



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE CUENCA  
CARRERA DE INGENIERÍA ELÉCTRICA**

**MODELACIÓN DE MÁQUINAS DE INDUCCIÓN  
HEXAFÁSICAS UTILIZANDO EL MÉTODO DE  
ELEMENTOS FINITOS**

Trabajo de titulación previo a la obtención del  
título de Ingeniero Eléctrico

**AUTORES: FERNANDO IVÁN CHACÓN GOMEZCOELLO  
ELFER MICHAEL TORRES CONDE**

**TUTOR: ING. JOSÉ MANUEL ALLER CASTRO, PhD**

Cuenca - Ecuador  
2022

## CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Fernando Iván Chacón Gomezcoello con documento de identificación N° 0105458053 y Elfer Michael Torres Conde con documento de identificación N° 1105148389; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 17 de marzo del 2022

Atentamente,



---

Fernando Iván Chacón Gomezcoello.

0105458053



---

Elfer Michael Torres Conde.

1105148389

# CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, Fernando Iván Chacón Gomezcoello con documento de identificación N° 0105458053 y Elfer Michael Torres Conde con documento de identificación N° 1105148389, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: “Modelación de máquinas de inducción hexafásicas utilizando el método de elementos finitos”, el cual ha sido desarrollado para optar por el título de: Ingeniero Eléctrico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 17 de marzo del 2022

Atentamente,

---

Fernando Iván Chacón Gomezcoello.

---

Elfer Michael Torres Conde.

0105458053

1105148389

## CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, José Manuel Aller Castro con documento de identificación N° 0151561800, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: MODELACIÓN DE MÁQUINAS DE INDUCCIÓN HEXAFÁSICAS UTILIZANDO EL MÉTODO DE ELEMENTOS FINITOS, realizado por Fernando Iván Chacón Gomezcoello con documento de identificación N° 0105458053 y por Elfer Michael Torres Conde con documento de identificación N° 1105148389, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 17 de marzo del 2022

Atentamente,



---

Ing. José Manuel Aller Castro. PhD  
0151561800

Cuenca, marzo de 2022

## DEDICATORIA

*Al finalizar este proyecto técnico quiero agradecer primeramente a Dios, por guiarme por el camino correcto. Mi proyecto técnico dedico a mis padres Hernán y Yolanda por el apoyo incondicional y ser el motor principal en mi vida, a mi esposa Silvia y mi hijo Joaquín que son mi inspiración y mi fuerza, a mi hermano Bryan y a mi familia en general, que de una u otra manera siempre me han brindado su apoyo. A mis amigos por todos los buenos y malos momentos vividos en la universidad.*

*Fernando Ch.*

Cuenca, marzo de 2022

## DEDICATORIA

*Al finalizar mi tesis, primeramente, quiero dar gracias a la Virgen Del Cisne por tantos milagros que ha realizado en mi vida y me ha ayudado en todo momento, dedico esta tesis a mi familia, en especial a mi abuelo Santos Torres por haberme inculcado buenos valores de superación, honestidad, sencillez y lealtad, a mis tíos: Rolando Torres, Alfonso Torres, Jesús Torres, Estela Torres, Beatriz Torres, Vicente Torres, mis primos: Ramiro Sánchez, Olger Torres, a la familia Pauta Pineda, a la familia Jumbo Castillo, quienes hicieron posible de una u otra manera que fuera posible llegar a cumplir esta meta tan anhelada. De igual manera dar gracias a mis amigos quienes formaron parte de esta trayectoria, compartiendo conocimientos, alegrías, triunfos, diversiones. De igual forma dar gracias a mi amigo y compañero de tesis Fernando Chacón, por toda la responsabilidad y compañerismo tomado en este proyecto de titulación.*

*Michael T.*

Cuenca, marzo de 2022

## AGRADECIMIENTOS

*Agradecemos primeramente a Dios por brindarnos la salud y la fuerza para llegar a alcanzar nuestras metas. A nuestro director de tesis Dr. José Manuel Aller Castro que por sus conocimientos y experiencias nos brindó todo su apoyo para culminar este proyecto, al igual que a nuestros compañeros de universidad que siempre siempre nos brindaron ayuda en todo momento.*

*Fernando I. Chacón G.*

*Elfer M. Torres C.*

# RESUMEN

En el presente documento se realiza la modelación de máquinas de inducción hexafásicas utilizando el método de elementos finitos, mediante la herramienta de modelación Elmer FEM que utiliza el método de los elementos finitos. En este trabajo, en primer lugar se definen las especificaciones de la máquina eléctrica de inducción hexafásica, se crea la geometría tanto del rotor como del estátor. Se genera un malla a la geometría y esta se exporta al programa de simulación Elmer FEM. Ahora, mediante el script del programa Elmer FEM se desarrolla el modelo de simulación en estado estacionario. Esto para verificar la validez del modelo por medio del programa de postprocesamiento Paraview.

Por último, se genera la simulación transitoria en el tiempo, verificando mediante el programa Paraview que los flujos magnéticos estén correctamente modelados. se importa a la plataforma de Matlab los resultados de la simulación y se analizan los resultados de par, corrientes y voltajes del estátor.



# ABSTRACT

In this paper, the modeling of hexa-phase induction machines is performed using the finite element method, by means of the Elmer FEM modeling tool that uses the finite element method. In this work, first of all, the specifications of the electric hexaphase induction machine are defined, the geometry of both the rotor and the stator are created. A mesh is generated to the geometry and this is exported to the Elmer FEM simulation program. Now, using the Elmer FEM program script, the steady state simulation model is developed. This is realized to verify the validity of the model by means of the post-processing program Paraview.

Finally, the time transient simulation is generated, verifying through the Paraview program that the magnetic fluxes are correctly modeled. The simulation results are imported to the Matlab platform and the results of torque, currents and voltages of the stator are analyzed.

# PREFACIO

Este proyecto técnico modela una máquina de inducción hexafásica mediante el método de elementos finitos, donde, el proceso se lo realiza mediante varios programas de modelación geométrica y simulación en elementos finitos. GMSH es el software que utilizado la modelación geométrica y se genera la malla en  $2D$ . Elmer realiza toda la simulación con el método de elementos finitos y mediante la plataforma Paraview se visualiza y analizan los resultados de la simulación. Todos los programas mencionados son de código abierto. Es indispensable un conocimiento previo de todas estas herramientas de modelación para poder realizar una modelación y análisis exitoso de los convertidores electromecánicos. El objetivo de este trabajo consiste en comparar las características de operación de una máquina de inducción hexafásica modelada mediante elementos finitos con su modelo circuital en vectores espaciales.

# Índice general

INTRODUCCIÓN . . . . .	1
1 Problema de Estudio y Objetivos . . . . .	3
1.1 Problema de Estudio . . . . .	3
1.2 Grupo Objetivo . . . . .	3
1.3 Objetivos . . . . .	3
1.3.1 Objetivo General . . . . .	3
1.3.2 Objetivos Específicos . . . . .	4
1.4 Metodología Aplicada . . . . .	4
2 Fundamentos Teóricos . . . . .	5
2.1 Historia de la máquina de inducción . . . . .	5
2.2 Características generales . . . . .	5
2.3 La máquina de inducción hexafásica . . . . .	5
2.4 Funcionamiento de la máquina de inducción . . . . .	6
2.5 Principales partes de una máquina eléctrica de inducción hexafásica . . . . .	7
2.5.1 Estátor . . . . .	7
2.5.2 Rotor . . . . .	8
2.6 Modelo de la máquina de inducción hexafásica con el método de vectores espaciales . . . . .	9
2.6.1 Vectores espaciales . . . . .	11
3 Herramientas para la modelación . . . . .	13
3.1 Elmer FEM . . . . .	13
3.2 Características principales . . . . .	13
3.3 Estructura de Elmer . . . . .	14
3.4 Gmsh . . . . .	15
3.4.1 Módulo de geometría . . . . .	16
3.4.2 Módulo de mallado . . . . .	16
3.4.3 Principales ventajas de Gmsh . . . . .	17
3.5 Paraview . . . . .	17
3.5.1 Visualización del modelo en Paraview . . . . .	18
3.5.2 Interface gráfica de Paraview . . . . .	18
4 Simulación Elmer FEM . . . . .	19
4.1 Especificaciones de la Máquina Eléctrica de Inducción Hexafásica . . . . .	19
4.2 Geometría y mallado mediante GMSH . . . . .	20
4.2.1 Geometría del estátor . . . . .	21
4.2.2 Geometría del rotor . . . . .	33
4.2.3 Importación a Elmer FEM . . . . .	39

