico, entonces:
5 mcararm2 = mcararm1; // Se define el valor de la variable "mcararm2" es decir el valor de la resistencia aplicada
a la red de secuencia negativa del modelo bifásico del GI monofásico.
6 end; // Fin de la rutina de bifurcación que comienza en la línea 4.
7 CIC = -i*VL*C*fe*2*pi; // Se halla la corriente compleja del capacitor:
8 // Si el sistema es monofásico la rutina de iteración siguiente se ejecuta dos veces (una para la red de
secuencia positiva y otra para la red de secuencia negativa del modelo bifásico del GI monofásico, de modo que la
potencia monofásica total será la suma de las potencias de secuencia positiva y de secuencia negativa), si es
trifásico balanceado el lazo siguiente se ejecuta una sola vez y las potencias trifásicas serán simplemente el triple
de las potencias halladas, de lo contrario (si el GI es trifásico desbalanceado) se deberá ejecutar el lazo tres
veces y calcular la potencias trifásicas mediante un algoritmo que tiene en cuenta el desbalance del sistema:
9 for m = 1:balance, // Rutina de iteración para calcular cada uno de los valores de tensión, corriente y potencia
de la carga y de transformación en régimen permanente.
10 maux = sci2exp(m); // Se define la variable "maux" como el valor string de la variable numérica "m".
11 execstr('CVT = VL + (VL/mcararm' + maux + '- i*VL*C*fe*2*pi)*(R1 + i*fe*2*pi*L1) + (VL/mcararm' + maux +
'- i*VL*C*fe*2*pi + ((VL/mcararm' + maux + '- i*VL*C*fe*2*pi)*(R1 + i*fe*2*pi*L1) + VL)*(1/Rh +
1/(i*fe*2*pi*Lm)))*(R2 + i*fe*2*pi*L2)'); // Se halla la tensión de transformación (compleja).
12 MVT = sqrt(real(CVT)^2+imag(CVT)^2); // Se halla la tensión de transformación máxima.
13 VT = MVT/sqrt(2); // Se halla el valor RMS de la tensión de transformación.
14 // Se hallan los parametros con la fase referida a la tensión de la carga:
15 execstr('IL = VL/mcararm' + maux); // Se halla la corriente compleja de carga.
16 CI2 = IL+CIC+(VL+(IL+CIC)*(R1+i*2*pi*fe*L1))/(i*Rh*Lm*2*pi*fe/(Rh+i*Lm*2*pi*fe)); // Se halla la corriente
compleja de transformación.
17 MI2 = sqrt(real(CI2)**2 + imag(CI2)**2); // Se hallan la magnitud de la corriente de transformación.
18 FI2 = asin(imag(CI2)/MI2); // Se hallan la fase de la corriente de transformación.
19 // Se determinan los valores RMS de los parámetros de transformación y de carga de la fase y sus respectivas
potencias activa y reactiva:
20 I2 = MI2/sqrt(2); // Se halla el valor RMS de la corriente de transformación.
21 VL = VL/sqrt(2); // Se halla el valor RMS de la tensión de la carga.
22 execstr('I2f' + maux + '= I2'); // Se guarda el valor RMS de la corriente de transformación en la variable
I2f[maux].
23 execstr('FI2f' + maux + '= FI2'); // Se guarda la fase de la corriente de transformación en la variable
FI2f[maux].
24 IL = IL/sqrt(2); // Se halla el valor RMS de la corriente de la carga.
25 execstr('ILf' + maux + '= IL'); // Se guarda el valor RMS de la corriente de la carga del GI en la variable
ILf[maux].
26 execstr('PT' + maux + '= I2*VT*cos(FI2)'); // Se halla las potencia activa de transformación de la fase o
secuencia correspondiente y se guarda este valor en la variable PT[maux].
27 execstr('PL' + maux + '= IL*VL'); // Se halla las potencia activa de carga de la fase o secuencia
correspondiente y se guarda este valor en la variable PL[maux].
28 execstr('QT' + maux + '= I2*VT*sin(FI2)'); // Se halla la potencia reactiva de transformación de la fase o
secuencia correspondiente.
29 execstr('ST' + maux + '= I2*VT'); // Se halla la potencia de dimensionamiento de transformación de la fase o
secuencia correspondiente.
30 end; // Fin de la rutina de iteración que comienza en la línea 9.
31 // Dependiendo de si el sistema es monofásico o trifásico, se hallan las potencias totales mediante el algoritmo
correspondiente:
32 if fases == 1 then // Si el sistema es monofásico, entonces:
33 execstr('I2' + valvaraux + '= I2f1+I2f2'); // Se halla la corriente de total transformación del GI y se guarda
este valor en la variable I2[valvaraux].
34 execstr('IL' + valvaraux + '= ILf1+ILf2'); // Se halla la corriente total de la carga del GI y se guarda este
valor en la variable IL[valvaraux].
35 execstr('PL' + valvaraux + '= PL1+PL2'); // Se halla la potencia activa total de la carga del GI y se guarda este

```

```

36   execstr('PT'+ valvaraux + '= PT1+PT2'); // Se halla la potencia activa total de transformación del GI y se guarda
    este valor en la variable PT[valvaraux].
37   execstr('QBT'+ valvaraux + '= QT1+QT2'); // Se halla la potencia reactiva total de transformación del GI y se
    guarda este valor en la variable QBT[valvaraux].
38   execstr('SAT'+ valvaraux + '= ST1+ST2'); // Se halla la potencia de dimensionamiento aritmética de transformación
    del GI y se guarda este valor en la variable SAT[valvaraux].
39   execstr('SVT'+ valvaraux + '= sqrt(PT'+ valvaraux + '**2+QBT'+ valvaraux + '**2)'); // Se halla la potencia de
    dimensionamiento vectorial de transformación del GI y se guarda este valor en la variable SVT[valvaraux].
40   execstr('salidasGIperm(valvar,1:7) = [I2' + valvaraux + ', IL' + valvaraux + ', PL' + valvaraux + ', PT' +
    valvaraux + ', QBT' + valvaraux + ', SAT' + valvaraux + ', SVT' + valvaraux + ']'); // Se guardan las variables
    mencionadas en las líneas de la 33 a la 39 en las ubicaciones desde ([valvar],1) hasta ([valvar],7) de la matriz
    "salidasGIperm".
41   else // En cambio, si el sistema es trifásico, entonces:
42     // Dependiendo de si el sistema es balanceado o no, se hallan las potencias trifásicas mediante el algoritmo
    correspondiente:
43     if balance == 1 then // Si el sistema está balanceado, entonces:
44       execstr('I2'+ valvaraux + '= I2f1'); // Se guarda el valor de la corriente de transformación de la fase 1 del
        GI en la variable I2[valvaraux].
45       execstr('IL'+ valvaraux + '= ILf1'); // Se guarda el valor de la corriente de la carga de la fase 1 del GI en la
        variable IL[valvaraux].
46       execstr('PL'+ valvaraux + '= 3*PL1'); // Se halla la potencia activa trifásica de la carga del GI y se guarda
        este valor en la variable PL[valvaraux].
47       execstr('PT'+ valvaraux + '= 3*PT1'); // Se halla la potencia activa trifásica de transformación del GI y se
        guarda este valor en la variable PT[valvaraux].
48       execstr('QBT'+ valvaraux + '= 3*QBT1'); // Se halla la potencia reactiva trifásica de Budeanu de transformación
        del GI y se guarda este valor en la variable QBT[valvaraux].
49       execstr('SAT'+ valvaraux + '= 3*ST1'); // Se halla la potencia de dimensionamiento aritmética de transformación
        del GI y se guarda este valor en la variable SAT[valvaraux].
50       execstr('SVT'+ valvaraux + '= sqrt(PT'+ valvaraux + '**2+QBT'+ valvaraux + '**2)'); // Se halla la potencia de
51   execstr('salidasGIperm(valvar,1:7) = [I2' + valvaraux + ', IL' + valvaraux + ', PL' + valvaraux + ', PT' +
    valvaraux + ', QBT' + valvaraux + ', SAT' + valvaraux + ', SVT' + valvaraux + ']'); // Se guardan las variables
    mencionadas en las líneas de la 44 a la 50 en las ubicaciones desde ([valvar],1) hasta ([valvar],7) de la matriz
    "salidasGIperm".
52   else // En cambio, si el sistema está desbalanceado, entonces:
53     CI2SP = (1/3)*((I2f1*cos(FI2f1) - 0.5*(I2f2*cos(FI2f2+2*pi/3)+I2f3*cos(FI2f3-2*pi/3)) -
    (sqrt(3)/2)*(I2f2*sin(FI2f2+2*pi/3)- I2f3*sin(FI2f3-2*pi/3))) + %i*(I2f1*sin(FI2f1) -
    0.5*(I2f2*sin(FI2f2+2*pi/3)+I2f3*sin(FI2f3-2*pi/3)) + (sqrt(3)/2)*(I2f2*cos(FI2f2+2*pi/3)-
    I2f3*cos(FI2f3-2*pi/3))); // Se halla la corriente compleja de transformación de secuencia positiva.
54     MI2SP = sqrt(real(CI2SP)^2 + imag(CI2SP)^2); // Se halla la magnitud de la corriente de transformación de
    secuencia positiva.
55     FI2SPSR = asin(imag(CI2SP)/MI2SP); // Se halla la fase sin referenciar de la corriente de transformación de
    secuencia positiva.
56     if real(CI2SP)<0 then // Si la parte real de la corriente compleja de transformación de secuencia positiva es
        menor que "0", entonces:
57       if imag(CI2SP)<0 then // Si la parte imaginaria de la corriente compleja de transformación de secuencia
        positiva es menor que "0", entonces:
58         FI2SPSR = %pi + FI2SPSR; // Se redefine la fase sin referenciar de la corriente de transformación de
        secuencia positiva ubicándola con pi radianes de adelanto.
59       else // En cambio, si la parte imaginaria de la corriente compleja de transformación de secuencia positiva es
        mayor que "0", entonces:
60         FI2SPSR = %pi - FI2SPSR; // Se redefine la fase sin referenciar de la corriente de transformación de
        secuencia positiva ubicándola con pi menos 2*FI2SPSR radianes de adelanto.
61     end; // Fin de la rutina de bifurcación que comienza en la línea 57.
62     end; // Fin de la rutina de bifurcación que comienza en la línea 56.
63     CI2SPAR = (1/3)*((I2f1*cos(FI2f1) - 0.5*(I2f2*cos(FI2f2)+I2f3*cos(FI2f3))) +
    %i*((-sqrt(3)/2)*(I2f2*cos(FI2f2)-I2f3*cos(FI2f3)))); // Se halla la corriente compleja que define la referencia que
    se usa para hallar la verdadera fase de la corriente compleja de transformación de secuencia positiva.
64     MI2SPAR = sqrt(real(CI2SPAR)^2 + imag(CI2SPAR)^2); // Se halla la magnitud de la corriente compleja hallada en

```

```

65     FI2SPAR = asin(imag(CI2SPAR)/MI2SPAR); // Se halla la fase de referencia para hallar la verdadera fase de la
corriente compleja de transformación de secuencia positiva.
66     if real(CI2SPAR)<0 then // Si la parte real de la corriente compleja de referencia es menor que cero, entonces:
67         if imag(CI2SPAR)<0 then // Si la parte imaginaria de la corriente compleja de referencia es menor que cero,
entonces:
68             FI2SPAR = %pi + FI2SPAR; // Se redefine la fase de la corriente de referencia ubicándola con pi radianes de
adelanto.
69         else // En cambio, si la parte imaginaria de la corriente compleja de referencia es mayor que cero, entonces:
70             FI2SPAR = %pi - FI2SPAR; // Se redefine la fase sin referenciar de la corriente de transformación de
secuencia positiva ubicándola con pi menos 2*FI2SPAR radianes de adelanto.
71         end; // Fin de la rutina de bifurcación que comienza en la línea 67.
72     end; // Fin de la rutina de bifurcación que comienza en la línea 66.
73     FI2SP = FI2SPSR - FI2SPAR; // Se halla la fase referenciada de la corriente de transformación de secuencia
positiva.
74     MILSP = (1/3)*sqrt((ILf1 - 0.5*(ILf2+ILf3))^2 + ((-sqrt(3)/2)*(ILf2-ILf3))^2); // Se halla la corriente de
secuencia positiva de la carga.
75     PTtrif = PT1+PT2+PT3; // Se halla la potencia activa trifásica de transformación.
76     PLtrif = PL1+PL2+PL3; // Se halla la potencia activa trifásica de la carga.
77     QBTtrif = QT1+QT2+QT3; // Se halla la potencia reactiva de Budeanu trifásica de transformación.
78     PTPtrif = 3*MI2SP*VT*cos(FI2SP); // Se halla la potencia activa trifásica de secuencia positiva de
transformación.
79     QTSPtrif = 3*MI2SP*VT*sin(FI2SP); // Se halla la potencia reactiva trifásica de secuencia positiva de
transformación.
80     STSP = 3*MILSP*VL; // Se halla la potencia de dimensionamiento trifásica de secuencia positiva de
transformación.
81     SLSP = 3*MILSP*VL; // Se halla la potencia de dimensionamiento trifásica de secuencia positiva de la carga.
82     PLSPtrif = 3*MILSP*VL; // Se halla la potencia activa trifásica de secuencia positiva de la carga.
83     SATtrif = ST1+ST2+ST3; // Se halla la potencia de dimensionamiento aritmética trifásica de transformación.
84     SVTtrif = sqrt(PTtrif**2+QBTtrif**2); // Se halla la potencia de dimensionamiento vectorial trifásica de
transformación.
85     if conf_estat == 'delta' then // Si los devanados del estator están conectados en delta, entonces:
86         execstr('I2E'+ valvaraux + '= sqrt((1/3)*(I2f1^2+I2f2^2+I2f3^2))'); // Se halla la corriente equivalente (de
un sistema trifásico balanceado que tendría el mismo consumo de potencia del sistema desbalanceado) de transformación
y se guarda este valor en la variable I2E[valvaraux].
87         execstr('STE'+ valvaraux + '= 3*I2E*VT'); // Se halla la potencia de dimensionamiento trifásica equivalente
de transformación y se guarda este valor en la variable STE[valvaraux].
88         execstr('STU'+ valvaraux + '= sqrt(STE^2-STSP^2)'); // Se halla la potencia de dimensionamiento trifásica de
desbalance de transformación y se guarda este valor en la variable STU[valvaraux].
89     else // En cambio, si los devanados del estator están conectados en estrella, entonces:
90         if conf_estat_estrella == 'trifilar' then // Si el estator en estrella es trifilar, entonces:
91             execstr('I2E'+ valvaraux + '= sqrt((1/3)*(I2f1^2+I2f2^2+I2f3^2))'); // Se halla la corriente equivalente
(de un sistema trifásico balanceado que tendría el mismo consumo de potencia del sistema desbalanceado) de
transformación y se guarda este valor en la variable I2E[valvaraux].
92             execstr('STE'+ valvaraux + '= 3*I2E*VT'); // Se halla la potencia de dimensionamiento trifásica equivalente
de transformación y se guarda este valor en la variable STE[valvaraux].
93             execstr('STU'+ valvaraux + '= sqrt(STE^2-STSP^2)'); // Se halla la potencia de dimensionamiento trifásica
de desbalance de transformación y se guarda este valor en la variable STU[valvaraux].
94         else // En cambio, si el estator en estrella es tetrafilar, entonces:
95             execstr('InE'+ valvaraux + '= sqrt((I2f1*cos(FI2f1)+I2f2*cos(FI2f2)+I2f3*cos(FI2f3))^2
+(I2f1*sin(FI2f1)+I2f2*sin(FI2f2)+I2f3*sin(FI2f3))^2)'); // Se halla la corriente del neutro del estator y se guarda
este valor en la variable InE[valvaraux].
96             execstr('PnE'+ valvaraux + '= RnE*InE^2'); // Se halla la potencia del neutro del estator y se guarda este
valor en la variable PnE[valvaraux].
97             execstr('I2E'+ valvaraux + '= sqrt((1/3)*(I2f1^2+I2f2^2+I2f3^2+InE^2))'); // Se halla la corriente
equivalente (de un sistema trifásico balanceado que tendría el mismo consumo de potencia del sistema desbalanceado)
de transformación y se guarda este valor en la variable I2E[valvaraux].
98             execstr('STE'+ valvaraux + '= 3*I2E*VT'); // Se halla la potencia de dimensionamiento trifásica equivalente
de transformación y se guarda este valor en la variable STE[valvaraux].
99             execstr('STU'+ valvaraux + '= sqrt(STE^2-STSP^2)'); // Se halla la potencia de dimensionamiento trifásica

```

```

100     end; // Fin de la rutina de bifurcación que comienza en la línea 90.
101     end; // Fin de la rutina de bifurcación que comienza en la línea 85.
102     if conf_carga == 'delta' then // Si la carga está conectada en delta, entonces:
103         execstr('ILE'+ valvaraux + ' ILE = sqrt((1/3)*(ILf1^2+ILf2^2+ILf3^2)); // Se halla la corriente equivalente
(de un sistema trifásico balanceado que tendría el mismo consumo de potencia del sistema desbalanceado) de la carga y
se guarda este valor en la variable ILE[valvaraux].
104         execstr('SLE'+ valvaraux + ' = 3*ILE*VT'); // Se halla la potencia de dimensionamiento trifásica equivalente
de la carga y se guarda este valor en la variable SLE[valvaraux].
105         execstr('SLU'+ valvaraux + ' = sqrt(SLE^2-SLSP^2)'); // Se halla la potencia de dimensionamiento trifásica de
desbalance de la carga y se guarda este valor en la variable SLU[valvaraux].
106     else // En cambio, si la carga está conectada en estrella, entonces:
107         if conf_carga_estrella == 'trifilar' then // Si la carga en estrella es trifilar, entonces:
108             execstr('ILE'+ valvaraux + ' = sqrt((1/3)*(ILf1^2+ILf2^2+ILf3^2)); // Se halla la corriente equivalente
(de un sistema trifásico balanceado que tendría el mismo consumo de potencia del sistema desbalanceado) de la carga y
se guarda este valor en la variable ILE[valvaraux].
109             execstr('SLE'+ valvaraux + ' = 3*ILE*VT'); // Se halla la potencia de dimensionamiento trifásica equivalente
de la carga y se guarda este valor en la variable SLE[valvaraux].
110             execstr('SLU'+ valvaraux + ' = sqrt(SLE^2-SLSP^2)'); // Se halla la potencia de dimensionamiento trifásica
de desbalance de la carga y se guarda este valor en la variable SLU[valvaraux].
111         else // En cambio, si la carga en estrella es tetrafilar, entonces:
112             execstr('InL'+ valvaraux + ' = sqrt((ILf1*cos(0)+ILf2*cos(2*pi/3)+ILf3*cos(2*pi/3)^2
+ILf2*sin(2*pi/3)+ILf3*sin(-2*pi/3)^2)); // Se halla la corriente del neutro de la carga y se guarda este valor
en la variable InL[valvaraux].
113             execstr('PnL'+ valvaraux + ' = RnL*InL^2'); // Se halla la potencia del neutro de la carga y se guarda este
valor en la variable PnL[valvaraux].
114             execstr('ILE'+ valvaraux + ' = sqrt((1/3)*(ILf1^2+ILf2^2+ILf3^2+InL^2)); // Se halla la corriente
equivalente (de un sistema trifásico balanceado que tendría el mismo consumo de potencia del sistema desbalanceado)
de la carga y se guarda este valor en la variable ILE[valvaraux].
115             execstr('SLE'+ valvaraux + ' = 3*ILE*VL'); // Se halla la potencia de dimensionamiento trifásica equivalente
116             execstr('SLU'+ valvaraux + ' = sqrt(SLE^2-SLSP^2)'); // Se halla la potencia de dimensionamiento trifásica
de desbalance de la carga y se guarda este valor en la variable SLU[valvaraux].
117         end; // Fin de la rutina de bifurcación que comienza en la línea 107.
118     end; // Fin de la rutina de bifurcación que comienza en la línea 102.
119     execstr('I2f1'+ valvaraux + ' = I2f1'); // Se guarda el valor de la corriente de transformación de la fase 1 del
GI en la variable I2f1[valvaraux].
120     execstr('ILf1'+ valvaraux + ' = ILf1'); // Se guarda el valor de la corriente de la carga de la fase 1 del GI en
la variable ILf1[valvaraux].
121     execstr('I2f2'+ valvaraux + ' = I2f2'); // Se guarda el valor de la corriente de transformación de la fase 2 del
GI en la variable I2f2[valvaraux].
122     execstr('ILf2'+ valvaraux + ' = ILf2'); // Se guarda el valor de la corriente de la carga de la fase 2 del GI en
la variable ILf2[valvaraux].
123     execstr('I2f3'+ valvaraux + ' = I2f3'); // Se guarda el valor de la corriente de transformación de la fase 3 del
GI en la variable I2f3[valvaraux].
124     execstr('ILf3'+ valvaraux + ' = ILf3'); // Se guarda el valor de la corriente de la carga de la fase 3 del GI en
la variable ILf3[valvaraux].
125     execstr('PT'+ valvaraux + ' = PTtrif'); // Se guarda el valor de la potencia activa trifásica de transformación
del GI en la variable PT[valvaraux].
126     execstr('PL'+ valvaraux + ' = PLtrif'); // Se guarda el valor de la potencia trifásica de la carga del GI en la
variable PL[valvaraux].
127     execstr('QBT'+ valvaraux + ' = QBTtrif'); // Se guarda el valor de la potencia reactiva trifásica de Budeanu de
transformación del GI en la variable PT[valvaraux].
128     execstr('SAT'+ valvaraux + ' = SATtrif'); // Se guarda el valor de la potencia de dimensionamiento aritmética
trifásica de transformación del GI en la variable SAT[valvaraux].
129     execstr('SVT'+ valvaraux + ' = SVTtrif'); // Se guarda el valor de la potencia de dimensionamiento vectorial
trifásica de transformación del GI en la variable SVT[valvaraux].
130     if conf_estat == 'delta' & conf_carga == 'delta' then // Si tanto los devanados del estator como la carga del
GI están conectados en delta, entonces:
131         execstr('salidasGIperm(valvar,1:17) = [I2f1' + valvaraux + ', ILf1' + valvaraux + ', I2f2' + valvaraux + ',

```

```

ILF2' + valvaraux + ', I2f3' + valvaraux + ', ILf3' + valvaraux + ', PL' + valvaraux + ', PT' + valvaraux + ', OBT' +
valvaraux + ', SAT' + valvaraux + ', SVT' + valvaraux + ', I2E' + valvaraux + ', STE' + valvaraux + ', STU' + valvaraux
+ ', ILE' + valvaraux + ', SLE' + valvaraux + ', SLU' + valvaraux + ']); // Se guardan los valores definidos en las
lineas 86 a 88, 103 a 105 y 119 a 129 en las posiciones de ([valvar],1) hasta ([valvar],17) de la matriz
"salidasGIperm".
132 elseif conf_estat == 'estrella' & conf_carga == 'delta' then // En cambio, si los devanados del estator del GI
están conectados en estrella y la carga del GI está conectada en delta, entonces:
133 if conf_estat_estrella == 'trifilar' then // Si el estator en estrella es trifilar, entonces:
134   execstr('salidasGIperm(valvar,1:17) = [I2f1' + valvaraux + ', ILf1' + valvaraux + ', I2f2' + valvaraux + ',
ILf2' + valvaraux + ', I2f3' + valvaraux + ', ILf3' + valvaraux + ', PL' + valvaraux + ', PT' + valvaraux + ', OBT' +
valvaraux + ', SAT' + valvaraux + ', SVT' + valvaraux + ', I2E' + valvaraux + ', STE' + valvaraux + ', STU' + valvaraux
+ ', ILE' + valvaraux + ', SLE' + valvaraux + ', SLU' + valvaraux + ']); // Se guardan los valores definidos en las
lineas 91 a 93, 103 a 105 y 119 a 129 en las posiciones de ([valvar],1) hasta ([valvar],17) de la matriz
"salidasGIperm".
135 else // En cambio, si el estator en estrella es tetrafilar, entonces:
136   execstr('salidasGIperm(valvar,1:19) = [I2f1' + valvaraux + ', ILf1' + valvaraux + ', I2f2' + valvaraux + ',
ILf2' + valvaraux + ', I2f3' + valvaraux + ', ILf3' + valvaraux + ', PL' + valvaraux + ', PT' + valvaraux + ', OBT' +
valvaraux + ', SAT' + valvaraux + ', SVT' + valvaraux + ', I2E' + valvaraux + ', STE' + valvaraux + ', STU' + valvaraux
+ ', ILE' + valvaraux + ', SLE' + valvaraux + ', SLU' + valvaraux + ', InE' + valvaraux + ', PnE' + valvaraux + ']); //
Se guardan los valores definidos en las líneas 95 a 99, 103 a 105 y 119 a 129 en las posiciones de ([valvar],1) hasta
([valvar],19) de la matriz "salidasGIperm".
137 end; // Fin de la rutina de bifurcación que comienza en la línea 133.
138 elseif conf_estat == 'delta' & conf_carga == 'estrella' then // En cambio, si los devanados del estator del GI
están conectados en delta y la carga del GI está conectada en estrella, entonces:
139 if conf_carga_estrella == 'trifilar' then // Si la carga en estrella es trifilar, entonces:
140   execstr('salidasGIperm(valvar,1:17) = [I2f1' + valvaraux + ', ILf1' + valvaraux + ', I2f2' + valvaraux + ',
ILf2' + valvaraux + ', I2f3' + valvaraux + ', ILf3' + valvaraux + ', PL' + valvaraux + ', PT' + valvaraux + ', OBT' +
valvaraux + ', SAT' + valvaraux + ', SVT' + valvaraux + ', I2E' + valvaraux + ', STE' + valvaraux + ', STU' + valvaraux
+ ', ILE' + valvaraux + ', SLE' + valvaraux + ', SLU' + valvaraux + ']); // Se guardan los valores definidos en las
lineas 86 a 88, 108 a 110 y 119 a 129 en las posiciones de ([valvar],1) hasta ([valvar],17) de la matriz
"salidasGIperm".
141 else // En cambio, si la carga en estrella es tetrafilar, entonces:
142   execstr('salidasGIperm(valvar,1:19) = [I2f1' + valvaraux + ', ILf1' + valvaraux + ', I2f2' + valvaraux + ',
ILf2' + valvaraux + ', I2f3' + valvaraux + ', ILf3' + valvaraux + ', PL' + valvaraux + ', PT' + valvaraux + ', OBT' +
valvaraux + ', SAT' + valvaraux + ', SVT' + valvaraux + ', I2E' + valvaraux + ', STE' + valvaraux + ', STU' + valvaraux
+ ', ILE' + valvaraux + ', SLE' + valvaraux + ', SLU' + valvaraux + ', InL' + valvaraux + ', PnL' + valvaraux + ']); //
Se guardan los valores definidos en las líneas 86 a 88, 112 a 116 y 119 a 129 en las posiciones de ([valvar],1) hasta
([valvar],19) de la matriz "salidasGIperm".
143 end; // Fin de la rutina de bifurcación que comienza en la línea 139.
144 elseif conf_estat == 'estrella' & conf_carga == 'estrella' then // En cambio, si tanto los devanados del
estator como la carga del GI están conectados en estrella, entonces:
145 if conf_estat_estrella == 'trifilar' & conf_carga_estrella == 'trifilar' then // Si tanto el estator en
estrella como la carga en estrella del GI son trifilares, entonces:
146   execstr('salidasGIperm(valvar,1:17) = [I2f1' + valvaraux + ', ILf1' + valvaraux + ', I2f2' + valvaraux + ',
ILf2' + valvaraux + ', I2f3' + valvaraux + ', ILf3' + valvaraux + ', PL' + valvaraux + ', PT' + valvaraux + ', OBT' +
valvaraux + ', SAT' + valvaraux + ', SVT' + valvaraux + ', I2E' + valvaraux + ', STE' + valvaraux + ', STU' + valvaraux
+ ', ILE' + valvaraux + ', SLE' + valvaraux + ', SLU' + valvaraux + ']); // Se guardan los valores definidos en las
lineas 91 a 93, 108 a 110 y 119 a 129 en las posiciones de ([valvar],1) hasta ([valvar],17) de la matriz
"salidasGIperm".
147 elseif conf_estat_estrella == 'tetrafilar' & conf_carga_estrella == 'trifilar' then // En cambio, si el
estator en estrella es tetrafilar y la carga en estrella es trifilar, entonces:
148   execstr('salidasGIperm(valvar,1:19) = [I2f1' + valvaraux + ', ILf1' + valvaraux + ', I2f2' + valvaraux + ',
ILf2' + valvaraux + ', I2f3' + valvaraux + ', ILf3' + valvaraux + ', PL' + valvaraux + ', PT' + valvaraux + ', OBT' +
valvaraux + ', SAT' + valvaraux + ', SVT' + valvaraux + ', I2E' + valvaraux + ', STE' + valvaraux + ', STU' + valvaraux
+ ', ILE' + valvaraux + ', SLE' + valvaraux + ', SLU' + valvaraux + ', InE' + valvaraux + ', PnE' + valvaraux + ']); //
Se guardan los valores definidos en las líneas 95 a 99, 108 a 110 y 119 a 129 en las posiciones de ([valvar],1) hasta
([valvar],19) de la matriz "salidasGIperm".
149 elseif conf_estat_estrella == 'trifilar' & conf_carga_estrella == 'tetrafilar' then // En cambio, si el

```

```

estator en estrella es trifilar y la carga en estrella es tetrafilar, entonces:
150     execstr('salidasGIperm(valvar,1:19) = [I2f1' + valvaraux + ', ILf1' + valvaraux + ', I2f2' + valvaraux + ',
ILf2' + valvaraux + ', I2f3' + valvaraux + ', ILf3' + valvaraux + ', PL' + valvaraux + ', PT' + valvaraux + ', OBT' +
valvaraux + ', SAT' + valvaraux + ', SVT' + valvaraux + ', I2E' + valvaraux + ', STE' + valvaraux + ', STU' + valvaraux
+ ', ILE' + valvaraux + ', SLE' + valvaraux + ', SLU' + valvaraux + ', InL' + valvaraux + ', PnL' + valvaraux + ']'); //
Se guardan los valores definidos en las líneas 91 a 93, 112 a 116 y 119 a 129 en las posiciones de ([valvar],1) hasta
([valvar],19) de la matriz "salidasGIperm".
151     elseif conf_estat_estrella == 'tetrafilar' & conf_carga_estrella == 'tetrafilar' then // Si tanto el estator
en estrella como la carga en estrella del GI son tetrafilares, entonces:
152     execstr('salidasGIperm(valvar,1:21) = [I2f1' + valvaraux + ', ILf1' + valvaraux + ', I2f2' + valvaraux + ',
ILf2' + valvaraux + ', I2f3' + valvaraux + ', ILf3' + valvaraux + ', PL' + valvaraux + ', PT' + valvaraux + ', OBT' +
valvaraux + ', SAT' + valvaraux + ', SVT' + valvaraux + ', I2E' + valvaraux + ', STE' + valvaraux + ', STU' + valvaraux
+ ', ILE' + valvaraux + ', SLE' + valvaraux + ', SLU' + valvaraux + ', InE' + valvaraux + ', PnE' + valvaraux + ', InL'
+ valvaraux + ', PnL' + valvaraux + ']'); // Se guardan los valores definidos en las líneas 95 a 99, 112 a 116 y 119
a 129 en las posiciones de ([valvar],1) hasta ([valvar],21) de la matriz "salidasGIperm".
153     end; // Fin de la rutina de bifurcación que comienza en la línea 145.
154     end; // Fin de la rutina de bifurcación que comienza en la línea 130.
155     end; // Fin de la rutina de bifurcación que comienza en la línea 43.
156     end; // Fin de la rutina de bifurcación que comienza en la línea 32.
157 endfunction

```

Fuente: El autor

Figura B.8 Algoritmo para el cálculo de las potencias totales del GI

```

1 function GITotal // Función para calcular la potencia total consumida por el GI.
2     execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
'_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGIperm.bin','salidasGIperm'); // Se cargan desde el
archivo "salidasGIperm.bin" (ubicado en la carpeta correspondiente a la fecha y hora de la simulación) las variables
eléctricas en régimen permanente de salida del GI guardadas en la variable "salidasGIperm".
3     execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
'_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGItran.bin','salidasGItran'); // Se cargan desde el
archivo "salidasGItran.bin" (ubicado en la carpeta correspondiente a la fecha y hora de la simulación) las variables
eléctricas en régimen transitorio de salida del GI guardadas en la variable "salidasGItran".
4     for vartotal = 1:nRL, // Rutina de iteración para calcular las potencias totales del GI.
5         vartotalaux = sci2exp(vartotal);
6         execstr('SAT' + vartotalaux + '= salidasGIperm(vartotal,6)'); // Se define la variable "SAT" como el valor
correspondiente a la potencia de dimensionamiento aritmética total del GI en régimen permanente guardada en la
variable "salidasGIperm".
7         execstr('PT' + vartotalaux + '= salidasGIperm(vartotal,4)'); // Se define la variable "PT" como el valor
correspondiente a la potencia activa total del GI en régimen permanente guardada en la variable "salidasGIperm".
8         execstr('PL' + vartotalaux + '= salidasGIperm(vartotal,3)'); // Se define la variable "PL" como el valor
correspondiente a la potencia activa de la carga del GI en régimen permanente guardada en la variable
"salidasGIperm".
9         execstr('OBT' + vartotalaux + '= salidasGIperm(vartotal,5)'); // Se define la variable "OBT" como el valor
correspondiente a la potencia reactiva total del GI en régimen permanente guardada en la variable "salidasGIperm".
10        execstr('STtran' + vartotalaux + '= salidasGItran(vartotal,4)'); // Se define la variable "STtran" como el valor
correspondiente a la potencia de dimensionamiento transitoria total del GI en régimen transitorio guardada en la
variable "salidasGItran".
11        execstr('PTtran' + vartotalaux + '= salidasGItran(vartotal,2)'); // Se define la variable "PTtran" como el valor
correspondiente a la potencia activa transitoria total del GI en régimen transitorio guardada en la variable
"salidasGItran".
12        execstr('PLtran' + vartotalaux + '= salidasGItran(vartotal,1)'); // Se define la variable "PLtran" como el valor
correspondiente a la potencia activa transitoria de la carga del GI en régimen transitorio guardada en la variable

```

```

"salidasGITran".
13   execstr('QTtran' + vartotalaux + '= salidasGITran(vartotal,3)'); // Se define la variable "QTtran" como el valor
correspondiente a la potencia reactiva transitoria total del GI en régimen transitorio guardada en la variable
"salidasGITran".
14   execstr('STtotal' + vartotalaux + '= SAT' + vartotalaux + '+' STtran' + vartotalaux); // Se calcula la potencia de
dimensionamiento total del GI como la suma de la potencia de dimensionamiento aritmética total en régimen permanente
más la potencia de dimensionamiento total en régimen transitorio.
15   execstr('PTtotal' + vartotalaux + '= PT' + vartotalaux + '+' PTtran' + vartotalaux); // Se calcula la potencia
activa del GI como la suma de la potencia activa total en régimen permanente más la potencia activa total en régimen
transitorio.
16   execstr('PLtotal' + vartotalaux + '= PL' + vartotalaux + '+' PLtran' + vartotalaux); // Se calcula la potencia
activa de la carga del GI como la suma de la potencia activa de la carga en régimen permanente más la potencia activa
de la carga en régimen transitorio.
17   execstr('QTtotal' + vartotalaux + '= QBT' + vartotalaux + '+' QTtran' + vartotalaux); // Se calcula la potencia
reactiva total del GI como la suma de la potencia reactiva total en régimen permanente más la potencia reactiva total
en régimen transitorio.
18   execstr('salidasGITotales(' + vartotalaux + ',1:4) = [PLtotal' + vartotalaux + ', PTtotal' + vartotalaux + ',
QTtotal' + vartotalaux + ', STtotal' + vartotalaux + ']'); // Se guardan en el vector "salidasGITotales" los valores
de las variables hallados desde la línea 14 hasta la 17.
19 end; // Fin de la rutina de iteración que comienza en la línea 4.
20 execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
 '_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGITotales.bin', salidasGITotales)'); // Se guarda el
vector "salidasGITotales" en el archivo "salidasGITotales.bin" ubicándolo en la carpeta correspondiente a la fecha y
hora de la simulación.
21 endfunction

```

Fuente: El autor

Figura B.9 Algoritmo para el cálculo de las variables eléctricas del GI en régimen transitorio con carga lineal