4.3	Simulación mediante Elmer FEM . . . . .	40
4.3.1	Simulación en estado estacionario . . . . .	40
4.3.2	Simulación transitoria en el tiempo . . . . .	47
5	Simulación mediante el método de vectores espaciales . . . . .	54
5.1	Modelo circuital . . . . .	54
6	Análisis de Resultados. . . . .	58
6.1	Análisis en estado estacionario . . . . .	58
6.2	Análisis de simulación transitoria en el tiempo . . . . .	59
6.3	Simulación de la MI hexafásica con el método de vectores espaciales . . . . .	64
6.3.1	Corrientes del estátor . . . . .	64
6.4	Comparación mediante el modelo de elementos finitos y el modelo circuital de vectores espaciales . . . . .	65
7	Conclusiones y Recomendaciones . . . . .	67
7.1	Conclusiones . . . . .	67
7.2	Recomendaciones . . . . .	68
A	Anexos . . . . .	69
A.1	Código de la simulación en estado estacionario . . . . .	69
A.2	Código para simulación transitoria en el tiempo . . . . .	72
A.3	Parámetros de la MI . . . . .	78
	Bibliografía . . . . .	81

# Índice de figuras

2.1	Modelo de desplazamiento de fases de la máquina de inducción hexafásica [1]	6
2.2	Tensión hexafásica desfasadas[2]	7
2.3	Estátor	8
2.4	Rotor	8
2.5	Modelo de la MI referido al estátor [3]	12
3.1	Interface gráfica de ElmerGUI.	14
3.2	Extructura de Elmer FEM [4]	15
3.3	Plataforma Gmsh	16
3.4	Módulo de mallado Gmsh	17
3.5	Visualización modelo en Paraview [5]	18
3.6	Interfaz gráfica de Paraview	18
4.1	Dimensiones principales [6]	19
4.2	Geometría original de una máquina eléctrica de inducción [6]	21
4.3	Sección periódica del diseño del devanado del estátor	21
4.4	Dimensiones de la ranura del estator. [6]	22
4.5	Geometría de una máquina eléctrica de inducción trifásica [6]	23
4.6	Geometría de la ranura del estátor	24
4.7	Geometría de la ranura del estátor	26
4.8	Ingreso de datos	27
4.9	Ingreso de puntos de la geometría del estátor	28
4.10	Puntos de la geometría del devanado del estátor.	28
4.11	Densidad de la malla del estátor	29
4.12	Líneas y curvas de la geometría del estátor	29
4.13	Unión de líneas y puntos del estátor	30
4.14	Bucle para formar toda la geometría	30
4.15	Geometría completa del estátor	31
4.16	Entidades físicas de la geometría del estátor	31
4.17	Nombres de las regiones del estátor	32
4.18	Regiones del estátor	32
4.19	Malla resultante de la geometría del estátor	33
4.20	Dimensiones de la ranura del rotor	34
4.21	Ingreso de datos del rotor	35
4.22	Puntos de la ranura del rotor	35
4.23	Puntos de la ranura del rotor	35
4.24	Líneas y curvas de la geometría del rotor	36
4.25	Unión de líneas y puntos del rotor	36
4.26	Rotación de puntos definidos del rotor	37
4.27	Geometría de un polo del rotor	37

4.28	Entidades físicas de la geometría del rotor . . . . .	38
4.29	Generación de la malla del rotor . . . . .	38
4.30	Malla resultante de la geometría del rotor . . . . .	39
4.31	Comando de importación . . . . .	39
4.32	Integración de las mallas . . . . .	40
4.33	Modelo de la MI Hexafásica importada en ElmerGUI . . . . .	41
4.34	Sección del encabezado del SIF . . . . .	42
4.35	Ingreso de materiales . . . . .	42
4.36	Ingreso del material acero eléctrico . . . . .	43
4.37	Fuerzas externas . . . . .	44
4.38	Ingreso de los cuerpos . . . . .	44
4.39	Solucionadores . . . . .	46
4.40	Límite interno y externo . . . . .	46
4.41	Límites periódicos . . . . .	47
4.42	Ingreso de los parámetros de la MI hexafásica . . . . .	47
4.43	Encabezado de la simulación transitoria . . . . .	48
4.44	Materiales para la simulación transitoria en el tiempo . . . . .	48
4.45	Fuerzas corporales de la simulación transitoria en el tiempo . . . . .	49
4.46	Cuerpos para la simulación transitoria en el tiempo . . . . .	50
4.47	Componentes . . . . .	50
4.48	Solucionadores del estado transitorio . . . . .	51
4.49	Circuitos . . . . .	51
4.50	Diagrama de circuito de fase [6] . . . . .	52
4.51	Kirchhoff 1 y Kirchhoff 2 . . . . .	52
4.52	Diagrama de bobinado del rotor tipo jaula de ardilla . . . . .	53
5.1	Fuente de alimentación del modelo circuital . . . . .	54
5.2	Modelo dinámico . . . . .	55
5.3	Modelo completo de la máquina de inducción en Simulink . . . . .	55
5.4	Diagrama de las corrientes del modelo dinámico . . . . .	56
5.5	Diagrama del cálculo del par eléctrico . . . . .	56
5.6	Diagrama de enlace de flujo y la fem . . . . .	57
6.1	Densidad de flujo en estado estacionario . . . . .	58
6.2	Líneas de flujo magnético en estado estacionario . . . . .	59
6.3	Tiempo de simulación en estado estacionario . . . . .	60
6.4	Interface de usuario principal de Paraview . . . . .	61
6.5	Potencial del vector magnético . . . . .	61
6.6	Densidad de flujo B . . . . .	62
6.7	Densidad de corriente . . . . .	62
6.8	Par eléctrico de la MI con el FEM . . . . .	63
6.9	Corrientes del estátor de la MI mediante la FEM . . . . .	63
6.10	Voltajes del estátor de la MI . . . . .	64
6.11	Corrientes del estátor con el método circuital . . . . .	64
6.12	Par eléctrico con el método de vectores espaciales . . . . .	65
6.13	Gráfico de las corrientes del estátor mediante la FEM . . . . .	65
6.14	Comparación de par eléctrico . . . . .	66
A.1	Estado estacionario 1 . . . . .	69

A.2	Estado estacionario 2 . . . . .	70
A.3	Estado estacionario 3 . . . . .	71
A.4	Simulación transitoria en el tiempo 1 . . . . .	72
A.5	Simulación transitoria en el tiempo 2 . . . . .	73
A.6	Simulación transitoria en el tiempo 3 . . . . .	74
A.7	Simulación transitoria en el tiempo 4 . . . . .	75
A.8	Simulación transitoria en el tiempo 5 . . . . .	76
A.9	Simulación transitoria en el tiempo 6 . . . . .	77
A.10	Parámetros de la MI 1 . . . . .	78
A.11	Parámetros de la MI 2 . . . . .	79
A.12	Parámetros de la MI 3 . . . . .	80

# Índice de tablas

4.1	Parámetros de la máquina eléctrica . . . . .	20
4.2	Parámetros de la geometría del estátor. . . . .	22
4.3	Parámetros de la geometría del rotor [6] . . . . .	34



# Glosario

VBR	Voltage behind reactance
MI	Máquina de inducción
SIF	Archivo de Entrada del Solucionador

# INTRODUCCIÓN

La máquina de inducción tiene un gran número de aplicaciones en los sectores industriales, residenciales y comerciales. Actualmente es la más utilizada como accionamiento de bombas, compresores, ventiladores, equipos electrodomésticos. Sus características operativas, la facilidad de mantenimiento y su robustez eléctrica y mecánica la han posicionado como el motor más utilizado en todo el mundo.

Hoy en día se plantea la conveniencia de utilizar máquinas de inducción de un mayor número de fases para incrementar la confiabilidad de estos motores y reducir las tensiones línea a línea, con las ventajas que esto tiene sobre el aislamiento [7, 8, 9, 10, 11].

Existen varios modelos dinámicos y de régimen permanente de la máquina de inducción [12, 13, 14], que permiten el análisis de su comportamiento mediante ecuaciones diferenciales y algebraicas. Sin embargo, cuando la máquina es alimentada por fuentes de electrónica de potencia, es conveniente disponer de modelos circuitales dinámicos y en esta dirección resaltan los modelos VBR (Voltage Behind Reactance) [15, 16, 17] que permiten representar los convertidores en herramientas tipo SIMULINK, ATP, SPICE, etc.

El modelo de [15] es un desarrollo VBR de una máquina de inducción hexafásica simétrica, pero no utiliza técnicas vectoriales modernas sino la clásica transformación  $dq$ . Por otra parte [16] desarrolla los modelos de máquinas asimétricas mediante esta técnica empleando vectores espaciales en su planteamiento. Más recientemente [17], desarrollaron el modelo de una máquina síncrona de imán permanente y polos salientes de seis fases. Por otra parte en [18] presenta un procedimiento para poder obtener los parámetros de un circuito equivalente de una máquina de inducción de seis fases mediante la prueba de rotor bloqueado (LTR) y prueba de estator eliminado (RST). Mientras en [19], el autor evalúa el rendimiento de máquinas de inducción de tres fases y de seis fases alimentadas por un inversor de seis pulsos de onda cuadrada.

En [20] el autor compara dos máquinas polifásicas, la primera de cinco fases y la segunda de seis fases. Estas pruebas se realizan mediante condiciones normales y en falla, introduciendo en la entrada ondas senoidales y cuadrada. Se obtienen cambios repentinos de par, variación de velocidad, etc.

En [21] se realiza un análisis de máquinas de inducción operando como generador autoexcitado hexafásico, alimentando cargas resistivas y resistivo-inductivas.

En [22] se analiza mediante elementos finitos el par desarrollado por el motor de inducción de jaula de ardilla multifásico, utilizando ANSYS MAXWELL 2D. Se desarrolla la modelación en dos pasos: Obtener la densidad de flujo del entrehierro óptimo por una parte, y posteriormente determinar el diámetro de división óptima; es decir cuál es la relación óptima entre el diámetro de estator interior y el exterior.

En [14] se describe el modelo y operación de la máquina síncrona y se introducen los métodos para analizar las máquinas polifásicas. Por otra parte en [23], el autor presenta una investigación sobre el control de un motor de inducción de seis fases, en el cual mediante el análisis teórico y simulación puede demostrar y verificar el funcionamiento del modelo matemático.

Por otra parte y de gran interés para el desarrollo de este proyecto de investigación, es necesario referirse a una herramienta de código abierto y de libre distribución denominada o ELMER [24, 25]. Este programa permite realizar modelos muy detallados de la máquina de inducción con cualquier número de fases e incluir la dinámica, el comportamiento 3D y resolver las ecuaciones multifísicas que integran respuestas eléctricas, magnéticas, térmicas y de resistencia de materiales. En [26] el autor modela una PMSM utilizando software de elementos finitos Elmer. En este tutorial se desarrolla un documento explicativo de la programación necesaria para el modelado de la máquina eléctrica. En [6] se presenta el flujo de trabajo para modelar una máquina eléctrica de inducción de  $5\text{ kW}$  utilizando esta herramienta, con un ejemplo de un motor de inducción para una aplicación de soplado. Este trabajo constituye la base metodológica de esta investigación para realizar los modelos de elementos finitos de la máquina de inducción hexafásica.

# 1 Problema de Estudio y Objetivos

## 1.1. Problema de Estudio

El modelado preciso de las máquinas eléctricas de inducción es de gran interés debido a la enorme difusión de estos convertidores en los sectores industriales, comerciales y residenciales. Actualmente, se ha venido incrementando el interés en disponer de máquinas de inducción con un mayor número de fases [18, 19, 20, 21, 22]. Esto tiene ventajas en la reducción de las tensiones línea a línea y en la posibilidad de utilizar sistema tolerantes a fallas.

Las características de los convertidores electrónicos de potencia, también tienen ciertas ventajas en el incremento de su confiabilidad y reducción de las especificaciones de las componentes electrónicas, cuando se incrementa el número de ramas.

La modelación detallada de la máquina tiene retos importantes y dos vertientes principales: La modelación circuital y los modelos en elementos finitos [22, 21]. Ambas metodologías se complementan y en el análisis del comportamiento se suelen utilizar circuitos equivalentes estáticos o dinámicos, mientras que en el área de diseño se prefieren los elementos finitos que son mucho más precisos.

## 1.2. Grupo Objetivo

En este proyecto se desarrolla una modelación precisa de la máquina de inducción hexafásica en elementos finitos, para comprobar las características y precisiones de modelos circuitales hexafásicos desarrollados en trabajos anteriores del Grupo de Investigación en Energía de la Universidad Politécnica Salesiana [16, 17]. Por este motivo este proyecto es de gran utilidad para las investigaciones realizadas por este grupo y también tiene impacto en el desarrollo de este tipo de máquinas, tanto en Ecuador como en el resto del mundo.

## 1.3. Objetivos

### 1.3.1. Objetivo General

Comparar las características de operación de una máquina de inducción hexafásica modelada mediante elementos finitos con su modelo circuital en vectores espaciales.

### 1.3.2. Objetivos Específicos

- Obtener una geometría apropiada al diseño de la máquina de inducción hexafásicas.
- Realizar el modelo de la máquina de inducción hexafásica en elementos finitos utilizando herramientas computacionales de libre acceso.
- Determinar las características operativas de la máquina de inducción hexafásica mediante el modelo de elementos finitos.
- Comparación de los resultados obtenidos al simular el comportamiento de las máquinas de inducción hexafásica, mediante elementos finitos y en vectores espaciales.

## 1.4. Metodología Aplicada

Para esta investigación se desarrollaron diferentes metodologías y actividades para lograr alcanzar los objetivos propuestos:

- Con la metodología cualitativa se recolectó información y conocimientos sobre el campo de las máquinas eléctricas de inducción hexafásica.
- Se utilizan herramientas de simulación de elementos finitos, como el programa Elmer, con todas sus plataformas como: ElmerGUI, ElmerPost y ElmerSolver. Así mismo, se analizan herramientas paralelas a Elmer FEM para obtener resultados de la investigación para poder comparar y contrastar las soluciones obtenidas [24, 26, 6].
- Mediante la metodología descriptiva se desarrolla la geometría, descripción de las regiones físicas, el mallado adecuado al problema, las condiciones de contorno, la definición de los materiales y todas las configuraciones numéricas que requiere la simulación.
- Por último se utiliza en esta investigación la metodología aprehensiva para verificar, comparar y validar los resultados de la simulación en elementos finitos del modelo de la máquina de inducción hexafásica con modelos circuitales desarrollados en vectores espaciales mediante la técnica VBR (Voltage Behaind Reactance) [16, 17].

## **2 Fundamentos Teóricos**

### **2.1. Historia de la máquina de inducción**

El desarrollo de las máquinas eléctricas, empieza en el siglo XIX, con las investigaciones en electromagnetismo realizadas por grandes científicos como Oersted, Faraday, Henry, Lenz, Barlow y Maxwell, entre muchos otros. Las leyes que enunciaron, posteriormente y con gran rapidez fueron utilizadas para diseñar máquinas eléctricas para la producción de energía eléctrica y movimiento.

La máquina de inducción, desde que fue creada simultáneamente por Tesla y Ferranti [27], debido a sus características constructivas, operativas y de mantenimiento, se ha transformado en una herramienta fundamental para el desarrollo industrial. Esta máquina ha permitido un mejoramiento productivo tanto en la industria como en los sectores residenciales y comerciales.

### **2.2. Características generales**

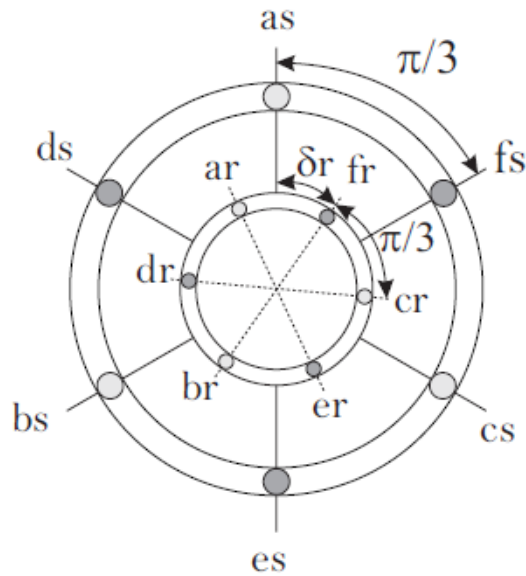
La máquina de inducción tiene un sin número de aplicaciones en nuestro medio y en la actualidad es la más utilizada en la parte de los accionamientos eléctricos, porque tiene grandes ventajas, especialmente debido a su robustez, confiabilidad y costo. La máquina de inducción se conforma de dos partes importantes: una giratoria denominada rotor y otra denominada estátor, que al ser alimentado mediante tensiones desfasadas, produce el campo magnético rotatorio que induce un campo en el rotor, para producir la iteración electromagnética [3].