```

1 function GItrancargal // Función para calcular los parámetros correspondientes al GI en régimen transitorio con carga
eléctrica lineal.
2   global VT; // Se define el tipo de la variable "VT" como "global" para poder utilizarla en otras funciones.
3   global salidasGITran; // Se define el tipo de la variable "salidasGITran" como "global" para poder utilizarla en
otras funciones.
4   ff = poly([(2*pi*fe)**2 0 1], 's', 'coeff'); // Se define el polinomio "ff" (el cuál es la respuesta forzada del
circuito equivalente del GI trifásico) con los siguientes coeficientes: "(2*pi*fe)^2" para "s^0", "0" para "s^1" y
"1" para "s^2".
5   gf = roots(ff); // Se define el vector "gf" cuyos componentes son las raíces del polinomio "ff".
6   gf1 = gf; // Se define el vector "gf1" (raíces de la respuesta forzada de la fase 1, si el GI es trifásico).
7   gf2 = gf; // Se define el vector "gf2" (raíces de la respuesta forzada de la fase 2, si el GI es trifásico).
8   gf3 = gf; // Se define el vector "gf3" (raíces de la respuesta forzada de la fase 3, si el GI es trifásico).
9   if fases == 1 then // Si el GI es monofásico, entonces:
10    balance = 2; // Se le da el valor de "2" a la variable "balance" para que el siguiente lazo se ejecute dos veces:
11    ff1 = poly([(2*pi*(RPM*polos/120 - fe))**2 0 1], 's', 'coeff'); // Se define el polinomio "ff1" (el cuál es la
respuesta forzada de la red de secuencia negativa del circuito equivalente bifásico del GI monofásico) con los
siguientes coeficientes: "(2*pi*(RPM*polos/120 - fe))^2" para "s^0", "0" para "s^1" y "1" para "s^2".
12    gf1 = roots(ff1); // Se define el vector "gf1" (raíces de la respuesta forzada de la red de secuencia negativa
del circuito equivalente bifásico del GI monofásico).
13    ff2 = poly([(2*pi*(RPM*polos/120 + fe))**2 0 1], 's', 'coeff'); // Se define el polinomio "ff2" (el cuál es la
respuesta forzada de la red de secuencia positiva del circuito equivalente bifásico del GI monofásico) con los
siguientes coeficientes: "(2*pi*(RPM*polos/120 + fe))^2" para "s^0", "0" para "s^1" y "1" para "s^2".
14    gf2 = roots(ff2); // Se define el vector "gf2" (raíces de la respuesta forzada de la red de secuencia positiva
del circuito equivalente bifásico del GI monofásico).
15    VTmax = VT*sqrt(2)/2; // Se halla el valor máximo de la tensión de transformación.
16    mcararm1 = mcararm; // Se define la carga de la red de secuencia negativa del circuito equivalente bifásico del
GI monofásico.
17    mcararm2 = mcararm; // Se define la carga de la red de secuencia positiva del circuito equivalente bifásico del
GI monofásico.

```

```

18 else // En cambio, si el GI es trifásico:
19     Vmax = VT*sqrt(2); // Se halla el valor máximo de la tensión de transformación.
20 end; // Fin de la rutina de bifurcación que comienza en la línea 4.
21 // Si el GI es trifásico y está desbalanceado, se hallan los valores transitorios para cada fase ejecutándose el
siguiente lazo 3 veces, en el caso en el que el GI sea trifásico y su carga esté balanceada, se hallarán los valores
transitorios correspondientes corriendo el lazo siguiente una sola vez:
22 for o = 1:balance, // Rutina de iteración para calcular cada uno de los valores de tensión, corriente y potencia
de la carga y de transformación en régimen transitorio.
23     oaux = sci2exp(o); // Se define la variable "oaux" como el valor string de la variable numérica "o".
24     if balance == 1 then // Si el GI es trifásico y está balanceado, entonces:
25         execstr('mcararm' + oaux + '= mcararm'); // Se define el valor de la carga de cada fase como el valor guardado
en la variable "mcararm".
26     end; // Fin de la rutina de bifurcación que comienza en la línea 24.
27     // En las líneas 28 a 32, se definen los coeficientes del denominador de las funciones de transferencia (es el
mimp para todas, pues estas sólo difieren en su numerador):
28     execstr('a = C*L1*Im*mcararm' + oaux + '*L2/Rh');
29     execstr('b = C*L1*Im*mcararm' + oaux + '*L2/Rh + C*L1*mcararm' + oaux + '*R2 + C*mcararm' + oaux + '*L1*Im +
C*mcararm' + oaux + '*L2 + C*R1*Im*mcararm' + oaux + '*L2/Rh + L1*L2*Im/Rh');
30     execstr('c = R2*C*mcararm' + oaux + '*L1 + L2*R1*Im/Rh + L2*mcararm' + oaux + '*Lm/Rh + L1*Im + L1*R2*Im/Rh +
L1*L2 + R1*C*Im*mcararm' + oaux + '*R1*R2*mcararm' + oaux + '*C*Lm/Rh + L2*R1*mcararm' + oaux + '*C +
R2*C*Im*mcararm' + oaux + '*L2*Im');
31     execstr('d = R2*L1 + R1*R2*C*mcararm' + oaux + '*R1 + R1*R2/Rh + R1*L2 + Lm*mcararm' + oaux + '*R2*Im*mcararm'
+ oaux + '*Lm/Rh + L2*mcararm' + oaux + '*R2*Im');
32     execstr('e = R1*R2 + R2*mcararm' + oaux);
33     f = poly([e d c b a], "s", "coeff"); // Con los coeficientes hallados en las líneas 29 a 33 se define el polinomio
del denominador de las funciones de transferencia.
34     g = roots(f); // Se determinan las raíces que definen la respuesta natural del circuito equivalente del GI y se
guardan estos valores en el vector "g".
35     // En las líneas 36 v 37, se agrega al vector de raíces "g" la respuesta forzada del circuito equivalente del GI.
para poder hallar las constantes que dependen de las condiciones iniciales:
36     execstr('g(5) = gf' + oaux + '(1)');
37     execstr('g(6) = gf' + oaux + '(2)');
38     for n = 1:6, // Rutina de iteración para aproximar a cero los valores muy pequeños de las raíces halladas en las
líneas 12, 14 y 34:
39         if abs(real(g(n))) < 1D-16 then // Si el valor de la parte real de una raíz es menor que 1*10^-16, entonces:
40             g(n) = imag(g(n))*%i, // El valor de la raíz se hace igual a la parte imaginaria de la raíz.
41         end, // Fin de la rutina de bifurcación que comienza en la línea 39.
42         if abs(imag(g(n))) < 1D-16 then // Si el valor de la parte imaginaria de una raíz es menor que 1*10^-16,
entonces:
43             g(n) = real(g(n)), // El valor de la raíz se hace igual a la parte real de la raíz.
44         end, // Fin de la rutina de bifurcación que comienza en la línea 42.
45     end; // Fin de la rutina de iteración que comienza en la línea 38.
46     // En las líneas 47 a 52, se definen los coeficientes de los polinomios de las fracciones parciales que son
utilizadas para hallar la respuesta transitoria del circuito equivalente del GI:
47     h1 = coeff(poly([g(2) g(3) g(4) g(5) g(6)], "s"));
48     h2 = coeff(poly([g(1) g(3) g(4) g(5) g(6)], "s"));
49     h3 = coeff(poly([g(1) g(2) g(4) g(5) g(6)], "s"));
50     h4 = coeff(poly([g(1) g(2) g(3) g(5) g(6)], "s"));
51     h5 = coeff(poly([g(1) g(2) g(3) g(4) g(6)], "s"));
52     h6 = coeff(poly([g(1) g(2) g(3) g(4) g(5)], "s"));
53     for m = 1:6, // Rutina de iteración para aproximar a cero los valores muy pequeños de los coeficientes hallados
en las líneas 47 a 52, en este caso "m" representa el índice del vector "h[m]":
54         maux = sci2exp(m); // Se define la variable "maux" como la versión tipo string de la variable numérica "m".
55         for n = 1:6, // Rutina de iteración para aproximar a cero los valores muy pequeños de los coeficientes hallados
en las líneas 47 a 52, en este caso "n" representa un componente del vector "h[m]", es decir "h[m]([n])":
56             naux = sci2exp(n); // Se define la variable "naux" como la versión tipo string de la variable numérica "n".
57             execstr('if abs(real(h' + maux + '(' + naux + ')) < 1D-16 then h' + maux + '(' + naux + ') = imag(h' + maux +
 '(' + naux + '))*%i; end;'), // Si el valor de la parte real de un coeficiente es menor que 1*10^-16, entonces, el

```

```

valor del coeficiente se hace igual a la parte imaginaria de la raiz.
58     execstr('if abs(imag(h' + maux + '(' + naux + ')) < 1D-16 then h' + maux + '(' + naux + ') = real(h' + maux +
'(' + naux + '); end;'), // Si el valor de la parte imaginaria de un coeficiente es menor que 1*10^-16, entonces, el
valor del coeficiente se hace igual a la parte real de la raiz.
59     execstr('if abs(imag(h' + maux + '(' + naux + ')) < 1D-16 & abs(real(h' + maux + '(' + naux + ')) < 1D-16
then h' + maux + '(' + naux + ') = 0; end;'), // Si tanto el valor de la parte imaginaria de un coeficiente como el
valor de su parte imaginaria son menores que 1*10^-16, entonces, el valor del coeficiente se hace igual a cero.
60     end; // Fin de la rutina de iteración que comienza en la línea 55.
61     end; // Fin de la rutina de iteración que comienza en la línea 53.
62     A = [h1(1) h2(1) h3(1) h4(1) h5(1) h6(1) ; h1(2) h2(2) h3(2) h4(2) h5(2) h6(2) ; h1(3) h2(3) h3(3) h4(3) h5(3)
h6(3) ; h1(4) h2(4) h3(4) h4(4) h5(4) h6(4) ; h1(5) h2(5) h3(5) h4(5) h5(5) h6(5) ; h1(6) h2(6) h3(6) h4(6) h5(6)
h6(6)]; // Se define la matriz "A" compuesta por los valores de los coeficientes definidos en las líneas 47 a 52.
63     [U,S,V]=svd(A); // Se descompone la matriz "A" produciendo la matriz diagonal "S" (de la misma dimensión de "A" y
con elementos de la diagonal no negativos y en orden decreciente) y dos matrices ortogonales o unitarias "U" y "V" de
modo tal que A = U*S*V'.
64     for n = 1:6, // Rutina de iteración para crear una matriz "S1" cuyos elementos son los recíprocos de los
respectivos elementos de "S".
65         S1(n,n) = 1/S(n,n); // Se define cada elemento de "S1" como el recíproco del respectivo valor de "S"
66     end; // Fin de la rutina de iteración que comienza en la línea 64.
67     U1 = [U']; // Se define U1 como la matriz traspuesta de la matriz <U>.
68     Ainv = V*S1*U1; // A partir de las matrices halladas en las líneas 63, 65 y 67, se halla la matriz inversa de "A".
69     BL = [0 0 Lm 0 0 0]'; // Se define el vector de los coeficientes del numerador de la función de transferencia de
la corriente de la carga (ordenados desde la menor potencia a la mayor potencia de <s>).
70     execstr('BC = [0 0 0 Lm*C*mcaram' + oaux + ' 0 0]'''); // Se define el vector de los coeficientes del numerador
de la función de transferencia de la corriente del capacitor (ordenados desde la menor potencia a la mayor potencia
de s).
71     execstr('BM = [0 (R1+mcaram' + oaux + ') (L1+R1*C*mcaram' + oaux + ') (L1*C*mcaram' + oaux + ') 0 0]'''); //
Se define el vector de los coeficientes del numerador de la función de transferencia de la corriente de magnetización
(ordenaos desde la menor potencia a la mayor potencia de s).
72     execstr('BH = [0 0 Lm*(R1+mcaram' + oaux + ')/(Rh) Lm*(L1+R1*C*mcaram' + oaux + ')/(Rh) Lm*(L1*C*mcaram' +
oaux + ')/(Rh) 0]'''); // Se define el vector de los coeficientes del numerador de la función de transferencia de la
corriente de pérdidas en el entrehierro (ordenados desde la menor potencia a la mayor potencia de s).
73     XL = VTmax*Ainv*BL; // Se definen los valores de los numeradores de las fracciones parciales de la función de
transferencia de la carga y se guardan en el vector "XL".
74     XC = VTmax*Ainv*BC; // Se definen los valores de los numeradores de las fracciones parciales de la función de
transferencia del capacitor y se guardan en el vector "XC".
75     XM = VTmax*Ainv*BM; // Se definen los valores de los numeradores de las fracciones parciales de la función de
transferencia de la inductancia de magnetización y se guardan en el vector "XM".
76     XH = VTmax*Ainv*BH; // Se definen los valores de los numeradores de las fracciones parciales de la función de
transferencia de la resistencia de pérdidas en el entrehierro y se guardan en el vector "XH".
77     for maux = ['L' 'C' 'M' 'H'], // Rutina de iteración para aproximar a cero los valores muy pequeños de los
numeradores de las fracciones parciales guardados en los vectores "XL", "XC", "XM", "XH":
78         for n = 1:6, // Rutina de iteración para aproximar a cero los valores muy pequeños de cada numerador guardado
en los vectores "XL", "XC", "XM", "XH":
79             naux = sci2exp(n); // Se define la variable "naux" como la versión tipo string de la variable numérica "n".
80             execstr('if abs(real(X' + maux + '(' + naux + ')) < 1D-16 then X' + maux + '(' + naux + ') = imag(X' + maux +
'(' + naux + '))*&i; end;'), // Si el valor de la parte imaginaria de un numerador es menor que 1*10^-16, entonces,
el valor del coeficiente se hace igual a la parte real de la raiz.
81             execstr('if abs(imag(X' + maux + '(' + naux + ')) < 1D-16 then X' + maux + '(' + naux + ') = real(X' + maux +
'(' + naux + '); end;'), // Si el valor de la parte real de un numerador es menor que 1*10^-16, entonces, el valor
del coeficiente se hace igual a la parte imaginaria de la raiz.
82             end; // Fin de la rutina de iteración que comienza en la línea 78.
83             end; // Fin de la rutina de iteración que comienza en la línea 77.
84             XLi = imag(XL); // Se define el vector "XLi" cuyos componentes son las partes imaginarias de las componentes del
vector "XL".
85             XCi = imag(XC); // Se define el vector "XCi" cuyos componentes son las partes imaginarias de las componentes del
vector "XC".
86             XMi = imag(XM); // Se define el vector "XMi" cuyos componentes son las partes imaginarias de las componentes del

```