Esta máquina ha ido evolucionando en su mejoramiento y rendimiento gracias a los avances de la tecnología, que se centran en su operatividad y regulación de velocidad por medio de la electrónica de potencia, creando controladores que mejoran las limitaciones que tenían originalmente para la regulación de las corrientes de arranque y de su velocidad [23, 28, 29, 30].

### **2.3. La máquina de inducción hexafásica**

Entre las máquinas polifásicas más estudiadas actualmente está la hexafásica, mantiene los beneficios y tiene una ventaja debido a que se puede modificar a partir de los diseños clásicos de una máquina trifásica [1]. Por otro lado, existen dispositivos de control, que permiten el control de velocidad y par de este tipo de máquinas. Las máquinas polifásicas pueden tener el desplazamiento entre sus fases de forma simétrica o asimétrica. Para el caso de estudio se modela una máquina de inducción

hexafásica, en la cual el desplazamiento entre las bobinas del estátor es de  $\pi/3$ , tal como se muestra en la Fig. 2.1.



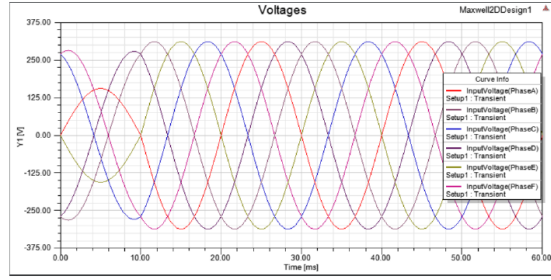
**Figura 2.1:** Modelo de desplazamiento de fases de la máquina de inducción hexafásica [1]

## 2.4. Funcionamiento de la máquina de inducción

La máquina de inducción nace de una aplicación a dar solución a los problemas de uso de la corriente alterna, como es de conocimiento en la actualidad la máquina de inducción se utiliza en la industria, comercio y en el hogar. La máquina de inducción hexafásica consta de 6 devanados monofásicos distribuidos en las ranuras del estátor. El principio de funcionamiento de la máquina de inducción es alimentar con corriente alterna en el estátor, esto hace que toda la energía fluya hacia o desde el estátor [3, 1].

Los flujos producidos por las corrientes que circulan en el estátor hacen que generen un campo magnético rotatorio en el rotor. “La velocidad de giro del campo magnético rotatorio está definida por la frecuencia de las corrientes inyectadas en el estátor de la máquina” [3, 1].

En la Fig. 2.2 se puede visualizar las tensiones desfasadas de una máquina de inducción hexafásica, como se muestra en el modelo de la Fig. 2.1 .



**Figura 2.2:** Tensión hexafásica desfasadas[2]

El devanado del estátor se conecta a una fuente, produciendo un campo magnético con magnitud constante [31], que gira a una velocidad síncrona alrededor de la periferia del rotor, en la Ecuación. 2.1 muestra la ecuación para determinar la velocidad síncrona en revoluciones por minuto [ $rpm$ ], o en la ecuación 2.2 en  $\left[\frac{rad}{s}\right]$ .

$$\eta = \frac{60f}{p} \quad (2.1)$$

Donde:

- f: es la frecuencia de la corriente en el devanado del estátor.
- p: es el número de polos.

$$w_s = \frac{2\pi f}{p} \quad (2.2)$$

Para el rotor, el deslizamiento para una máquina en operación está entre el 2% y 8% [32], se puede calcular de la siguiente forma:

$$S = \frac{\eta_{estator} - \eta_{rotor}}{\eta_{estator}} * 100\% \quad (2.3)$$

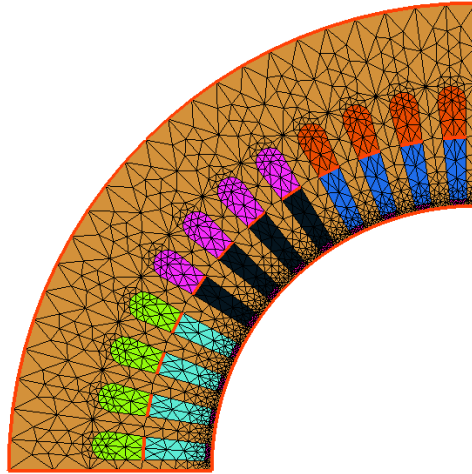
## 2.5. Principales partes de una máquina eléctrica de inducción hexafásica

Las partes principales de la máquina son el estátor y el rotor.

### 2.5.1. Estátor

El estátor es la parte estacionaria de la máquina de inducción, está formado por un conjunto de ranuras, que son de acero de alta permeabilidad, [31] se localizada dentro de la carcasa de la máquina. En la Fig. 2.3 se muestra 1 polo del estátor.

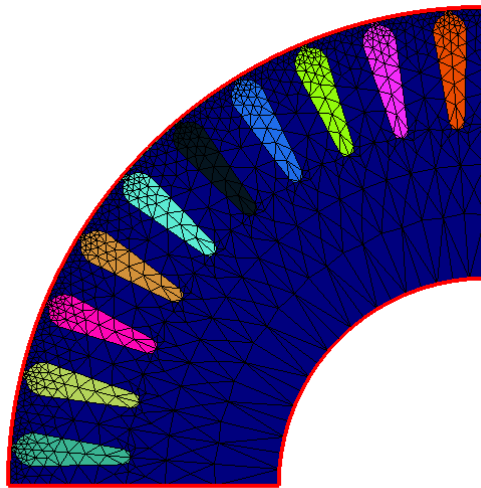




**Figura 2.3:** Estátor

### 2.5.2. Rotor

Para el ejemplo, se utiliza un modelo de rotor tipo jaula de ardilla. Este tipo de devanado es el más utilizado en la industria, debido a la facilidad de mantenimiento, bajo costo y simplicidad de construcción, etc. [2] En la Fig. 2.4 se puede visualizar 1 polo del rotor.



**Figura 2.4:** Rotor

## 2.6. Modelo de la máquina de inducción hexafásica con el método de vectores espaciales

Las ecuaciones en coordenadas primitivas del modelo circuital de la máquina hexafásica [3], se basa en las siguientes ecuaciones diferenciales:

$$[v] = [R] [i] + \rho [\lambda] = [R] [i] + [L(\theta)] \rho [i] + \dot{\theta} [\tau(\theta)] [i] \quad (2.4)$$

$$T_e - T_m = \frac{1}{2} [i]^t [\tau] [i] - T_m = J\ddot{\theta} + \rho\dot{\theta} \quad (2.5)$$

donde:

$$[v] = \begin{bmatrix} [V_e] \\ [V_r] \end{bmatrix} = \begin{bmatrix} [v_{ae} \ v_{be} \ v_{ce} \ v_{de} \ v_{ee} \ v_{fe}]^t \\ [v_{ar} \ v_{br} \ v_{cr} \ v_{dr} \ v_{er} \ v_{fr}]^t \end{bmatrix} \quad (2.6)$$

$$[i] = \begin{bmatrix} [i_e] \\ [i_r] \end{bmatrix} = \begin{bmatrix} [i_{ae} \ i_{be} \ i_{ce} \ i_{de} \ i_{ee} \ i_{fe}]^t \\ [i_{ar} \ i_{br} \ i_{cr} \ i_{dr} \ i_{er} \ i_{fr}]^t \end{bmatrix} \quad (2.7)$$

$$[\lambda] = \begin{bmatrix} [\lambda_e] \\ [\lambda_r] \end{bmatrix} = \begin{bmatrix} [\lambda_{ae} \ \lambda_{be} \ \lambda_{ce} \ \lambda_{de} \ \lambda_{ee} \ \lambda_{fe}]^t \\ [\lambda_{ar} \ \lambda_{br} \ \lambda_{cr} \ \lambda_{dr} \ \lambda_{er} \ \lambda_{fr}]^t \end{bmatrix} \quad (2.8)$$

$$[R] = \begin{bmatrix} [R_{ee}] & [R_{er}] \\ [R_{re}] & [R_{rr}] \end{bmatrix} = \begin{bmatrix} R_e [I] & [0] \\ [0] & R_r [I] \end{bmatrix} \quad (2.9)$$

$$[L(\theta)] = \begin{bmatrix} [L_{ee}] & [L_{er}(\theta)] \\ [L_{re}(\theta)] & [L_{rr}] \end{bmatrix} \quad (2.10)$$

$$[L(\theta)] = \begin{bmatrix} L_{\sigma e} [I] + L_{me} [S] & L_{er} [C(\theta)] \\ L_{er} [C(\theta)]^t & L_{\sigma r} [I] + L_{mr} [S] \end{bmatrix} \quad (2.11)$$

$$[\tau(\theta)] = \begin{bmatrix} \frac{\delta}{\delta\theta} [L_{ee}] & \frac{\delta}{\delta\theta} [L_{er}(\theta)] \\ \frac{\delta}{\delta\theta} [L_{re}(\theta)] & \frac{\delta}{\delta\theta} [L_{rr}] \end{bmatrix} \quad (2.12)$$

$$[\tau(\theta)] = \begin{bmatrix} [0] & L_{er} \frac{\delta}{\delta\theta} [C(\theta)] \\ L_{er} \frac{\delta}{\delta\theta} [C(\theta)]^t & [0] \end{bmatrix} \quad (2.13)$$

La matriz identidad  $[I]$ , es donde todos los elementos de la diagonal principal son 1 y el resto de los elementos de la matriz son 0. La matriz simétrica  $[S]$ , donde la diagonal principal es 1 ya que el  $\cos(0)$  es igual a 1, el resto de los componentes de la matriz es el resultado que correspondería a la simetría de un sistema hexafásico desfasado  $\frac{\pi}{3}$ . La matriz cíclica  $[C]$ , es la que tiene una secuencia o un ciclo de trabajo. [3]

$$[I] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

$$[S] = \begin{bmatrix} 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} & -\frac{1}{2} & -1 & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & -\frac{1}{2} & -1 \\ -1 & -\frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -1 & -\frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & -1 & -\frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix} \quad (2.15)$$

$$[0] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.16)$$

$$[C(\theta)] = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 \\ a_6 & a_1 & a_2 & a_3 & a_4 & a_5 \\ a_5 & a_6 & a_1 & a_2 & a_3 & a_4 \\ a_4 & a_5 & a_6 & a_1 & a_2 & a_3 \\ a_3 & a_4 & a_5 & a_6 & a_1 & a_2 \\ a_2 & a_3 & a_4 & a_5 & a_6 & a_1 \end{bmatrix} \quad (2.17)$$

donde:

$$\begin{aligned} a_1 &\rightarrow \cos(\theta) \\ a_2 &\rightarrow \cos\left(\theta + \frac{\pi}{3}\right) \\ a_3 &\rightarrow \cos\left(\theta + \frac{2\pi}{3}\right) \\ a_4 &\rightarrow \cos(\theta + \pi) \\ a_5 &\rightarrow \cos\left(\theta + \frac{4\pi}{3}\right) \\ a_6 &\rightarrow \cos\left(\theta + \frac{5\pi}{3}\right) \end{aligned}$$

$$\frac{\delta}{\delta\theta} [C(\theta)] = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 \\ b_6 & b_1 & b_2 & b_3 & b_4 & b_5 \\ b_5 & b_6 & b_1 & b_2 & b_3 & b_4 \\ b_4 & b_5 & b_6 & b_1 & b_2 & b_3 \\ b_3 & b_4 & b_5 & b_6 & b_1 & b_2 \\ b_2 & b_3 & b_4 & b_5 & b_6 & b_1 \end{bmatrix} \quad (2.18)$$

donde:

$$\begin{aligned}
b_1 &\rightarrow -\sin(\theta) \\
b_2 &\rightarrow -\sin\left(\theta + \frac{\pi}{3}\right) \\
b_3 &\rightarrow -\sin\left(\theta + \frac{2\pi}{3}\right) \\
b_4 &\rightarrow -\sin(\theta + \pi) \\
b_5 &\rightarrow -\sin\left(\theta + \frac{4\pi}{3}\right) \\
b_6 &\rightarrow -\sin\left(\theta + \frac{5\pi}{3}\right)
\end{aligned}$$

A continuación, definimos los parámetros de modelo de la MI hexafásica.

$R_e$  → “resistencia de cada una de las bobinas del estátor”

$R_r$  → “resistencia de cada una de las bobinas del rotor”

$L_{\sigma e}$  → “inductancia de dispersión del estátor”

$L_{\sigma r}$  → “inductancia de dispersión del rotor”

$L_{me}$  → “inductancia de magnetización del estátor”

$L_{mr}$  → “inductancia de magnetización del rotor”

$L_{er}$  → “inductancia mutua de acoplamiento estátor-rotor”

### 2.6.1. Vectores espaciales

Las matrices simétricas  $[S]$  y las cíclicas “pueden ser diagonalizadas utilizando el método de autovalores y autovectores” [3] .

$$\vec{x}_s = \frac{1}{\sqrt{3}} \{x_a + \alpha x_b + \alpha^2 x_c + \alpha^3 x_d + \alpha^4 x_e + \alpha^5 x_f\} \quad (2.19)$$

donde:

$$\begin{bmatrix} x_x \\ x_y \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{12}} & -\frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{12}} & \frac{1}{\sqrt{12}} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} x_a \\ x_b \\ x_c \\ x_d \\ x_e \\ x_f \end{bmatrix} \quad (2.20)$$

El par eléctrico obtenemos de la siguiente manera:

$$T_e = P \Im m \{ \lambda_r^e \cdot (i_r^e)^* \} \quad (2.21)$$

$$T_e = \Im m \left\{ \lambda_r^e \left( \frac{1}{L_r} (\lambda_r^e - M_{er} i_e) \right)^* \right\} \quad (2.22)$$

$$T_e = P \frac{M_{er}}{L_r} (\lambda_{rx}^e i_{ey} - \lambda_{ry}^e i_{ex}) \quad (2.23)$$

donde:

$P \rightarrow \#$  de pares de polos de la MI hexafásica.

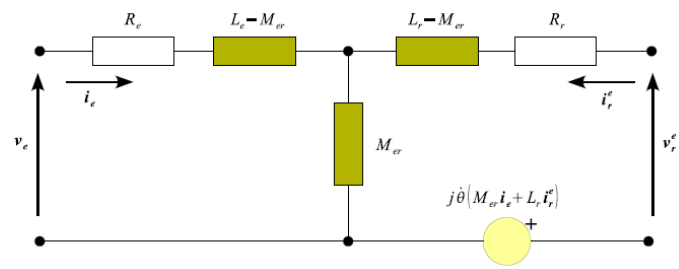


Figura 2.5: Modelo de la MI referido al estator [3]

# 3 Herramientas para la modelación

## 3.1. Elmer FEM

Elmer es una plataforma para la simulación de elementos finitos mediante la resolución de ecuaciones diferenciales parciales (EDP's). Esta herramienta se entrega con licencia pública general (GPL), es difundido por CSC – Finnish IT Center for Science, entidad que pertenece al ministerio de educación de Finlandia . La creación del programa es gracias a varias universidades de Finlandia, la industria y varios grupos de investigación [25].

El programa se puede descargar directamente desde la internet, mediante la plataforma `hppt://www.csc.fi/Elmer` y esta disponible para varios sistemas operativos tales como Linux, Windows, Unix y Mac [25]. Además, se puede encontrar una extensa documentación acerca del programa:

- Manual para la instalación
- Ejemplos
- Manual para el usuario
- Tutoriales
- Foro Elmer, etc.

## 3.2. Características principales

Elmer tiene una gran capacidad para solucionar modelos matemáticos de elementos finitos, en la siguiente lista se muestra unos ejemplos que se pueden desarrollar [25]:

- Transferencia de calor.
- Mecánica de fluidos.
- Transporte de especies.
- Elasticidad.
- Acústica.
- Electromagnetismo.
- Microfluidos.
- Mecánica cuántica.
- Problemas de superficie libre.