```

vector "XM".
87   XHi = imag(XH); // Se define el vector "XHi" cuyos componentes son las partes imaginarias de las componentes del
vector "XH".
88   XL = real(XL); // Se redefine el vector "XL" para que sus componentes sean las partes reales de las componentes
del vector "XL" original.
89   XC = real(XC); // Se redefine el vector "XC" para que sus componentes sean las partes reales de las componentes
del vector "XC" original.
90   XM = real(XM); // Se redefine el vector "XM" para que sus componentes sean las partes reales de las componentes
del vector "XM" original.
91   XH = real(XH); // Se redefine el vector "XH" para que sus componentes sean las partes reales de las componentes
del vector "XH" original.
92   gi = imag(g); // Se define el vector "gi" cuyos componentes son las partes imaginarias de las componentes del
vector "g".
93   g = real(g); // Se redefine el vector "g" para que sus componentes sean las partes reales de las componentes del
vector "g" original.
94   tao = 1/min(abs(g(1:4))); // Se define la variable "tao" como la constante de tiempo máxima de la respuesta
transitoria del GI.
95   EL = 0; // Se inicializa en "0" la variable "EL".
96   EC = 0; // Se inicializa en "0" la variable "EC".
97   EM = 0; // Se inicializa en "0" la variable "EM".
98   EH = 0; // Se inicializa en "0" la variable "EH".
99   ER1 = 0; // Se inicializa en "0" la variable "ER1".
100  EL1 = 0; // Se inicializa en "0" la variable "EL1".
101  ER2 = 0; // Se inicializa en "0" la variable "ER2".
102  EL2 = 0; // Se inicializa en "0" la variable "EL2".
103  for n = 1:4, // Rutina de iteración para, junto con la rutina de iteración que comienza en la línea 104, hallar
los valores de la energía por unidad de impedancia.
104  for m = 1:4, // Rutina de iteración para, junto con la rutina de iteración que comienza en la línea 103,
hallar los valores de la energía por unidad de impedancia.
105      EL = EL +
((XLi(m)*XLi(n)-XL(m)*XL(n))*(g(m)+g(n))+2*XL(m)*XLi(n)*(gi(m)+gi(n)))/((g(m)+g(n))**2+(gi(m)+gi(n))**2) -
((XLi(m)*XLi(n)+XL(m)*XL(n))*(g(m)+g(n))+2*XL(m)*XLi(n)*(gi(n)-gi(m)))/((g(m)+g(n))**2+(gi(m)-gi(n))**2), // Se
calcula la energía por unidad de impedancia consumida por la carga del GI, multiplicada por 2.
106      EC = EC +
1/(g(m)**2+gi(m)**2)*(((XC(m)*XC(n)-XCi(m)*XCi(n))*g(m)+(XCi(m)*XC(n)-XC(m)*XCi(n))*gi(m))*(g(m)+g(n))/((g(m)+g(n))**2
+(gi(m)+gi(n))**2) +
((XC(m)*XC(n)-XCi(m)*XCi(n))*g(m)+(XCi(m)*XC(n)-XC(m)*XCi(n))*gi(m))*(g(m)+g(n))/((g(m)+g(n))**2+(gi(m)-gi(n))**2) +
((XC(m)*XCi(n)+XCi(m)*XC(n))*g(m)+(XC(m)*XC(n)-XCi(m)*XCi(n))*gi(m))*(gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)
+
((XC(m)*XCi(n)-XCi(m)*XC(n))*g(m)-(XC(m)*XC(n)+XCi(m)*XCi(n))*gi(m))*(gi(m)-gi(n))/((g(m)+g(n))**2+(gi(m)-gi(n))**2)
+ (XCi(m)*gi(m)-XC(m)*g(m))*(XCi(n)*gi(n)-XCi(n)*g(n))/(g(n)**2+gi(n)**2)), // Se calcula la energía por unidad de
impedancia consumida por el capacitor de arranque del GI, multiplicada por 2.
107      EM = EM +
(XM(m)*XMi(n)*g(m)+XMi(m)*XMi(n)*gi(m))*((gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(gi(m)-gi(n))/((g(m)+g(n))**2
+(gi(m)-gi(n))**2)) +
(XMi(m)*XM(n)*g(m)-XM(m)*XM(n)*gi(m))*((gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(gi(m)-gi(n))/((g(m)+g(n))**2+(
gi(m)-gi(n))**2)) +
(XMi(m)*XMi(n)*g(m)-XM(m)*XMi(n)*gi(m))*((g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(g(m)+g(n))/((g(m)+g(n))**2+(gi
(m)-gi(n))**2)) -
(XM(m)*XM(n)*g(m)+XMi(m)*XM(n)*gi(m))*((g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(g(m)-g(n))/((g(m)+g(n))**2+(gi(m)
)-gi(n))**2)), // Se calcula la energía por unidad de impedancia consumida por el proceso de magnetización del GI,
multiplicada por 2.
108      EH = EH +
((XHi(m)*XHi(n)-XH(m)*XH(n))*(g(m)+g(n))+2*XH(m)*XHi(n)*(gi(m)+gi(n)))/((g(m)+g(n))**2+(gi(m)+gi(n))**2) -
((XHi(m)*XHi(n)+XH(m)*XH(n))*(g(m)+g(n))+2*XH(m)*XHi(n)*(gi(n)-gi(m)))/((g(m)+g(n))**2+(gi(m)-gi(n))**2), // Se
calcula la energía por unidad de impedancia consumida debido a las pérdidas en el entrehierro del GI, multiplicada
por 2.
109      ER1 = ER1 +

```

```

((XCi(m)*XCi(n)-XC(m)*XC(n))*(g(m)+g(n))+2*XC(m)*XCi(n)*(gi(m)+gi(n)))/((g(m)+g(n))**2+(gi(m)+gi(n))**2) -
((XCi(m)*XCi(n)+XC(m)*XC(n))*(g(m)+g(n))+2*XC(m)*XCi(n)*(gi(n)-gi(m)))/((g(m)+g(n))**2+(gi(m)-gi(n))**2) +
((XLi(m)*XLi(n)-XL(m)*XL(n))*(g(m)+g(n))+2*XL(m)*XLi(n)*(gi(m)+gi(n)))/((g(m)+g(n))**2+(gi(m)+gi(n))**2) -
((XLi(m)*XLi(n)+XL(m)*XL(n))*(g(m)+g(n))+2*XL(m)*XLi(n)*(gi(n)-gi(m)))/((g(m)+g(n))**2+(gi(m)-gi(n))**2), // Se
calcula la energía por unidad de impedancia consumida por la resistencia del estator del GI, multiplicada por 2.
110 EL1 = EL1 +
(XC(m)*XCi(n)*g(n)+XCi(m)*XCi(n)*gi(n))*(gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(gi(m)-gi(n))/((g(m)+g(n))**2
+(gi(m)-gi(n))**2) +
(XC(m)*XCi(n)*g(n)-XC(m)*XC(n)*gi(n))*(gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(gi(m)-gi(n))/((g(m)+g(n))**2+(
gi(m)-gi(n))**2) +
(XCi(m)*XCi(n)*g(n)-XC(m)*XCi(n)*gi(n))*(g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(g(m)+g(n))/((g(m)+g(n))**2+(gi
(m)-gi(n))**2) -
(XC(m)*XC(n)*g(n)+XCi(m)*XC(n)*gi(n))*(g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(g(m)+g(n))/((g(m)+g(n))**2+(gi(m
)-gi(n))**2) +
(XL(m)*XLi(n)*g(n)+XLi(m)*XLi(n)*gi(n))*(gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(gi(m)-gi(n))/((g(m)+g(n))**2
+(gi(m)-gi(n))**2) +
(XLi(m)*XLi(n)*g(n)-XL(m)*XL(n)*gi(n))*(gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(gi(m)-gi(n))/((g(m)+g(n))**2+(
gi(m)-gi(n))**2) +
(XLi(m)*XLi(n)*g(n)-XL(m)*XLi(n)*gi(n))*(g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(g(m)+g(n))/((g(m)+g(n))**2+(gi
(m)-gi(n))**2) -
(XL(m)*XL(n)*g(n)+XLi(m)*XL(n)*gi(n))*(g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(g(m)+g(n))/((g(m)+g(n))**2+(gi(m
)-gi(n))**2), // Se calcula la energía por unidad de impedancia consumida por la inductancia del estator del GI,
multiplicada por 2.
111 ER2 = ER2 +
((XMi(m)*XMi(n)-XM(m)*XM(n))*(g(m)+g(n))+2*XM(m)*XMi(n)*(gi(m)+gi(n)))/((g(m)+g(n))**2+(gi(m)+gi(n))**2) -
((XMi(m)*XMi(n)+XM(m)*XM(n))*(g(m)+g(n))+2*XM(m)*XMi(n)*(gi(n)-gi(m)))/((g(m)+g(n))**2+(gi(m)-gi(n))**2) +
((XHi(m)*XHi(n)-XH(m)*XH(n))*(g(m)+g(n))+2*XH(m)*XHi(n)*(gi(m)+gi(n)))/((g(m)+g(n))**2+(gi(m)+gi(n))**2) -
((XHi(m)*XHi(n)+XH(m)*XH(n))*(g(m)+g(n))+2*XH(m)*XHi(n)*(gi(n)-gi(m)))/((g(m)+g(n))**2+(gi(m)-gi(n))**2) +
((XCi(m)*XCi(n)-XC(m)*XC(n))*(g(m)+g(n))+2*XC(m)*XCi(n)*(qi(m)+qi(n)))/((g(m)+g(n))**2+(qi(m)+qi(n))**2) -
((XCi(m)*XCi(n)+XC(m)*XC(n))*(g(m)+g(n))+2*XC(m)*XCi(n)*(gi(n)-gi(m)))/((g(m)+g(n))**2+(gi(m)-gi(n))**2) +
((XLi(m)*XLi(n)-XL(m)*XL(n))*(g(m)+g(n))+2*XL(m)*XLi(n)*(gi(m)+gi(n)))/((g(m)+g(n))**2+(gi(m)+gi(n))**2) -
((XLi(m)*XLi(n)+XL(m)*XL(n))*(g(m)+g(n))+2*XL(m)*XLi(n)*(gi(n)-gi(m)))/((g(m)+g(n))**2+(gi(m)-gi(n))**2), // Se
calcula la energía por unidad de impedancia consumida por la resistencia del rotor del GI, multiplicada por 2.
112 EL2 = EL2 +
(XM(m)*XMi(n)*g(n)+XMi(m)*XMi(n)*gi(n))*(gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(gi(m)-gi(n))/((g(m)+g(n))**2
+(gi(m)-gi(n))**2) +
(XMi(m)*XMi(n)*g(n)-XM(m)*XM(n)*gi(n))*(gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(gi(m)-gi(n))/((g(m)+g(n))**2+(
gi(m)-gi(n))**2) +
(XMi(m)*XMi(n)*g(n)-XM(m)*XMi(n)*gi(n))*(g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(g(m)+g(n))/((g(m)+g(n))**2+(gi
(m)-gi(n))**2) -
(XM(m)*XM(n)*g(n)+XMi(m)*XM(n)*gi(n))*(g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(g(m)+g(n))/((g(m)+g(n))**2+(gi(m
)-gi(n))**2) +
(XH(m)*XHi(n)*g(n)+XHi(m)*XHi(n)*gi(n))*(gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(gi(m)-gi(n))/((g(m)+g(n))**2
+(gi(m)-gi(n))**2) +
(XHi(m)*XHi(n)*g(n)-XH(m)*XH(n)*gi(n))*(gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(gi(m)-gi(n))/((g(m)+g(n))**2+(
gi(m)-gi(n))**2) +
(XHi(m)*XHi(n)*g(n)-XH(m)*XHi(n)*gi(n))*(g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(g(m)+g(n))/((g(m)+g(n))**2+(gi
(m)-gi(n))**2) -
(XH(m)*XH(n)*g(n)+XHi(m)*XH(n)*gi(n))*(g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(g(m)+g(n))/((g(m)+g(n))**2+(gi(m
)-gi(n))**2) +
(XC(m)*XCi(n)*g(n)+XCi(m)*XCi(n)*gi(n))*(gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(gi(m)-gi(n))/((g(m)+g(n))**2
+(gi(m)-gi(n))**2) +
(XCi(m)*XC(n)*g(n)-XC(m)*XC(n)*gi(n))*(gi(m)+gi(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(gi(m)-gi(n))/((g(m)+g(n))**2+(
gi(m)-gi(n))**2) +
(XCi(m)*XCi(n)*g(n)-XC(m)*XCi(n)*gi(n))*(g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)-(g(m)+g(n))/((g(m)+g(n))**2+(gi
(m)-gi(n))**2) -
(XC(m)*XC(n)*g(n)+XCi(m)*XC(n)*gi(n))*(g(m)+g(n))/((g(m)+g(n))**2+(gi(m)+gi(n))**2)+(g(m)+g(n))/((g(m)+g(n))**2+(gi(m
)-gi(n))**2) +

```

```

(XL(m)*XLi(n)*g(n)+XLi(m)*XLi(n)*gi(n))*((g(m)+g(n))/(g(m)+g(n))^2+(g(m)+gi(n))^2)-(g(m)-gi(n))/(g(m)+g(n))^2
+(g(m)-gi(n))^2) +
(XLi(m)*XL(n)*g(n)-XL(m)*XL(n)*gi(n))*((g(m)+g(n))/(g(m)+g(n))^2+(g(m)+gi(n))^2+(g(m)-gi(n))/(g(m)+g(n))^2+(
gi(m)-gi(n))^2) +
(XLi(m)*XLi(n)*g(n)-XL(m)*XLi(n)*gi(n))*((g(m)+g(n))/(g(m)+g(n))^2+(g(m)+gi(n))^2)-(g(m)+g(n))/(g(m)+g(n))^2+(gi
(m)-gi(n))^2) -
(XL(m)*XL(n)*g(n)+XLi(m)*XL(n)*gi(n))*((g(m)+g(n))/(g(m)+g(n))^2+(g(m)+gi(n))^2+(g(m)+g(n))/(g(m)+g(n))^2+(gi(m)
)-gi(n))^2), // Se calcula la energía por unidad de impedancia consumida por la inductancia del rotor del GI,
multiplicada por 2.
113     end, // Fin de la rutina de iteración que comienza en la línea 104.
114     end; // Fin de la rutina de iteración que comienza en la línea 103.
115     execstr('PLtran = mcararm' + oaux + '*EL/(10*tao)'); // Se calcula la potencia activa transitoria de la carga como
la potencia consumida en esta cuando han pasado un tiempo de 5 veces la constante de tiempo máxima del GI (calculada
en la línea 94).
116     PTtran = PLtran + ( Rh*EH + R2*ER2 + R1*ER1 )/(10*tao); // Se calcula la potencia activa transitoria total del GI
como la suma de las potencias consumidas en la carga, en la resistencia del rotor y en la resistencia del estator del
GI, cuando han pasado un tiempo de 5 veces la constante de tiempo máxima del GI (calculada en la línea 94).
117     QTtran = ( Lm*EM + EC/C + L2*EL2 + L1*EL1 )/(10*tao); // Se calcula la potencia reactiva transitoria total del GI
como la suma de las potencias reactivas en la inductancia de magnetización, en el capacitor de arranque, en la
inductancia del rotor y en la inductancia del estator del GI, cuando han pasado un tiempo de 5 veces la constante de
tiempo máxima del GI (calculada en la línea 94).
118     STtran = sqrt(PTtran**2 + QTtran**2); // Se calcula la Potencia de dimensionamiento total del GI como el la raiz
del la suma de los cuadrados de las potencias totales activa y reactiva.
119     execstr('PLtran' + oaux + '= PLtran'), // Se guarda en la variable "PLtran[oaux]" el valor presente en la variable
"PLtran" de modo tal que se forma una lista de variables con la potencia de la carga en cada fase del GI.
120     execstr('PTtran' + oaux + '= PTtran'), // Se guarda en la variable "PTtran[oaux]" el valor presente en la variable
"PTtran" de modo tal que se forma una lista de variables con la potencia activa total en cada fase del GI.
121     execstr('QTtran' + oaux + '= QTtran'), // Se guarda en la variable "QTtran[oaux]" el valor presente en la variable
"QTtran" de modo tal que se forma una lista de variables con la potencia reactiva total en cada fase del GI.
122     execstr('STtran' + oaux + '= STtran'), // Se guarda en la variable "STtran[oaux]" el valor presente en la variable
"STtran" de modo tal que se forma una lista de variables con la potencia de dimensionamiento total en cada fase del
GI.
123     end; // Fin de la rutina de iteración que comienza en la línea 22.
124     if fases == 3 then // Si el GI es trifásico, entonces:
125         if balance == 1 then // Si el GI trifásico está balanceado, entonces:
126             PLtran = 3*PLtran1; // La potencia activa trifásica de la carga se halla multiplicando por 3 el valor de la
potencia activa de la carga monofásica.
127             PTtran = 3*PTtran1; // La potencia activa trifásica total se halla multiplicando por 3 el valor de la
potencia activa total monofásica.
128             QTtran = 3*QTtran1; // La potencia reactiva trifásica total se halla multiplicando por 3 el valor de la
potencia reactiva total monofásica.
129             STtran = sqrt(PTtran**2 + QTtran**2); // se define la potencia de dimensionamiento trifásica total como la
raiz cuadrada de la suma del cuadrado de las potencias trifásicas activa total y reactiva total.
130             execstr('PLtran'+ valvaraux + '= PLtran'); // Se guarda en la variable "PLtran[valvaraux]" el valor presente
en la variable "PLtran" de modo tal que se forma una lista de variables de cada valor de la potencia activa trifásica
en la carga correspondiente a cada valor de resistencia de la carga ingresado por el usuario.
131             execstr('PTtran'+ valvaraux + '= PTtran'); // Se guarda en la variable "PTtran[valvaraux]" el valor presente
en la variable "PTtran" de modo tal que se forma una lista de variables de cada valor de la potencia activa trifásica
total correspondiente a cada valor de resistencia de la carga ingresado por el usuario.
132             execstr('QTtran'+ valvaraux + '= QTtran'); // Se guarda en la variable "QTtran[valvaraux]" el valor presente
en la variable "QTtran" de modo tal que se forma una lista de variables de cada valor de la potencia reactiva
trifásica total correspondiente a cada valor de resistencia de la carga ingresado por el usuario.
133             execstr('STtran'+ valvaraux + '= STtran'); // Se guarda en la variable "STtran[valvaraux]" el valor presente
en la variable "STtran" de modo tal que se forma una lista de variables de cada valor de la potencia de
dimensionamiento trifásica total correspondiente a cada valor de resistencia de la carga ingresado por el usuario.
134             execstr('salidasGItran(' + valvaraux + ',1:4) = PLtran' + valvaraux + ', PTtran' + valvaraux + ', QTtran' +
valvaraux + ', STtran' + valvaraux); // Se guardan en el vector "salidasGItran[valvaraux]" los valores de las
variables definidas en las líneas de la 130 a la 133.

```