### 3.3. Estructura de Elmer

Elmer FEM es un conjunto de programas ejecutables y se pueden utilizar de forma independiente, en la Fig. 3.2 se puede observar la estructura de funcionamiento. Dentro del software Elmer, existe un grupo de programas para el desarrollo del proceso completo de la simulación [25, 4].

- **ElmerFront:** Este paquete permite definir el modelo numérico que vamos a utilizar y la ejecución de mismo se puede realizar mediante líneas de comando o mediante la interface de ElmerGUI [25].
- **ElmerGUI:** Es la interface gráfica de Elmer, la cual esta plataforma se ingresan elementos geométricos y elementos mallados en 2D y 3D importados desde otros programas, esto debido a que ElmerGUI no permite realizar la definición geométrica de los elementos. La interface gráfica es muy interactiva y simple al momento de utilizar, esta controla los ejecutables como ElmerSolver y ElmerPost, en la Fig. 3.1 se puede apreciar la ventana principal de ElmerGUI. [24, 25].

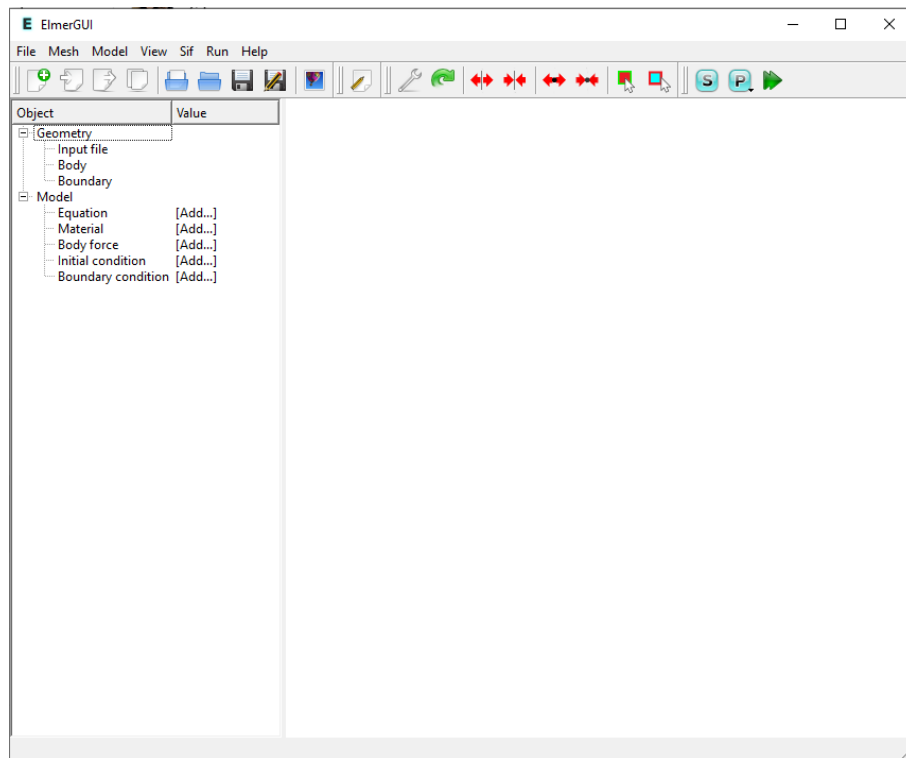


Figura 3.1: Interface gráfica de ElmerGUI.

- **ElmerSolver:** Es un programa ejecutable el cual cubre la mayor parte de trabajo que realiza Elmer, mediante este programa se resuelve todas las ecuaciones diferenciales parciales con líneas de comando [25].
- **ElmerPost:** Es la plataforma que entra en funcionamiento luego de ingresar los datos de la simulación. En esta pantalla se pueden visualizar todos los resultados obtenidos en ElmerGUI [25].
- **ElmerGrid:** Se utiliza para ingresar el mallado a las figuras a simular, se puede colocar mallas en 2D y 3D. Además mediante esta plataforma se puede

importar geometrías con malla como por ejemplo: gmsh, comsol, fidap, etc. [25].

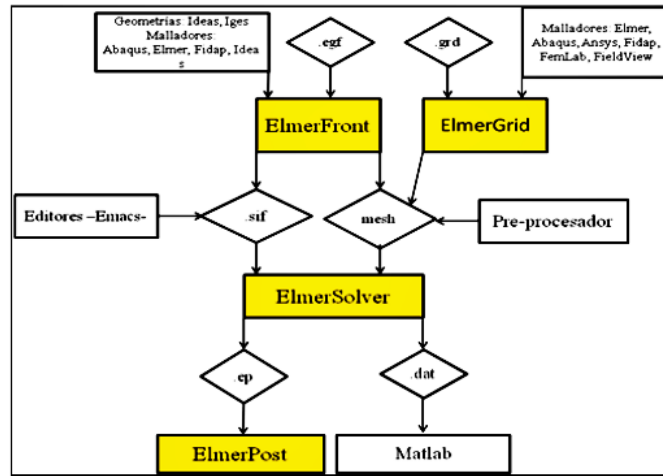
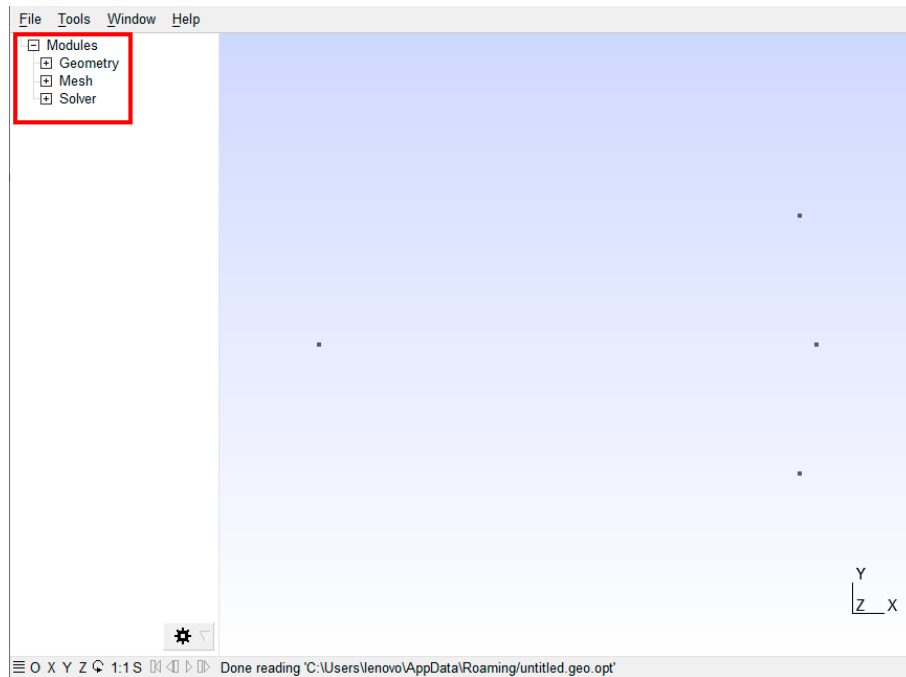


Figura 3.2: Estructura de Elmer FEM [4]

### 3.4. Gmsh

Es una herramienta CAD de código abierto para diseño que contiene una gran capacidad de generación de malla de elementos finitos en  $2D$  y  $3D$ . Su principal objetivo de diseño es el de proporcionar una herramienta de mallado rápida, liviana y fácil de utilizar con una entrada paramétrica y avanzadas capacidades de visualización, en la Fig. 3.3 se puede apreciar la interface gráfica Gmsh. La base del programa es cuatro módulos: geometría, malla, solucionador y post-procesamiento. La entrada a estos módulos es de forma interactiva utilizando la interface gráfica e ingresando código en archivos de texto ASCII en el propio lenguaje scripting de Gmsh [33, 6].





**Figura 3.3:** Plataforma Gmsh

### 3.4.1. Módulo de geometría

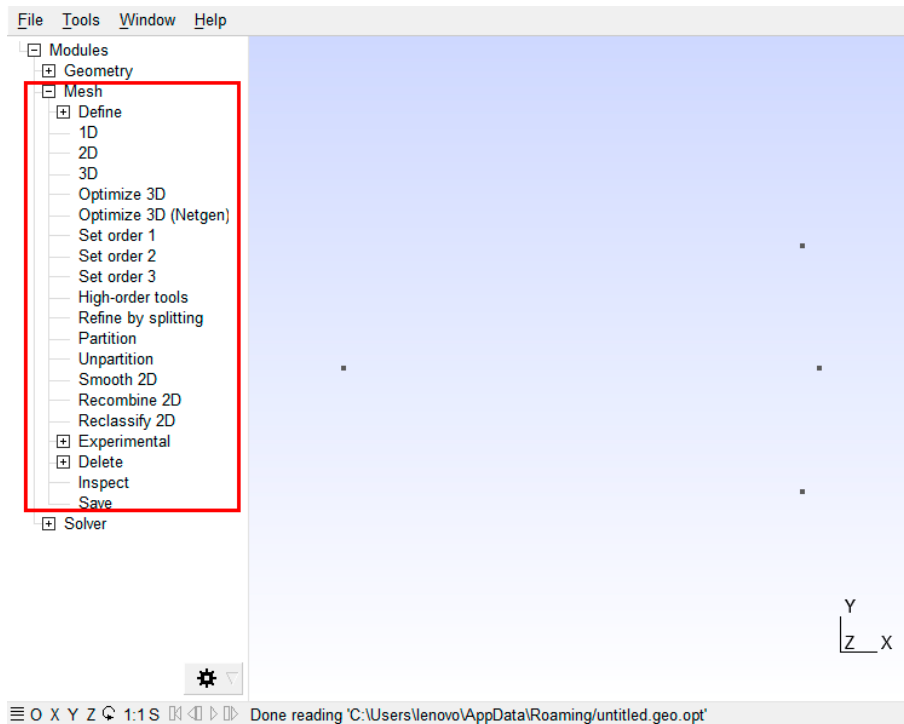
Se define cuatro tipos de entidades de modelo:

1. Modelo de puntos: Para 0 dimensiones.
2. Modelo de curvas: Para 1 dimensión (1D).
3. Modelo de superficies: Para 2 dimensiones (2D).
4. Modelo de volúmenes: Para 3 dimensiones (3D).

Las entidades de modelo son entidades topológicas, es decir que solo tratan con adyacencias en el modelo. Cualquier modelo puede construir su lista de adyacencias de cualquier dimensión utilizando operaciones locales [33].

### 3.4.2. Módulo de mallado

Gmsh contiene varios algoritmos de mallado con características específicas como 1D, 2D, y 3D que puede ser de lados curvos y rectos. La estructura de datos de malla permite manejar fácil y eficientemente la creación, modificación y destrucción de mallas de elementos finitos. Los datos de malla están formados por elementos puntos, líneas, triángulos, cuadrángulos, tetraedros, hexaedros, etc. Definidos por una lista ordenada de sus nodos [33].



**Figura 3.4:** Módulo de mallado Gmsh

### 3.4.3. Principales ventajas de Gmsh

- Crear geometrías rápidamente que sean simples o repetitivas, gracias a funciones definidas por el usuario, bucles, condiciones, etc.
- La configuración de la geometría, GMSH permite que todas las variables, comandos y argumentos se sometan a comandos de cálculos anteriores.
- La generación de las mallas de elementos finitos en 1D, 2D y 3D.
- Se puede dimensionar los elementos con exacta precisión, GMSH provee varias herramientas para monitorizar el tamaño de la malla final, a través de la intercalación de tamaños determinados en los diferentes puntos de la geometría

## 3.5. Paraview

Paraview es una herramienta para la muestra y el análisis de datos, la multiplataforma es de código abierto que permite el análisis y la visualización de conjuntos de datos considerablemente grandes. Es una licencia de código abierto (BSD licensed, commercial software friendly) [5].

Paraview cuenta con una comunidad activa de usuarios y desarrolladores en la cual existen constantes contribuciones tanto al código como al manual de usuario, que ayudan a mejorar la herramienta y la documentación [5].

### 3.5.1. Visualización del modelo en Paraview

La visualización es el proceso de convertir datos sin procesar en imágenes y representaciones para alcanzar un mejor conocimiento de los datos. Paraview utiliza VTK, el kit de herramientas de visualización y el procesamiento de datos [5].

El modelo VTK se basa en el patrón de flujo de datos, los mismos que fluyen a través del sistema siendo transformados en cada paso por módulos conocidos como algoritmo, que pueden ser operaciones comunes como recorte, corte o generando contornos a partir de los datos, o que podrían estar calculando cantidades derivadas. etc [5].

En la Fig. 3.5 se observa el modelo de visualización de Paraview, los objetos de proceso A, B y C ingresan y generan uno o más objetos de datos. Los objetos de datos representan y brindan acceso a los datos, los objetos de proceso operan sobre los datos. A,B y C son objetos fuente, filtro y asignado respectivamente [5].

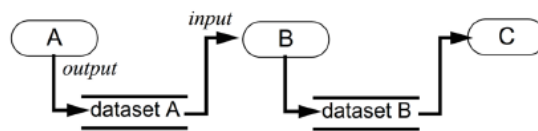


Figura 3.5: Visualización modelo en Paraview [5]

### 3.5.2. Interface gráfica de Paraview

La interfaz gráfica de Paraview contiene distintos componentes de la interface del usuario, cuando se ejecuta el software Paraview, se puede visualizar una pantalla como en la Fig. 3.6 la interfaz del usuario que se compone de menú, paneles acoplables, barras de herramientas y la ventana gráfica [5].

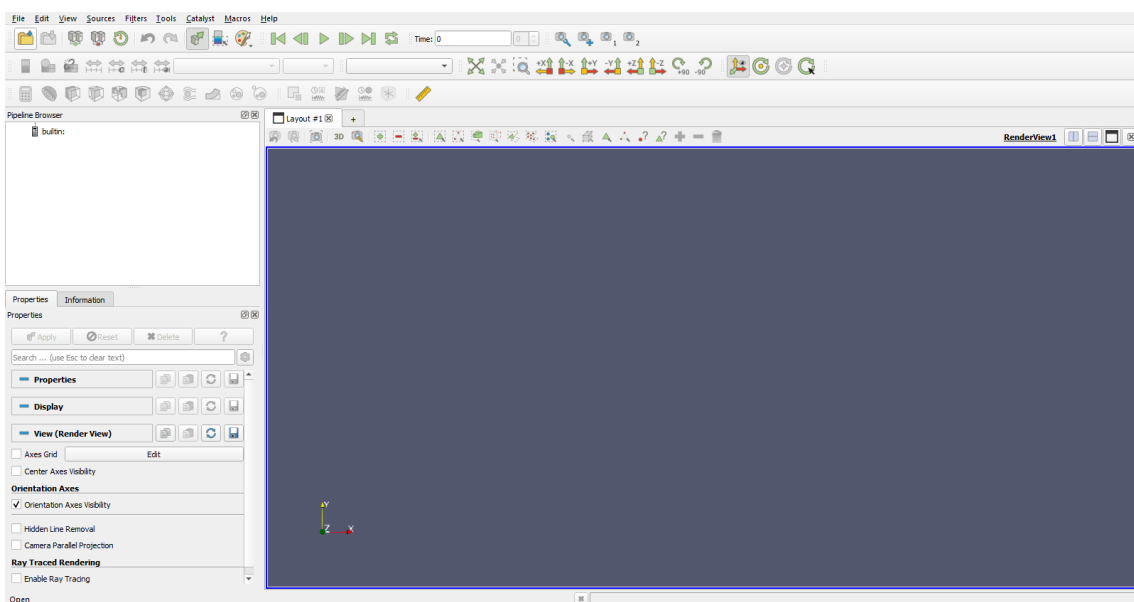


Figura 3.6: Interfaz gráfica de Paraview

## 4 Simulación Elmer FEM

### 4.1. Especificaciones de la Máquina Eléctrica de Inducción Hexafásica

En este proyecto se presenta la modelación de una máquina eléctrica de inducción hexafásica de  $5\text{ kW}$ , las dimensiones principales están especificadas en la Fig. 4.1