```

135     return; // Se regresa a la función anterior (GItransitorio).
136 else // En cambio, si el GI trifásico está desbalanceado, entonces:
137     PLtran = PLtran1 + PLtran2 + PLtran3; // Se calcula la potencia activa en la carga trifásica como la suma de
la potencia activa de la carga de cada fase del GI.
138     if conf_carga == 'estrella' then // Si la carga trifásica del GI está configurada en estrella, entonces:
139         if conf_carga_estrella == 'tetrafilar' then // Si la carga trifásica en estrella del GI es tetrafilar,
entonces:
140             PnL = RnL*(sqrt(PLtran1/RLf1) + sqrt(PLtran2/RLf2) + sqrt(PLtran3/RLf3))^2; // Se calcula el valor de la
potencia en el neutro de la carga trifásica.
141             PTtran = PTtran + PnL; // Se define nuevamente la potencia activa total sumándole el valor de la potencia
en el neutro de la carga trifásica.
142         end; // Fin de la rutina de bifurcación que comienza en la línea 139.
143     end; // Fin de la rutina de bifurcación que comienza en la línea 138.
144     PTtran = PTtran1 + PTtran2 + PTtran3; // Se calcula la potencia activa trifásica total como la suma de la
potencia total de cada fase del GI.
145     if conf_estat == 'estrella' then // Si el estator del GI está configurado en estrella, entonces:
146         if conf_estat_estrella == 'tetrafilar' then // Si el estator en estrella del GI es tetrafilar, entonces:
147             PnE = RnE*(PLtran1/VTmax + PLtran2/VTmax + PLtran1/VTmax^2)^2; // Se calcula el valor de la potencia en el
neutro del GI.
148             PTtran = PTtran + PnE; // Se define nuevamente la potencia activa total sumándole el valor de la potencia
en el neutro del GI.
149         end; // Fin de la rutina de bifurcación que comienza en la línea 146.
150     end; // Fin de la rutina de bifurcación que comienza en la línea 145.
151     QTtran = QTtran1 + QTtran2 + QTtran3; // Se calcula la potencia reactiva trifásica total como la suma de la
potencia total de cada fase del GI.
152     STtran = sqrt(PTtran**2 + QTtran**2); // Se calcula la potencia de dimensionamiento trifásica total como la
raiz cuadrada de la suma del cuadrado de las potencias trifásicas activa y reactiva totales.
153     execstr('PLtran'+ valvaraux + '= PLtran'); // Se guarda en la variable "PLtran[valvaraux]" el valor presente
en la variable "PLtran" de modo tal que se forma una lista de variables de cada valor de la potencia activa trifásica
en la carga correspondiente a cada valor de resistencia de la carga ingresado por el usuario.
154     execstr('PTtran'+ valvaraux + '= PTtran'); // Se guarda en la variable "PTtran[valvaraux]" el valor presente
en la variable "PTtran" de modo tal que se forma una lista de variables de cada valor de la potencia activa trifásica
total correspondiente a cada valor de resistencia de la carga ingresado por el usuario.
155     execstr('QTtran'+ valvaraux + '= QTtran'); // Se guarda en la variable "QTtran[valvaraux]" el valor presente
en la variable "QTtran" de modo tal que se forma una lista de variables de cada valor de la potencia reactiva
trifásica total correspondiente a cada valor de resistencia de la carga ingresado por el usuario.
156     execstr('STtran'+ valvaraux + '= STtran'); // Se guarda en la variable "STtran[oaux]" el valor presente en la
variable "STtran" de modo tal que se forma una lista de variables con la potencia de dimensionamiento total en cada
fase del GI.
157     execstr('salidasGItran(' + valvaraux + ',1:4) = PLtran' + valvaraux + ', PTtran' + valvaraux + ', QTtran' +
valvaraux + ', STtran' + valvaraux); // Se guardan en el vector "salidasGItran[valvaraux]" los valores de las
variables definidas en las líneas de la 153 a la 156.
158     return; // Se regresa a la función anterior (GItransitorio).
159 end; // Fin de la rutina de bifurcación que comienza en la línea 125.
160 else // Si el GI es monofásico, entonces:
161     PLtran = PLtran1 + PLtran2; // La potencia activa monofásica de la carga se halla sumando las potencias activas
en la carga debidas a la secuencia positiva y a la secuencia negativa de la tensión de transformación.
162     PTtran = PTtran1 + PTtran2; // La potencia activa monofásica total se halla sumando las potencias activas
totales debidas a la secuencia positiva y a la secuencia negativa de la tensión de transformación.
163     QTtran = QTtran1 + QTtran2; // La potencia reactiva monofásica total se halla sumando las potencias reactivas
totales debidas a la secuencia positiva y a la secuencia negativa de la tensión de transformación.
164     STtran = sqrt(PTtran**2 + QTtran**2); // Se calcula la potencia de dimensionamiento monofásica total como la
raiz cuadrada de la suma del cuadrado de las potencias monofásicas activa y reactiva totales.
165     execstr('PLtran'+ valvaraux + '= PLtran'); // Se guarda en la variable "PLtran[valvaraux]" el valor presente en
la variable "PLtran" de modo tal que se forma una lista de variables de cada valor de la potencia activa trifásica en
la carga correspondiente a cada valor de resistencia de la carga ingresado por el usuario.
166     execstr('PTtran'+ valvaraux + '= PTtran'); // Se guarda en la variable "PTtran[valvaraux]" el valor presente en
la variable "PTtran" de modo tal que se forma una lista de variables de cada valor de la potencia activa trifásica

```

```

167     execstr('QTtran'+ valvaraux + '= QTtran'); // Se guarda en la variable "QTtran[valvaraux]" el valor presente en
la variable "QTtran" de modo tal que se forma una lista de variables de cada valor de la potencia reactiva trifásica
total correspondiente a cada valor de resistencia de la carga ingresado por el usuario.
168     execstr('STtran'+ valvaraux + '= STtran'); // Se guarda en la variable "STtran[caux]" el valor presente en la
variable "STtran" de modo tal que se forma una lista de variables con la potencia de dimensionamiento total en cada
fase del GI.
169     execstr('salidasGITran(' + valvaraux + ',1:4) = PTtran' + valvaraux + ', FTtran' + valvaraux + ', QTtran' +
valvaraux + ', STtran' + valvaraux); // Se guardan en el vector "salidasGITran[valvaraux]" los valores de las
variables definidas en las líneas de la 165 a la 168.
170     return; // Se regresa a la función anterior (GITransitorio).
171 end; // Fin de la rutina de bifurcación que comienza en la línea 124.
172 endfunction

```

Fuente: El autor

Figura B.10 Algoritmo para el cálculo de las variables eléctricas del GI en régimen transitorio

```

1 function GITransitorio // Función para determinar las variables eléctricas del GI cuando este está en régimen
transitorio:
2     global salidasGITran; // Se define el tipo de la variable "salidasGITran" como "global" para poder utilizarla en
otras funciones.
3     if fases == 1 then // Si el GI es monofásico, se calculan los valores de la carga y se cargan las variables
correspondientes:
4         varcarga = abs((RLf-RLi)/(nRL-1)); // Se calcula el valor de la diferencia entre dos valores consecutivos de la
resistencia de la carga del GI.
5         if RLf > RLi then // Si el valor final de la resistencia de la carga del GI es mayor que el valor inicial de esta,
entonces:
6             signo = '+'; // Se le asigna el valor string "+" a la variable "signo".
7             elseif RLf < RLi then // En cambio, si el valor final de la resistencia de la carga del GI es menor que el valor
inicial de esta, entonces:
8                 signo = '-'; // Se le asigna el valor string "-" a la variable "signo".
9             else // En cambio, si el valor final de la resistencia de la carga del GI es igual a el valor inicial de esta,
entonces:
10                signo = '+'; // Se le asigna el valor string "+" a la variable "signo".
11            end; // Fin de la rutina de bifurcación que comienza en la línea 5.
12            for valvar = 1:nRL, // Rutina de iteración para calcular cada uno de los valores de la resistencia de la carga
del GI y los respectivos valores de tensión, corriente y potencia (de la carga y de transformación) en régimen
transitorio:
13                valvaraux = sci2exp(valvar); // Se define la variable "valvaraux" como la versión tipo string de la variable
"valvar", para utilizarla en la función "GITrancargal" que es llamada en las línea 21.
14                if valvar == 1 then // Si la variable "valvar" tiene un valor numérico de "1" (la iteración que comienza en la
línea 12 está en su primer ciclo), entonces:
15                    mcararm = RLi; // Se define el valor de la variable "mcararm1" como el valor inicial de la resistencia de la
carga del GI.
16                elseif valvar == nRL then // En cambio, si la variable "valvar" tiene un valor numérico igual al de la variable
"nRL" (la iteración que comienza en la línea 12 está en su último ciclo), entonces:

```

```

17 mcararm1 = RLf; // Se define el valor de la variable "mcararm1" como el valor final de la resistencia de la
carga del GI.
18 else // En cambio, si la variable "valvar" tiene un valor numérico diferente a "1" y al de la variable "nRL"
(la iteración que comienza en la línea 20 está en un ciclo entre el primero y el último), entonces:
19   execstr('mcararm = mcararm' + signo + 'varcarga'); // Se calcula el valor de la variable "mcararm1" como el
valor de esta variable en el ciclo anterior más (o menos dependiendo del valor de la variable "signo" definida entre
las líneas 5 y 11) más el valor de la variable "varcarga" definida en la línea 4.
20 end; // Fin de la rutina de bifurcación que comienza en la línea 14.
21 Gitrancargal; // Se ejecuta la función "Gitrancargal" que calcula los valores de tensión, corriente y potencia
(de la carga del GI y de transformación) en régimen transitorio para una carga lineal y resistiva.
22 end; // Fin de la rutina de iteración que comienza en la línea 12.
23 execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4)
+ '_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGITran.bin',salidasGITran)'); // Se guarda la matriz
"salidasGITran" (que contiene los valores de tensión, corriente y potencia calculados con la función "Gitrancargal")
en el archivo "salidasGITran.bin" ubicado en la carpeta cuyo nombre es la hora y fecha en que se realiza la
simulación.
24 else // En cambio, si el GI es trifásico:
25   if balance == 1 then // Si el GI trifásico está balanceado, se calculan los valores de la carga y se cargan las
variables correspondientes:
26     varcarga = abs((RLf-RLi)/(nRL-1)); // Se calcula el valor de la diferencia entre dos valores consecutivos de la
resistencia de la carga del GI.
27     if RLf > RLi then // Si el valor final de la resistencia de la carga del GI es mayor que el valor inicial de
esta, entonces:
28       signo = '+'; // Se le asigna el valor string "+" a la variable "signo".
29     elseif RLf < RLi then // En cambio, si el valor final de la resistencia de la carga del GI es menor que el
valor inicial de esta, entonces:
30       signo = '-'; // Se le asigna el valor string "-" a la variable "signo".
31     else // En cambio, si el valor final de la resistencia de la carga del GI es igual a el valor inicial de esta,
entonces:
32       signo = '+'; // Se le asigna el valor string "+" a la variable "signo".
33     end; // Fin de la rutina de bifurcación que comienza en la línea 27.
34     for valvar = 1:nRL, // Rutina de iteración para calcular cada uno de los valores de la resistencia de la carga
del GI y los respectivos valores de tensión, corriente y potencia (de la carga y de transformación) en régimen
transitorio:
35       valvaraux = sci2exp(valvar); // Se define la variable "valvaraux" como la versión tipo string de la variable
"valvar", para utilizarla en la función "Gitrancargal" que es llamada en la línea 41.
36       if valvar == 1 then // Si la variable "valvar" tiene un valor numérico de "1" (la iteración que comienza en
la línea 34 está en su primer ciclo), entonces:
37         mcararm = RLi; // Se define el valor de la variable "mcararm1" como el valor inicial de la resistencia de
la carga del GI.
38       elseif valvar == nRL then // En cambio, si la variable "valvar" tiene un valor numérico igual al de la
variable "nRL" (la iteración que comienza en la línea 34 está en su último ciclo), entonces:
39         mcararm1 = RLf; // Se define el valor de la variable "mcararm1" como el valor final de la resistencia de la
carga del GI.
40       else // En cambio, si la variable "valvar" tiene un valor numérico diferente a "1" y al de la variable "nRL"
(la iteración que comienza en la línea 34 está en un ciclo entre el primero y el último, excluyéndolos), entonces:
41         execstr('mcararm = mcararm' + signo + 'varcarga'); // Se calcula el valor de la variable "mcararm1" como el
valor de esta variable en el ciclo anterior más (o menos dependiendo del valor de la variable "signo" definida entre
las líneas 27 y 33) más el valor de la variable "varcarga" definida en la línea 26.
42         end; // Fin de la rutina de bifurcación que comienza en la línea 36.
43         Gitrancargal; // Se ejecuta la función "Gitrancargal" que calcula los valores de tensión, corriente y
potencia (de la carga del GI y de transformación) en régimen transitorio para una carga lineal y resistiva
44         end; // Fin de la rutina de iteración que comienza en la línea 34.
45         execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4)
+ '_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGITran.bin',salidasGITran)'); // Se guarda la matriz
"salidasGITran" (que contiene los valores de tensión, corriente y potencia calculados con la función "Gitrancargal")
en el archivo "salidasGITran.bin" ubicado en la carpeta cuyo nombre es la hora y fecha en que se realiza la
simulación.

```

```

46 else // En cambio, si el GI trifásico está desbalanceado, se calculan los valores de la carga y se cargan las
    variables correspondientes:
47     varcarga1 = abs((RLf1-RLi1)/(nRL-1)); // Se calcula el valor de la diferencia entre dos valores consecutivos de
    la resistencia de la carga de la fase 1 del GI.
48     varcarga2 = abs((RLf2-RLi2)/(nRL-1)); // Se calcula el valor de la diferencia entre dos valores consecutivos de
    la resistencia de la carga de la fase 2 del GI.
49     varcarga3 = abs((RLf3-RLi3)/(nRL-1)); // Se calcula el valor de la diferencia entre dos valores consecutivos de
    la resistencia de la carga de la fase 3 del GI.
50     if RLf1 > RLi1 then // Si el valor final de la resistencia de la carga de la fase 1 del GI es mayor que el valor
    inicial de esta, entonces:
51         signo1 = '+'; // Se le asigna el valor string "+" a la variable "signo1".
52     elseif RLf1 < RLi1 then // En cambio, si el valor final de la resistencia de la carga de la fase 1 del GI es
    menor que el valor inicial de esta, entonces:
53         signo1 = '-'; // Se le asigna el valor string "-" a la variable "signo1".
54     else // En cambio, si el valor final de la resistencia de la carga de la fase 1 del GI es igual a el valor
    inicial de esta, entonces:
55         signo1 = '+'; // Se le asigna el valor string "+" a la variable "signo1".
56     end; // Fin de la rutina de bifurcación que comienza en la línea 50.
57     if RLf2 > RLi2 then // Si el valor final de la resistencia de la carga de la fase 2 del GI es mayor que el valor
    inicial de esta, entonces:
58         signo2 = '+'; // Se le asigna el valor string "+" a la variable "signo2".
59     elseif RLf2 < RLi2 then // En cambio, si el valor final de la resistencia de la carga de la fase 2 del GI es
    menor que el valor inicial de esta, entonces:
60         signo2 = '-'; // Se le asigna el valor string "-" a la variable "signo2"
61     else // En cambio, si el valor final de la resistencia de la carga de la fase 2 del GI es igual a el valor
    inicial de esta, entonces:
62         signo2 = '+'; // Se le asigna el valor string "+" a la variable "signo2"
63     end; // Fin de la rutina de bifurcación que comienza en la línea 57.
64     if RLf3 > RLi3 then // Si el valor final de la resistencia de la carga de la fase 3 del GI es mayor que el valor
    inicial de esta, entonces:
65         signo3 = '+'; // Se le asigna el valor string "+" a la variable "signo3".
66     elseif RLf3 < RLi3 then // En cambio, si el valor final de la resistencia de la carga de la fase 3 del GI es
    menor que el valor inicial de esta, entonces:
67         signo3 = '-'; // Se le asigna el valor string "-" a la variable "signo3".
68     else // En cambio, si el valor final de la resistencia de la carga de la fase 3 del GI es igual a el valor
    inicial de esta, entonces:
69         signo3 = '+'; // Se le asigna el valor string "+" a la variable "signo3".
70     end; // Fin de la rutina de bifurcación que comienza en la línea 64.
71     for valvar = 1:nRL, // Rutina de iteración para calcular cada uno de los valores de la resistencia de la carga
    del GI y los respectivos valores de tensión, corriente y potencia (de la carga y de transformación) en régimen
    transitorio:
72         valvaraux = sci2exp(valvar); // Se define la variable "valvaraux" como la versión tipo string de la variable
    "valvar", para utilizarla en la función "GIpermcargal" que es llamada en la línea 82.
73         if valvar == 1 then // Si la variable "valvar" tiene un valor numérico de "1" (la iteración que comienza en la
    línea 71 está en su primer ciclo), entonces:
74             mcararm1 = RLi1; // Se define el valor de la variable "mcararm1" como el valor inicial de la resistencia de
    la carga de la fase 1 del GI.
75             mcararm2 = RLi2; // Se define el valor de la variable "mcararm1" como el valor inicial de la resistencia de
    la carga de la fase 2 del GI.
76             mcararm3 = RLi3; // Se define el valor de la variable "mcararm1" como el valor inicial de la resistencia de
    la carga de la fase 3 del GI.
77         elseif valvar == nRL then // En cambio, si la variable "valvar" tiene un valor numérico igual al de la variable
    "nRL" (la iteración que comienza en la línea 79 está en su último ciclo), entonces:
78             mcararm1 = RLf1; // Se define el valor de la variable "mcararm1" como el valor final de la resistencia de la
    carga de la fase 1 del GI.
79             mcararm2 = RLf2; // Se define el valor de la variable "mcararm1" como el valor final de la resistencia de la
    carga de la fase 2 del GI.
80             mcararm3 = RLf3; // Se define el valor de la variable "mcararm1" como el valor final de la resistencia de la

```

```

carga de la fase 3 del GI.
81     else // En cambio, si la variable "valvar" tiene un valor numérico diferente a "1" y al de la variable "nRL"
      (la iteración que comienza en la línea 71 está en un ciclo entre el primero y el último, excluyéndolos), entonces:
82     execstr('mcararm1 = mcararm1 + signo1 + 'varcarga1'); // Se calcula el valor de la variable "mcararm1" como
      el valor de esta variable en el ciclo anterior más (o menos dependiendo del valor de la variable "signo1" definida
      entre las líneas 50 y 56) más el valor de la variable "varcarga1" definida en la línea 47.
83     execstr('mcararm2 = mcararm2 + signo2 + 'varcarga2'); // Se calcula el valor de la variable "mcararm2" como
      el valor de esta variable en el ciclo anterior más (o menos dependiendo del valor de la variable "signo2" definida
      entre las líneas 57 y 63) más el valor de la variable "varcarga2" definida en la línea 48.
84     execstr('mcararm3 = mcararm3 + signo3 + 'varcarga3'); // Se calcula el valor de la variable "mcararm3" como
      el valor de esta variable en el ciclo anterior más (o menos dependiendo del valor de la variable "signo3" definida
      entre las líneas 64 y 70) más el valor de la variable "varcarga3" definida en la línea 49.
85     end; // Fin de la rutina de bifurcación que comienza en la línea 73.
86     GITrancargal; // Se ejecuta la función "GITrancargal" que calcula los valores de tensión, corriente y potencia
      (de la carga del GI y de transformación) en régimen transitorio para una carga lineal y resistiva.
87     end; // Fin de la rutina de iteración que comienza en la línea 71.
88     execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4)
      + '_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGITran.bin',salidasGITran)'); // Se guarda la matriz
      "salidasGITran" (que contiene los valores de tensión, corriente y potencia calculados con la función "GITrancargal")
      en el archivo "salidasGITran.bin" ubicado en la carpeta cuyo nombre es la hora y fecha en que se realiza la
      simulación.
89     end; // Fin de la rutina de bifurcación que comienza en la línea 25.
90 end; // Fin de la rutina de bifurcación que comienza en la línea 3.
91 endfunction

```

Fuente: El autor

Figura B.11 Algoritmo para el cálculo de las pérdidas mecánicas del MCI

```

1 function PerdidasMecanicas // Función para calcular las pérdidas mecánicas del MCI.
2 Pf = Vd*RPM*(0.97+0.15*(RPM/1000)+0.05*(RPM/1000)*2)/(5*Tiempos); // Se calculan las pérdidas mecánicas del MCI y se
      guarda este valor en la variable "Pf".
3 execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) + '_'
      + fechayhora(5) + '_' + fechayhora(6) + '\Perdidas_mecanicas.bin', Pf)'); // Se guarda la variable "Pf" en el
      archivo "Perdidas_mecanicas.bin" ubicándolo en la carpeta de la fecha y hora de la simulación.
4 return; // Se retorna a la función anterior (Simular).
5 endfunction

```

Fuente: El autor

Figura B.12 Algoritmo para el cálculo de las pérdidas térmicas del MCI

```

1 function PerdidasTermicas //Función para calcular las pérdidas térmicas del MCI.
2   nt = 1/sfc/Qf; // Se calcula las eficiencia térmica del MCI y se guarda este valor en la variable "nt".
3   execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
4     '_' + fechayhora(5) + '_' + fechayhora(6) + '\\salidasGITotales.bin', 'salidasGITotales'); // Se carga el
vector "salidasGITotales" desde el archivo "salidasGITotales.bin" ubicado en la carpeta de la fecha y hora de la
simulación.
5   execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
6     '_' + fechayhora(5) + '_' + fechayhora(6) + '\\Perdidas_mecanicas.bin', 'Pf'); // Se carga la variable "Pf"
desde el archivo "Perdidas_mecanicas.bin" ubicado en la carpeta de la fecha y hora de la simulación.
7   for varperter = 1:nRL, // Rutina de iteración para calcular los valores de las pérdidas térmicas correspondientes a
8     cada valor de la carga ingresado por el usuario:
9     varperteraux = sci2exp(varperter); // Se define la variable "varperteraux" como la versión string de la variable
numérica "varperter".
10    execstr('STtotal' + varperteraux + '= salidasGITotales(varperter,4)'); // Se define cada variable
"STtotal[varperteraux]" como el valor guardado en el vector "salidasGITotales" en la posición (varperter,4).
11    execstr('Pht(varperter) = (STtotal' + varperteraux + '+ Pf)*(1/nt - 1)'); // Se calculan las pérdidas térmicas del
MCI correspondientes al valor "varperter" de la carga ingresado por el usuario y se guarda este valor en el vector Pht
en la posición (varperter).
12   end; // Fin de la rutina de iteración que comienza en la línea 5.
13   execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
14     '_' + fechayhora(5) + '_' + fechayhora(6) + '\\Perdidas_termicas.bin', Pht); // Se guarda el vector "Pht" en el
archivo "Perdidas_termicas.bin" ubicándolo en la carpeta de la fecha y hora de la simulación.
15   return; // Se retorna a la función anterior (Simular).
16 endfunction

```

Fuente: El autor

Figura B.13 Algoritmo para el cálculo del flujo térmico de combustión del MCI

```

1 function PotMecTotal //Función para determinar el flujo térmico de combustión.
2   PmT = ST + Pm + Pt; // Se calculan el flujo térmico de combustión del MCI y se guarda este valor en la variable "PmT".
3   execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) + '_' +
4     fechayhora(5) + '_' + fechayhora(6) + '\\Flujo_term.bin', PmT); // Se guarda la variable "PmT" en el archivo
"Flujo_term.bin" ubicándolo en la carpeta de la fecha y hora de la simulación.
5   return; // Se retorna a la función anterior (Simular).
6 endfunction

```

Fuente: El autor

Figura B.14 Algoritmo para el cálculo de la potencia equivalente del MCI

```

1 function PotTotal // Función para calcular la potencia equivalente del MCI.
2   execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
3     '_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGITotales.bin', 'salidasGITotales'); // Se carga el
4     vector "salidasGITotales" desde el archivo "salidasGITotales.bin" ubicado en la carpeta de la fecha y hora de la
5     simulación.
6   execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
7     '_' + fechayhora(5) + '_' + fechayhora(6) + '\Perdidas_mecanicas.bin', 'Pf'); // Se carga la variable "Pf"
8     desde el archivo "Perdidas_mecanicas.bin" ubicado en la carpeta de la fecha y hora de la simulación.
9   execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
10    '_' + fechayhora(5) + '_' + fechayhora(6) + '\Perdidas_termicas.bin', 'Pht'); // Se carga el vector "Pht"
11    desde el archivo "Perdidas_termicas.bin" ubicado en la carpeta de la fecha y hora de la simulación.
12   for varPtotal = 1:nRL, // Rutina de iteración para calcular los valores de la potencia equivalente del MCI
13     correspondientes a cada valor de la resistencia de carga ingresados por el usuario:
14     varPtotalaux = sci2exp(varPtotal); // Se define la variable "varPtotalaux" como la versión string de la variable
15     numérica "varPtotal".
16     execstr('STtotal' + varPtotalaux + '= salidasGITotales(varPtotal,4)'); // Se define cada variable
17     "STtotal[varPtotalaux]" como el valor guardado en el vector "salidasGITotales" en la posición (varPtotal,4).
18     execstr('PTtotal(varPtotal) = STtotal' + varPtotalaux + ' + Pht(varPtotal) + Pf'); // Se calculan los valores de la
19     potencia equivalente del MCI correspondientes al valor "varPtotal" de la carga ingresado por el usuario y se guarda este
20     valor en el vector "PTtotal" en la posición (varPtotal).
21   end; // Fin de la rutina de iteración que comienza en la línea 5.
22   execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
23     '_' + fechayhora(5) + '_' + fechayhora(6) + '\Potencia_Total.bin', PTtotal); // Se guarda el vector "PTtotal" en
24     el archivo "Potencia_Total.bin" ubicándolo en la carpeta de la fecha y hora de la simulación.
25   return; // Se retorna a la función anterior (Simular).
26 endfunction

```

Fuente: El autor

Figura B.15 Algoritmo para el cálculo del torque mecánico del grupo generador

```

1 function Torquemecanico // función para realizar el cálculo del torque en un MCI.
2   execstr('load(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
3     '_' + fechayhora(5) + '_' + fechayhora(6) + '\salidasGITotales.bin', 'salidasGITotales'); // Desde el
4     directorio creado para la presente simulación, se cargan los valores de las salidas totales del GI, desde el vector
5     "salidasGITotales" guardada en el archivo "salidasGITotales.bin".
6   for varTorque = 1:nRL, // Rutina de iteración para calcular cada uno de los valores del torque del MCI
7     correspondientes a cada valor de la carga eléctrica del GI:
8     varTorqueaux = sci2exp(varTorque); // Se define la variable "varTorqueaux" como la versión string de la variable
9     varTorque.
10    execstr('STtotal' + varTorqueaux + '= salidasGITotales(varTorque,4)'); // Se crean las variables "STtotal1" hasta
11    "STtotal[nRL]" es decir los valores de la potencia de dimensionamiento total del GI.
12    execstr('Torque(varTorque) = STtotal' + varTorqueaux + '*30/(&pi;*RPM)'); // A partir de cada valor de la potencia de
13    dimensionamiento total del GI se calcula cada valor del torque mecánico del MCI, y se guardan estos valores en el vector
14    "Torque".
15   end; // Fin de la rutina de iteración que comienza en la línea 3.
16   execstr('save(directorio + fechayhora(1) + '_' + fechayhora(2) + '_' + fechayhora(3) + '_' + fechayhora(4) +
17     '_' + fechayhora(5) + '_' + fechayhora(6) + '\Torquemecanico.bin', Torque); // Se guarda el vector "Torque" en
18     el archivo "Torquemecanico.bin" y en el directorio creado para la presente simulación.
19   return; // Se termina la ejecución de esta función.
20 endfunction

```

Fuente: El autor

ANEXO C. MANUAL DE LA PLATAFORMA DE SIMULACIÓN

El presente manual se propone para su utilización en las prácticas de laboratorio de la asignatura Máquinas Térmicas Alternativas de la Escuela de Ingeniería Mecánica.

C.1. CREACIÓN DE USUARIOS

Pasos a seguir:

C.1.1. Abrir SCILAB.

Para esto se da doble clic en el ícono del escritorio que se muestra en la figura C.1.

Figura C.1 Acceso directo a Scilab



Fuente: El autor.

C.1.2. Cambiar el directorio actual

Este paso es necesario para facilitar el paso C.1.3. Para esto se da clic en el icono del menú superior de la consola de Scilab, de modo que aparece una subventana de selección de directorio (en la cual se debe navegar hasta encontrar el directorio donde están guardados los archivos y carpetas de la plataforma de simulación) tal como lo muestra la figura C.2.

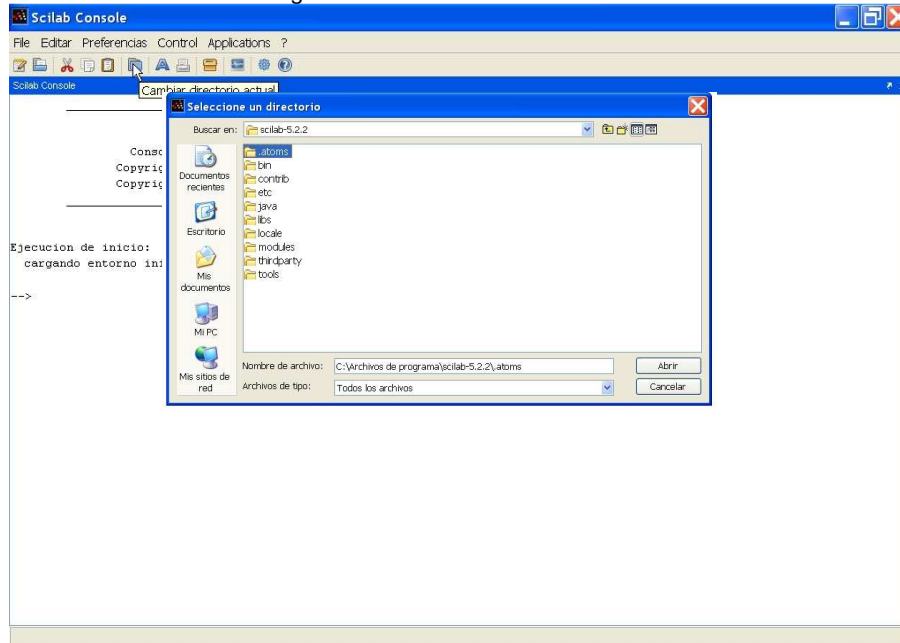
C.1.3. Ejecutar el programa de simulación

Digitar en la consola de SCILAB la siguiente instrucción: “exec Simulador_de_un_Grupo_Electrogeno.sce” (sin las comillas), tal como aparece en la figura C.3.

C.1.4. Abrir la ventana de creación de usuario

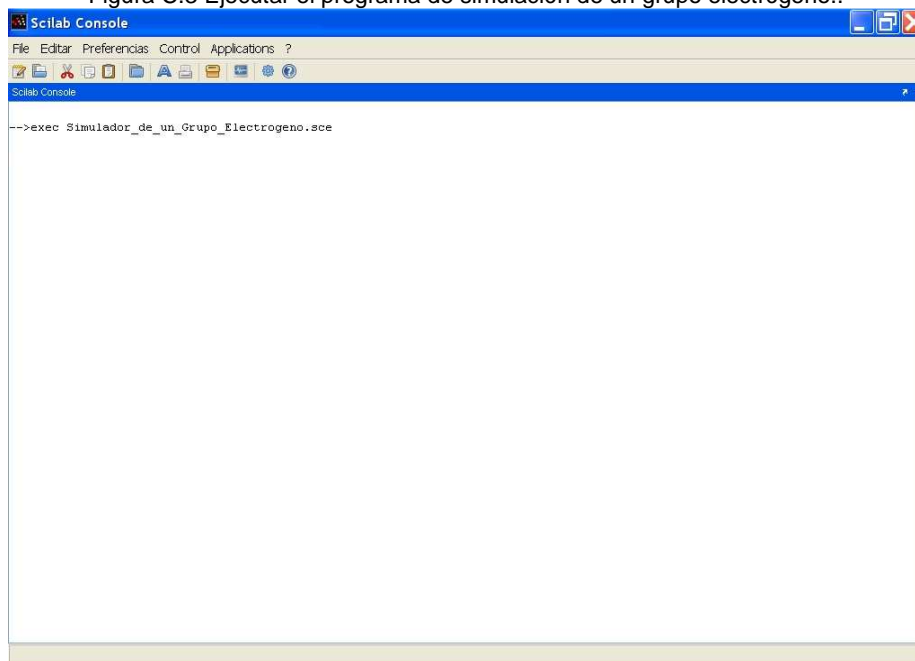
Después de hacer el paso C.1.3 y oprimir la tecla “Enter” aparece el menú principal, el cual se puede observar en la figura C.4. Escoger la opción “Crear Usuario”

Figura C.2 Cambiar directorio actual



Fuente: El autor

Figura C.3 Ejecutar el programa de simulación de un grupo electrógeno..



Fuente: El autor

Figura C.4 Menú principal de la plataforma de simulación, con la opción “Crear usuario” elegida

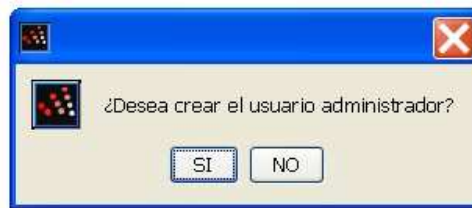


Fuente: El autor

C.1.5. Ingreso de usuario y contraseña

Si aún no se ha creado el usuario administrador, aparece la siguiente ventana:

Figura C.5 Ventana de decisión para crear o no el usuario administrador

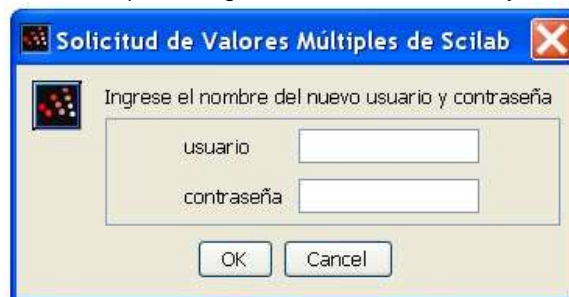


Fuente: El autor

Se recomienda crear el usuario administrador antes de que el programa sea utilizado por algún estudiante.

Independientemente de que se oprima el botón “SI” o el botón “NO” de la figura anterior, o si se escoge la opción “Crear Usuario” en la figura C.3 habiéndose creado ya el usuario administrador, aparecerá la siguiente ventana:

Figura C.6 Ventana para el ingreso de un nuevo usuario y su contraseña



Fuente: El autor

C.1.6. Entonces se ingresa el nombre de usuario y la contraseña y se oprime “OK”.

La creación de un usuario es indispensable para poder acceder a las otras opciones del menú principal.**C.2 PRÁCTICA TEÓRICA 1: MEDIDAS DE SEGURIDAD PARA TRABAJAR CON UN GRUPO ELECTRÓGENO**

Pasos a seguir:

C.2.1. Abrir SCILAB.

Para esto se da doble clic en el ícono del escritorio que se muestra en la figura C.1.

C.2.2. Cambiar el directorio actual

Este paso es necesario para facilitar el paso C.2.3. Para esto se da clic en el icono del menú superior de la consola de Scilab, de modo que aparece una ventana de selección de directorio (en la cual se debe navegar hasta encontrar el directorio donde están guardados los archivos y carpetas de la plataforma de simulación) tal como lo muestra la figura C.2.

C.2.3. Ejecutar el programa de simulación

Digitar en la consola de SCILAB la siguiente instrucción:

“exec Simulador_de_un_Grupo_Electrogeno.sce”, sin las comillas, tal como aparece en la figura C.3.

C.2.4. Abrir el menú de prácticas

Después de hacer el paso C.2.3 y oprimir la tecla “Enter” aparece el menú principal, se debe escoger la opción “Realizar prácticas”, como se ve en la figura C.7.

Figura C.7 Menú principal de la plataforma de simulación, con la opción “Realizar prácticas” elegida

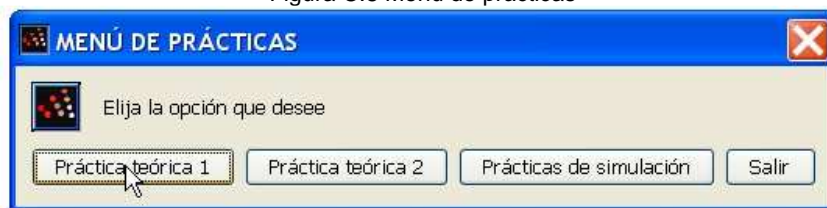


Fuente: El autor

C.2.5. Abrir el menú de la práctica teórica 1

Luego de dar clic al botón mencionado en el paso C.2.4, aparece el menú de prácticas, entonces, se debe escoger “Práctica teórica 1”, como puede verse en la figura C.8.

Figura C.8 Menú de prácticas



Fuente: El autor

C.2.6. Abrir el documento con el contenido de la práctica teórica 1

Al oprimir el botón mencionado en el paso C.2.5, aparece el menú de la práctica 1, el cual se muestra en la figura C.9.

Figura C.9 Abrir el documento de la práctica teórica 1



Fuente: El autor

Si se oprime el botón “Abrir documento” (el cual está señalado en la figura C.9), se abre un tutorial el cual aparece a continuación:

1. Información general sobre medidas de seguridad

¿Por qué es importante la seguridad al trabajar con un grupo electrógeno?

- Para conservar la vida y la buena salud de los usuarios de la máquina.
- Para evitar daños en el sistema.

¿Por qué se producen los accidentes y lesiones?

- Por ignorancia, descuido o torpeza ante un riesgo.
- Por el afán de ejecutar una tarea en un tiempo determinado.

¿Cuál es la mejor manera de concientizarse de la necesidad de tomar precauciones ante un riesgo?

Aunque las estadísticas nos informan de la probabilidad de sufrir daños personales o producirlos en la máquina, los testimonios de quienes han incurrido en errores que causaron daños graves son más convincentes que los datos, pues estos son impersonales y parecen lejanos de nuestra realidad, en cambio un testimonio se percibe como algo real y palpable.

¿Cuáles son las fuentes más comunes de riesgos en un grupo generador?

El combustible, la velocidad angular del sistema, el torque de la máquina, el calor producido por la combustión o por efecto Joule, la tensión eléctrica y la corriente eléctrica del generador y del sistema de arranque eléctrico del MCI, los sistemas de presiones y vacíos en el MCI y los gases contaminantes que salen del MCI; son las fuentes principales de riesgo en un grupo electrógeno.

2. Precauciones al manipular gasolina (y otros combustibles líquidos).

- Este combustible tiene un poder explosivo mayor que el TNT lo cual tiene sin cuidado a muchos usuarios que la utilizan con actitud temeraria, por ser algo cotidiano para ellos.
- Para evitar lesiones graves o incluso la muerte al manipular este combustible, se deben tomar las siguientes precauciones:
- Debe almacenarse en un lugar seguro, lejos de otros materiales inflamables. Unas recomendaciones respecto al lugar de almacenaje de la gasolina aparecen a continuación:
 - ✓ Utilice un recipiente metálico para guardar la gasolina.
 - ✓ Ubíquela en un sitio exterior (el laboratorio, por ejemplo) y no en una oficina o salón de clases.
 - ✓ No la almacene durante periodos largos (para esto hay empresas especializadas en el almacenaje de combustibles, las cuales tienen disponible el combustible para los usuarios)
 - ✓ Asegúrese que ni niños ni animales domésticos puedan tener alcance al recipiente que contiene el combustible.
 - ✓ Mantenga la gasolina alejada de lugares y objetos con estática (pues se pueden producir chispas), que estén a alta temperatura, con llamas, que tengan contactos eléctricos o que produzcan fricción mecánica.
 - ✓ Tenga siempre a la mano, en el lugar en donde se almacena la gasolina. un extintor adecuado para sofocar incendios provocados por combustibles.
- Recomendaciones al utilizar gasolina:
 - ✓ Tenga presente siempre que la gasolina es peligrosa.
 - ✓ Utilícela sólo donde realmente se necesita (es decir, como combustible de motores y otros aparatos), no para encender fogatas o como líquido limpiador o disolvente.
 - ✓ Nunca fume o haga fuego cerca de la gasolina.
 - ✓ Nunca llene el tanque de un motor que está caliente o en marcha, sobre todo si el tanque está cerca del motor.

- ✓ Cuando vierta gasolina de un envase metálico al tanque de un motor, el pico del recipiente debe ser apoyado firmemente al borde metálico de la abertura del tanque, para evitar que se produzca una chispa por acción de electricidad estática. Se debe también evitar utilizar gamuza como filtro, pues puede producirse una descarga estática.

3. Incendios

- Para poder combatirlos se debe conocer su naturaleza. Esta está catalogado en los extintores, de modo que para las tres clases principales de incendios están las siguientes categorías:
 - ✓ Clase A: madera, tela, papel, basura. Se apagan enfriando el material inflamable con agua o ácido.
 - ✓ Clase B: gasolina, aceite, grasa, pintura. Se apagan cortando el suministro de oxígeno que alimenta al fuego utilizando CO₂. Se debe evitar tocar la boquilla del extintor cuando está en uso, pues esa se pone muy fría y también evitar la sofocación si se utiliza el extintor en un lugar cerrado, ya que el CO₂ disminuye la cantidad de oxígeno del ambiente
 - ✓ Clase C: equipo eléctrico. Deben ser sofocados por materiales que no conduzcan la electricidad, si el suministro de la energía eléctrica al aparato no se puede cortar o si hay riesgo de que el aparato conserve una carga eléctrica remanente. Estos materiales pueden ser CO₂, químicos en polvo o líquidos vaporizantes.

4. Monóxido de Carbono (CO)

- Es un tóxico muy peligroso por ser inodoro, incoloro e insípido; y por que puede ser mortal, ya que reduce rápidamente la cantidad de oxígeno en el cuerpo a través de su unión con la hemoglobina, la cual es 300 veces más fuerte que la unión del oxígeno con la molécula sanguínea mencionada.
- Este tóxico es el producto de la combustión incompleta de carbonáceos (sólidos, líquidos o gaseosos). Un ejemplo de estas sustancias que producen CO en su combustión es la gasolina y los otros combustibles utilizados en los MCI.
- La primera precaución que debe tomarse al utilizar MCI, respecto a su producción de CO, es ventilar adecuadamente el recinto donde está ubicado el MCI, y/o conducir sus gases de escape al exterior por medio de ductos, tubos, chimeneas o ventiladores.
- Síntomas de envenenamiento con CO:

- ✓ Sensación de tirantez en la piel de la frente, seguida de palpitaciones en las sienes.
- ✓ Debilidad.
- ✓ Náuseas.
- ✓ Fatiga.
- ✓ Dolor de cabeza.
- ✓ Vértigos
- ✓ Control muscular deficiente.
- ✓ Ritmo cardíaco y respiratorio acelerados.

Si siente alguno de estos síntomas, salga inmediatamente al aire libre.

- Precauciones para evitar el envenenamiento con CO:
 - ✓ No maneje o viaje en un auto que tiene todas las ventanas cerradas.
 - ✓ No ponga en marcha un MCI en un cuarto que no posea un buen sistema de ventilación.
 - ✓ Antes de poner en marcha un MCI, asegúrese de que el sistema de ventilación del recinto donde se encuentra la máquina funciona correctamente.
 - ✓ Mantenga todo MCI ajustado y afinado para reducir la producción de CO.
 - ✓ No permanezca más de unos cuantos minutos en el interior de un vehículo estacionado con el motor en marcha.

5. Precauciones al utilizar un MCI

- Reglas de seguridad para prevenir lesiones personales y daños en el motor:
 - ✓ Siga las reglas de almacenamiento de gasolina explicada en el apartado C.1.1
 - ✓ No llene el tanque de combustible con el motor en marcha.
 - ✓ Limpie todos los derrames de combustible tan pronto ocurran.
 - ✓ Utilice un embudo para llenar el tanque de combustible y emplee sólo gasolina de grado regular o fresca.
 - ✓ Nunca permita que el MCI opere a su máxima velocidad sin carga.
 - ✓ Póngase una indumentaria apropiada para manipular el motor (como una bata de taller).
 - ✓ No toque el eje principal ni los pistones cuando el motor está en marcha y, si lleva ropa suelta, manténgase alejado de las partes móviles del MCI.
 - ✓ No toque el silenciador hasta que el motor se halla enfriado.
 - ✓ Pare siempre el motor utilizando el sistema diseñado para este propósito.
 - ✓ Proporcione suficiente ventilación para el escape del motor.
- Revisión pre-operacional (lo que se debe hacer antes de arrancar un MCI):

- ✓ Compruebe que el motor tenga suficiente combustible para el tiempo que se va a utilizar.
- ✓ Asegúrese de que el motor esté bien lubricado.
- ✓ Compruebe el funcionamiento del tacómetro.
- ✓ Revise el sistema en busca de fugas.

6. Precauciones al utilizar un GI

- Reglas de seguridad para prevenir lesiones personales y daños en la máquina:
 - ✓ Póngase una indumentaria apropiada para manipular el generador (como una bata de taller).
 - ✓ No toque el eje principal cuando la máquina está en marcha y, si lleva ropa suelta, manténgase alejado de las partes móviles de esta.
 - ✓ No toque los cables de la máquina ni sus terminales eléctricos, por muy bien aislados que parezcan, cuando esta esté energizada.
 - ✓ Cuando vaya a realizar medidas eléctricas como tensiones y corrientes, asegúrese de que el rango de estos parámetros en el aparato de medida y en el generador o en el circuito conectado a este coincidan y que los amperímetros sean conectados en serie con el circuito y los voltímetros en paralelo al elemento al cual se le desea medir el parámetro correspondiente.
 - ✓ Cuando utilice un voltmetro, evite el contacto de sus manos con los bornes de la máquina.
 - ✓ Si es necesaria alguna modificación en un circuito eléctrico conectado a la máquina, seccione la parte la parte del circuito donde vaya a trabajar y no energice el circuito hasta que esté totalmente seguro de que no existe riesgo alguno de corto circuito.
 - ✓ Interrumpa el suministro de corriente a los circuitos conectados a la máquina únicamente utilizando los elementos diseñados para tal fin (interruptores).
 - ✓ Mantenga todo tipo de materiales inflamables lejos de los contactos eléctricos del generador y del metal expuesto en los circuitos conectados a ella.
 - ✓ No toque la máquina luego de que ésta ha estado en funcionamiento un tiempo prolongado o con una carga elevada, especialmente si el punto de contacto no está recubierto por un aislante térmico, podría sufrir quemaduras.
 - ✓ Nunca conecte la máquina con carga excesiva (según los parámetros de diseño de la máquina) esto puede ocasionar daños graves en la misma y un riesgo al usuario al exponerlo a temperatura altas o a arcos eléctricos.
- Revisión pre-operacional:
 - ✓ Compruebe que la máquina tenga magnetismo remanente.
 - ✓ Asegúrese de que el generador se encuentre bien lubricado y que sus escobillas (si las posee) se encuentren en buen estado.

- ✓ Compruebe las conexiones en los bornes del generador para asegurarse que no existe un cortocircuito.
- ✓ Compruebe la conexión del capacitor para que se pueda producir la corriente de magnetización.

C.2.7. Abrir el video de la práctica teórica 1

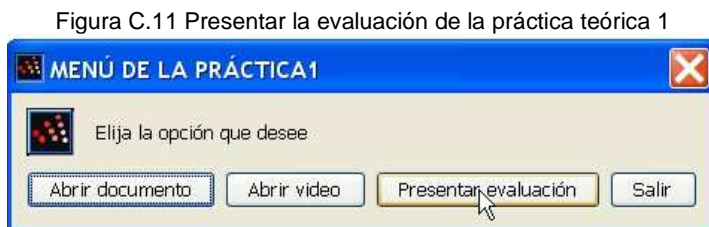
Si se oprime el botón “Abrir video” en el menú de la práctica 1 (como se observa en la figura C.10) se abre un video con contenido referente a la seguridad al operar un grupo electrógeno.



Fuente: El autor

C.2.8. Presentar la evaluación de la práctica teórica 1

Si en el menú de la práctica 1 oprimimos el botón “Presentar evaluación”, como se muestra en la figura C.11, se abre un documento que evalúa lo expuesto en los tutoriales.



Fuente: El autor

C.3. PRÁCTICA 2: OPERACIÓN DEL GRUPO ELECTRÓGENO

Pasos a seguir:

C.3.1. Abrir SCILAB.

Para esto se da doble clic en el ícono del escritorio que se muestra en la figura C.1.

C.3.2. Cambiar el directorio actual

Este paso es necesario para facilitar el paso C.2.3. Para esto se da clic en el icono del menú superior de la consola de Scilab, de modo que aparece una ventana de selección de directorio (en la cual se debe navegar hasta encontrar el directorio donde están guardados los archivos y carpetas de la plataforma de simulación) tal como lo muestra la figura C.2.

C.3.3. Ejecutar el programa de simulación

Digitar en la consola de SCILAB la siguiente instrucción:

“exec Simulador_de_un_Grupo_Electrogeno.sce”, sin las comillas, tal como aparece en la figura C.3.

C.3.4. Abrir el menú de prácticas

Después de hacer el paso C.3.3 y oprimir la tecla “Enter” aparece el menú principal, se debe escoger la opción “Realizar prácticas”, como se ve en la figura C.7.

C.3.5. Abrir el menú de la práctica teórica 2

Luego de dar clic al botón mencionado en el paso C.3.4, aparece el menú de prácticas, entonces, se debe escoger “Práctica teórica 2”, como puede verse en a figura C.12.

Figura C.12 Menú de prácticas



Fuente: El autor

C.3.6. Abrir el documento con el contenido de la práctica teórica 2

Al oprimir el botón mencionado en el paso C.3.5, aparece el menú de la práctica 2, el cual se muestra en la figura C.13.

Figura C.13 Menú de la práctica 2



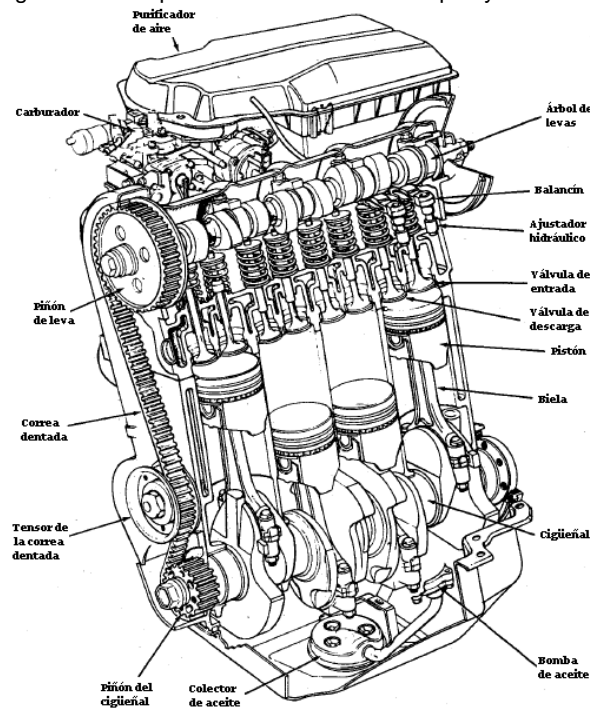
Fuente: El autor

Si se oprime el botón “Abrir documento” (el cual está señalado en la figura C.13), se abre un tutorial, el cual aparece a continuación:

1. componentes del MCI

En la figura C.14 puede observarse el esquema de los componentes principales de un MCI de 4 tiempos y 4 cilindros, el cual es un ejemplo típico de MCI, que además posee una complejidad suficiente para observar los sistemas principales que se encuentran en la mayoría de los MCI comerciales.

Figura C.14 Esquema de un MCI de 4 tiempos y 4 cilindros



Fuente: HEYWOOD John B, *Internal Combustion Engine Fundamentals*. Página 13

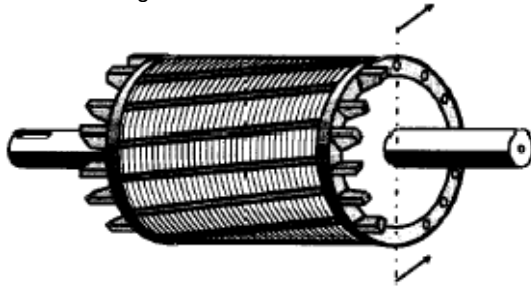
- Purificador de aire: su función es limpiar el aire del ambiente que va a entrar en los cilindros.

- Carburador: este aparato se encarga de realizar la mezcla aire –combustible en algunos MCI. En otros motores, el combustible es dosificado por un sistema de inyección.
- Sistema formado por los piñones de leva y cigüeñal, la correa dentada y el tensor de la correa dentada (este sistema puede ser remplazado por un sistema de engranajes): está diseñado para sincronizar el movimiento del cigüeñal y el del árbol de levas, de acuerdo a una relación establecida en el diseño del MCI.
- Árbol de levas: este sistema funciona como un programador mecánico para controlar la apertura de las válvulas, en algunos MCI también controla la distribución de aceite y la bomba de combustible.
- Cigüeñal: es un eje que posee codos y contrapesos, su función es, utilizando una biela y una manivela, transformar el movimiento rectilíneo alternativo del pistón en un movimiento rotatorio.
- Biela: esta pieza del MCI está conectada directamente al pistón, y transmite la energía del movimiento de este, a través de una manivela, al cigüeñal.
- Pistón: es un émbolo que se mueve por acción de la energía de combustión del MCI. Está ubicado dentro del cilindro en el cual se realiza la combustión.
- Válvula de descarga: es la válvula de la cámara de combustión del motor que abre y cierra la abertura para la salida de los gases de escape.
- Válvula de entrada: se encarga de abrir y cerrar la abertura de la cámara de combustión por donde entra la mezcla aire – combustible.
- Balancín: es el sistema que abre y cierra una válvula por acción de una leva.
- Ajustador hidráulico: permite la regulación del ángulo de una leva.
- Colector de aceite: es el depósito en el cual se ubica el aceite que lubrica el MCI.
- Bomba de aceite: se encarga de llevar el aceite del colector a las partes móviles del MCI (normalmente a los pistones).

2. Componentes del GI.

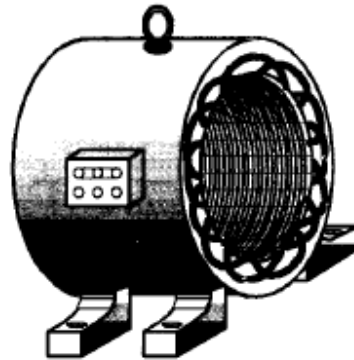
En las figuras C.15 a C.17 se observan los componentes principales del GI: el estator, el rotor y el capacitor (foto (a) de la figura C.17) o banco de capacitores (foto (b) de la figura C.17), dependiendo de si el generador es monofásico o trifásico, respectivamente.

Figura C.15 Rotor de un GI



Fuente: FRAILE MORA Jesús, Maquinas Eléctricas, página 261

Figura C.16 Estator de un GI



Fuente: IDEM, página 262

Figura C.17 Capacitor y banco de capacitores



Fuente: www.nierner.com.br/site/fator_potencia.htm

- Rotor: es la parte móvil del generador, en un grupo electrógeno, su eje está conectado al del MCI, posee conductores eléctricos que, en presencia de un campo magnético, conducen la corriente producida por la transformación de la potencia mecánica en potencia eléctrica. También posee un material ferromagnético (conductor de magnetismo), el cual se encarga de transmitir la potencia al estator, a través del campo magnético que fluye entre ellos.

- Estator: es una estructura hecha en un material ferromagnético que posee una o varias bobinas hechas con un conductor eléctrico. Este componente del GI, recibe la potencia del rotor a través del campo magnético entre ellos y la transforma en corriente y/o tensión eléctrica (si sus bornes están sin carga eléctrica sólo producirá tensión eléctrica).
- Capacitor o banco de capacitores: son los responsables de producir la corriente reactiva necesaria para generar la magnetización del GI, debido a que este tipo de generador no tiene un imán permanente ni otra fuente de magnetismo.

3. Principios de operación del MCI

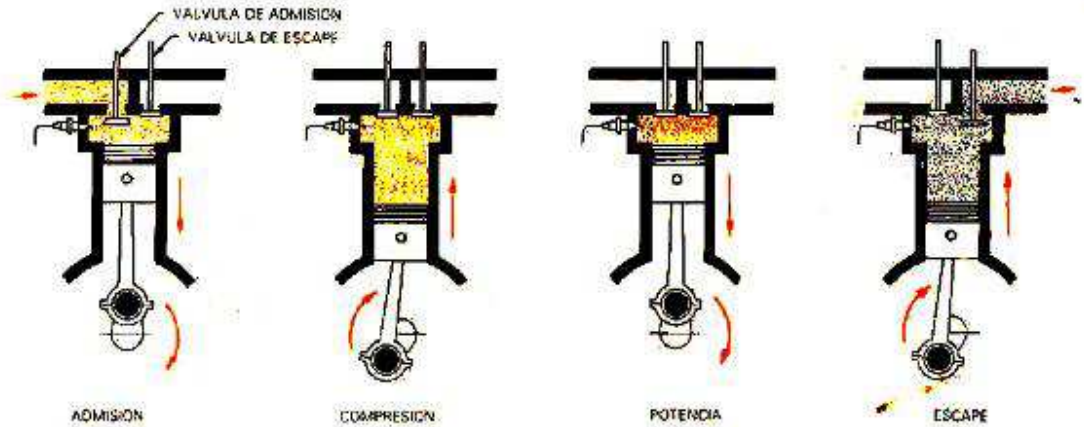
En general, el principio de operación de un MCI se puede resumir de la siguiente manera: el combustible es mezclado con aire (en muchos MCI este proceso lo realiza el carburador) , luego esta mezcla es aspirada a través de una válvula en cada cilindro, dentro del cual una chispa produce la combustión en la mezcla; entonces, la presión de los gases empujan el pistón hacia afuera del cilindro, el movimiento lineal del pistón es transformado en movimiento rotacional en el cigüeñal a través del sistema biela – manivela, hasta que este mismo sistema, produce que el pistón comience el movimiento de vuelta al cilindro. En uno de estos movimientos de retorno del pistón se expulsan los gases de escape a través de una válvula del cilindro.

Este proceso posee ciclos diferentes dependiendo de si se trata de un MCI de 4 o de 2 tiempos, a continuación se explica cada uno de estos ciclos:

- Ciclo de 4 tiempos: como puede observarse en la figura C.18, en estos MCI, el movimiento del pistón consta de 4 carreras diferentes:
 - ✓ Admisión: aquí, se abre la válvula de admisión y el pistón comienza su movimiento en el punto muerto inferior (el más cercano a las válvulas) y termina en el punto muerto superior. En esta etapa se aspira la mezcla hacia el cilindro.
 - ✓ Compresión: en este caso se mantienen las dos válvulas cerradas para evitar que la mezcla salga del cilindro. En este caso, el movimiento del pistón es inverso al de la carrera de admisión. En esta etapa se comprime la mezcla para luego producir la combustión de esta en la siguiente etapa (carrera de potencia).
 - ✓ Potencia (ignición y expansión de la mezcla aire – combustible). Aquí también se mantienen las válvulas cerradas por la misma razón que en la carrera de compresión. Comienza cuando la mezcla es encendida por una chispa, luego la combustión produce un cambio de presión en la mezcla que empuja al pistón en la misma dirección (y comenzando y terminando en los mismos puntos) que en la carrera de admisión.

- ✓ Escape. En esta carrera del pistón, se abre la válvula de descarga, su movimiento es idéntico al de la carrera de compresión. Aquí, se expulsan los productos de la combustión, para que el MCI pueda volver a comenzar su ciclo de operación.

Figura C.18 Ciclos de un MCI de 4 tiempos

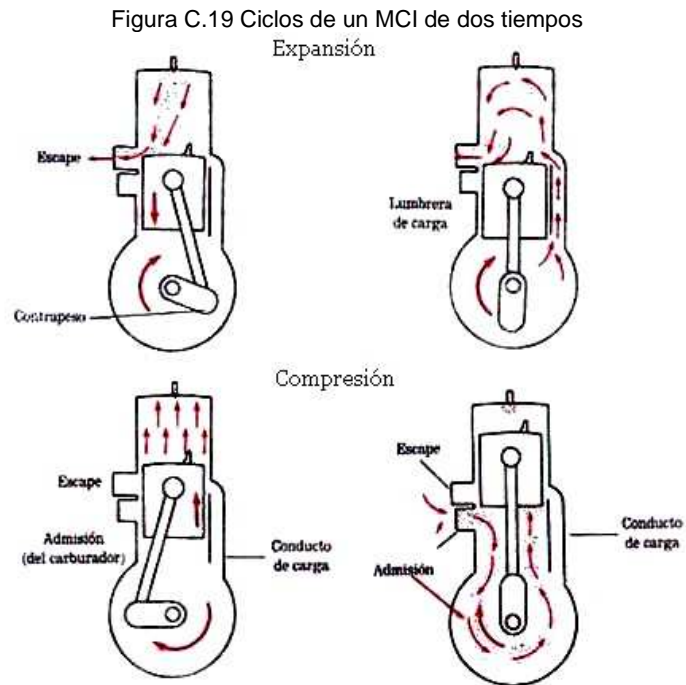


Fuente: www.clubmk1.com/Paginas/Datos/Funcion_Motor.html

De este modo, en este tipo de MCI, el ciclo completo dura dos vueltas del cigüeñal para un motor monocilíndrico. Además, en este tipo de MCI, un sistema que conecta el cigüeñal con el árbol de levas permite el control mecánico de la apertura de las válvulas del (de los) cilindro(s) y en algunos casos de otros sistemas del MCI como el de distribución de aceite y la bomba de combustible.

- Ciclo de 2 tiempos: en la figura C.19 puede observarse el esquema de un cilindro de un MCI de 2 tiempos. En este tipo de MCI, los mismos 4 procesos presentes en el MCI de 4 tiempos, ocurren en carreras del pistón, estas carreras son:
 - ✓ Expansión: aquí ocurren los procesos de potencia, el proceso de escape y parte del proceso de admisión, de la siguiente manera: cuando ocurre la ignición, se comienza el proceso de potencia hasta que el pistón deja abierto el tubo de escape (o, si el cilindro tiene válvula de escape, hasta que esta válvula sea abierta por el sistema), en ese momento se produce el escape y luego comienza la admisión, cuando el pistón deja abierto el tubo de admisión (o el sistema abre dicha válvula, si la hay) que ayuda a barrer los gases de la combustión a partir del momento en el cual la mezcla fresca pasa a través del puerto de transferencia (conducto de carga) por la acción del pistón.

- ✓ **Compresión:** aquí continúa la admisión y se realiza la compresión, puesto que el sistema comprime la mezcla al mismo tiempo que succiona una nueva.



Fuente: <http://www-org.autocity.com/documentos-tecnicos/?cat=3&codigoDoc=271>

4. Principio de operación del GI

Ahora, el principio de operación de un GI se expresa de la siguiente manera: si hay presente un campo magnético inicial en el hierro del generador, y el eje de la máquina se mueve por medio de una fuente externa (como un MCI) a una velocidad superior a la de sincronismo, el movimiento del campo magnético producirá una corriente inicial (que oscila a la frecuencia de sincronismo) en los conductores del rotor, esta corriente producirá otra corriente en el estator por inducción electromagnética, la cual cargará al capacitor o banco de capacitores conectado(s) a los bornes del estator, de modo que se producirá una corriente reactiva que a su vez produce el aumento de la intensidad del campo magnético del generador, de modo tal que el movimiento de este campo comienza a generar más corriente hasta que el sistema se estabiliza y se comporta como una carga constante para la fuente externa de potencia, la cual, en un grupo electrógeno, es un MCI.

Básicamente, existen tres formas de proveerle al GI el campo magnético inicial: con un circuito inversor y una batería, con un pequeño generador o mediante una conexión con una red eléctrica. El grupo electrógeno del Laboratorio de

Máquinas Térmicas Alternativas fue diseñado para utilizar la primera opción, de modo que posee unos bornes en los cuales se conecta una batería cuando el GI ha perdido su campo magnético residual.

C.3.7. Abrir el video de la práctica teórica 2

Ahora si en el menú de la práctica teórica 2 oprimimos el botón “Abrir video”, como se muestra en la figura C.20, se abre un video tutorial, que complementa de manera ilustrativa lo expuesto en el documento tutorial.

Figura C.20 Abrir el video de la práctica teórica 2



Fuente: El autor

C.3.8. Presentar evaluación de la práctica teórica 2

Si en el menú de la práctica teórica 2 oprimimos el botón “Presentar test” (como se ve en la figura C.21), se abre un documento que evalúa lo expuesto en los tutoriales.

Figura C.21 Presentar evaluación de la práctica teórica 2



Fuente: El autor

C.4 CÁLCULO DE LOS PARÁMETROS DE SALIDA DE UN GRUPO GENERADOR UTILIZANDO LA PLATAFORMA DE SIMULACIÓN

Pasos a seguir:

C.4.1. Abrir SCILAB.

Para esto se da doble clic en el ícono del escritorio que se muestra en la figura C.1.

C.4.2. Cambiar el directorio actual.

Este paso es necesario para facilitar el paso C.4.3. Para esto se da clic en el icono del menú superior de la consola de Scilab, de modo que aparece una subventana de selección de directorio (en la cual se debe navegar hasta encontrar el directorio donde están guardados los archivos y carpetas de la plataforma de simulación) tal como lo muestra la figura C.2.

C.4.3. Ejecutar el programa de simulación.

Digitar en la consola de SCILAB la siguiente instrucción: “exec Programa_de_Inicio.sce”, sin las comillas, tal como aparece en la figura C.3.

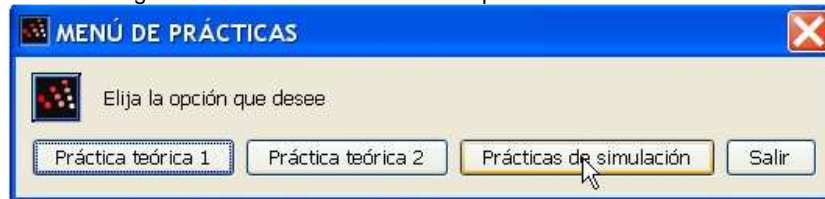
C.4.4. Abrir el menú de prácticas

Aparece el menú principal, el cual se puede observar en la figura C.7. Escoger la opción “Realizar prácticas”

C.4.5. Abrir la ventana de las prácticas de simulación

Luego de dar clic al botón mencionado en el paso C.4.4, aparece el menú de prácticas, entonces, se debe escoger “Prácticas de simulación”, como puede verse en la figura C.22.

Figura C.22 Abrir la ventana de las prácticas de simulación



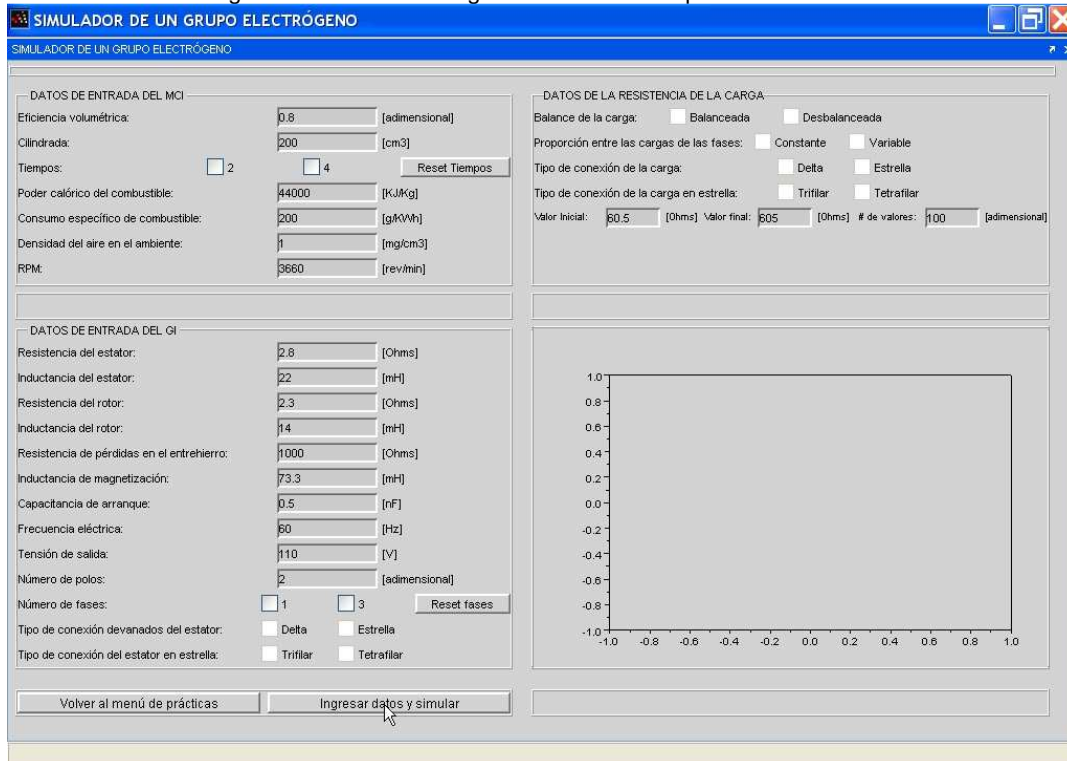
Fuente: El autor

C.4.6. Ingreso de los datos de entrada de la simulación

Se puede observar en la figura C.23 que la plataforma de simulación posee unos datos de entrada por defecto, los cuales se recomiendan para ser utilizados en las primeras prácticas, mientras el estudiantes se familiariza con los valores posibles que puede tomar cada una de estas variables.

Sólo hasta que se ingresan todos los valores necesarios, es posible comenzar el proceso de simulación oprimiendo el botón ingresar datos y simular el cual aparece seleccionado por el cursor en la figura C.23.

Figura C.23 Estado de ingreso de datos de la plataforma de simulación



Fuente: El autor

En la plataforma de simulación, en su estado de ingreso de datos de entrada (figura C.23), existen variables deshabilitadas por defecto, las cuales sólo se habilitan cuando se eligen ciertos valores de otras variables, de la siguiente manera:

Si el valor elegido para el número de fases del GI es “3”, entonces se habilita la variable de entrada “balance de la carga”.

Cuando se habilita la variable “balance de la carga” y se elige que esta sea “desbalanceada”, entonces se habilitan las variables de entrada “Tipo de conexión devanados del estator”, “Proporción entre las cargas de las fases” y “Tipo de conexión de la carga”.

Cuando está habilitada la variable “Tipo de conexión devanados del estator” y se elige que estos estén conectados en “Estrella”, entonces se habilita la variable “Tipo de conexión del estator en estrella”.

En el momento en el cual, estando habilitada la variable “Tipo de conexión del estator en estrella”, se elige que la conexión del estator en estrella sea “Tetrafilar”, aparece y se habilita la variable “R del neutro del GI”, con su respectivo valor por defecto.

Ahora, si, estando habilitada la variable “Proporción entre las cargas de las fases”, se elige que esta sea “Constante”, aparecen y se habilitan las variables de entrada “ $100 \cdot R_{Lf2} / R_{Lf1}$ ” y “ $100 \cdot R_{Lf3} / R_{Lf1}$ ” con sus respectivos valores por defecto, los cuales son la proporción porcentual de las resistencias de las cargas de las fases 2 y 3 del estator del GI (respectivamente), respecto a la resistencia de la carga de la fase 1 del estator del GI.

En cambio, si, estando habilitada la variable “Proporción entre las cargas de las fases”, se elige que esta proporción sea “Variable”, aparecen y se habilitan las variables de entrada “Valor Inicial fase 2”, “Valor Inicial fase 3”, “Valor Final fase 2”, “Valor Final fase 3” de la resistencia de la carga del GI, con sus respectivos valores por defecto.

Además, cuando está habilitada la variable “Tipo de conexión de la carga” y se elige que esta esté conectada en “Estrella”, entonces se habilita la variable “Tipo de conexión de la carga en estrella”.

Por último, si en el momento en el cual, estando habilitada la variable “Tipo de conexión la carga en estrella”, se elige que la conexión de la carga en estrella sea “Tetrafilas”, aparece y se habilita la variable “R del neutro de la carga”, con su respectivo valor por defecto.