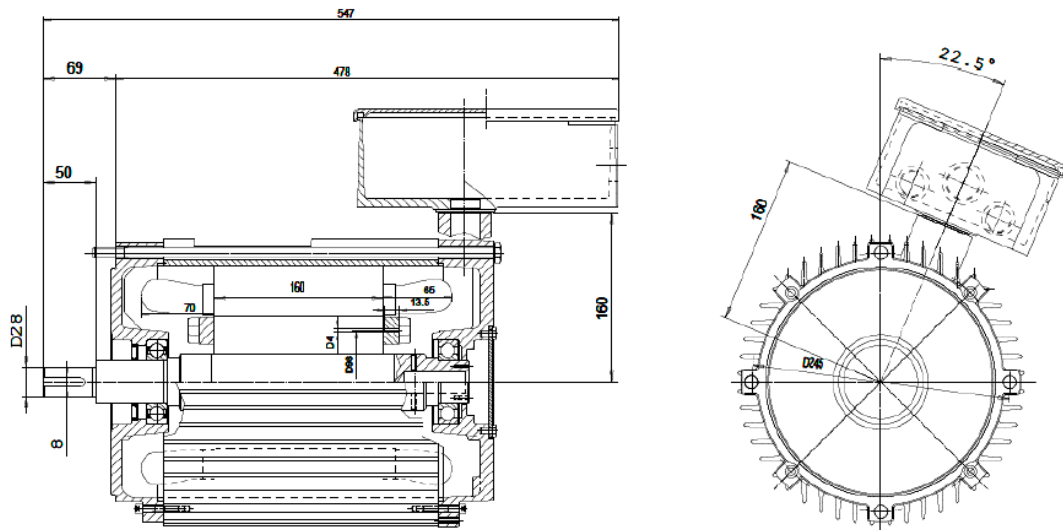


Figura 4.1: Dimensiones principales [6]

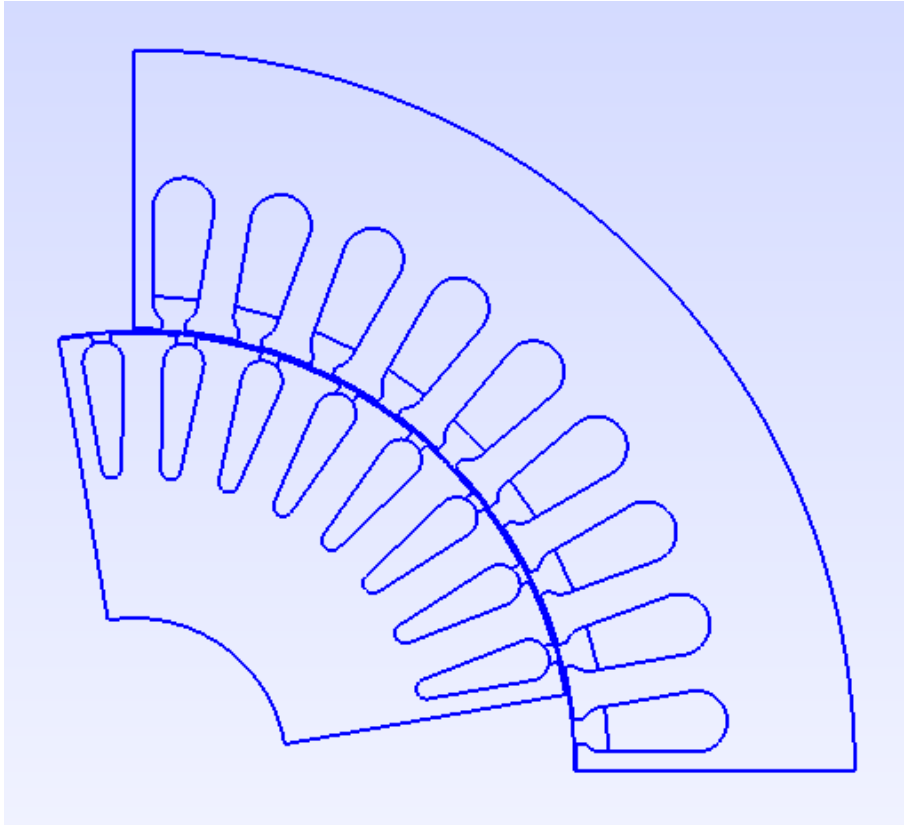
En la Tabla 4.1, se muestran las especificaciones principales de la máquina que será modelada mediante elementos finitos.

**Tabla 4.1:** Parámetros de la máquina eléctrica

<b>Parámetros</b>	<b>Valores</b>
Potencia	5 kW
Velocidad	1740 rpm - 60Hz
Voltaje	220 - 440V
Corriente	11 A
Corriente sin carga	6 A
Conexión	2Y
Número de fases	6
Número de pares de polos	2
Números de ranuras del estátor	48
Ranuras/Polos/Fases	4
Número de ranuras del rotor	40
Skew	8,5 grados
Torque	32 Nm

## 4.2. Geometría y mallado mediante GMSH

Para la creación de la geometría existen varios programas CAD, las cuales son de código abierto o licencia gratuita, por ejemplo: GMSH, Inventor, FreeCAD, Salome, etc. En este proyecto se utiliza la herramienta GMSH, la cual ofrece ventajas importantes para definir la geometría y realizar el mallado, tanto en 2D como en 3D. En esta plataforma se genera la geometría mediante secuencias de comando programadas en un archivo secuencial. También es posible construir la geometría mediante el GUI de GMSH, pero esto puede ser una limitante para realizar cambios posteriormente. [6] El modelo se basa en archivos GMSH contruidos originalmente por J. Gyselinck and R.V. Savariego y se puede visualizar en la Fig. 4.2 en el cual muestra 1 polo de la MI y están disponibles en [http://onelab.info/wiki/Electric\\_machines](http://onelab.info/wiki/Electric_machines).

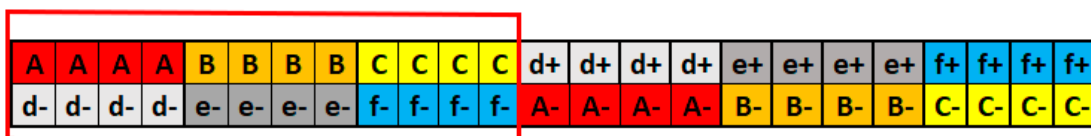


**Figura 4.2:** Geometría original de una máquina eléctrica de inducción [6]

La geometría se divide en archivos separados, el primero para el estátor y el segundo para el rotor. De esta forma, se generan dos mallas en el estado estacionario y transitorio. Sin embargo, Elmer FEM no permite cargar dos archivos de malla al mismo tiempo, por lo cual es necesario integrar ambas representaciones en un solo archivo utilizando la barra de comandos de windows descrito en la guía [24], donde indica la transformación e integración de los formatos de programa para la correcta utilización del software.

#### 4.2.1. Geometría del estátor

El diseño del devanado del estátor es periódico y contiene una distribución en 2 pares de polos. En la Fig. 4.3 se puede observar la sección periódica del diseño del devanado [6].

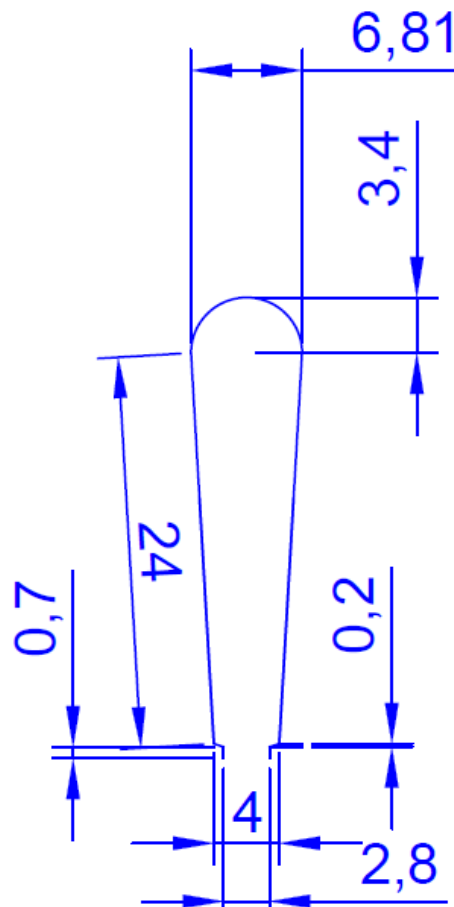


**Figura 4.3:** Sección periódica del diseño del devanado del estátor

Los principales parámetros del estátor se presentan en la Tabla 4.2 [6]. Y en la Fig. 4.4 se puede apreciar detalladamente las medidas dispuestas para cada ranura del estátor.

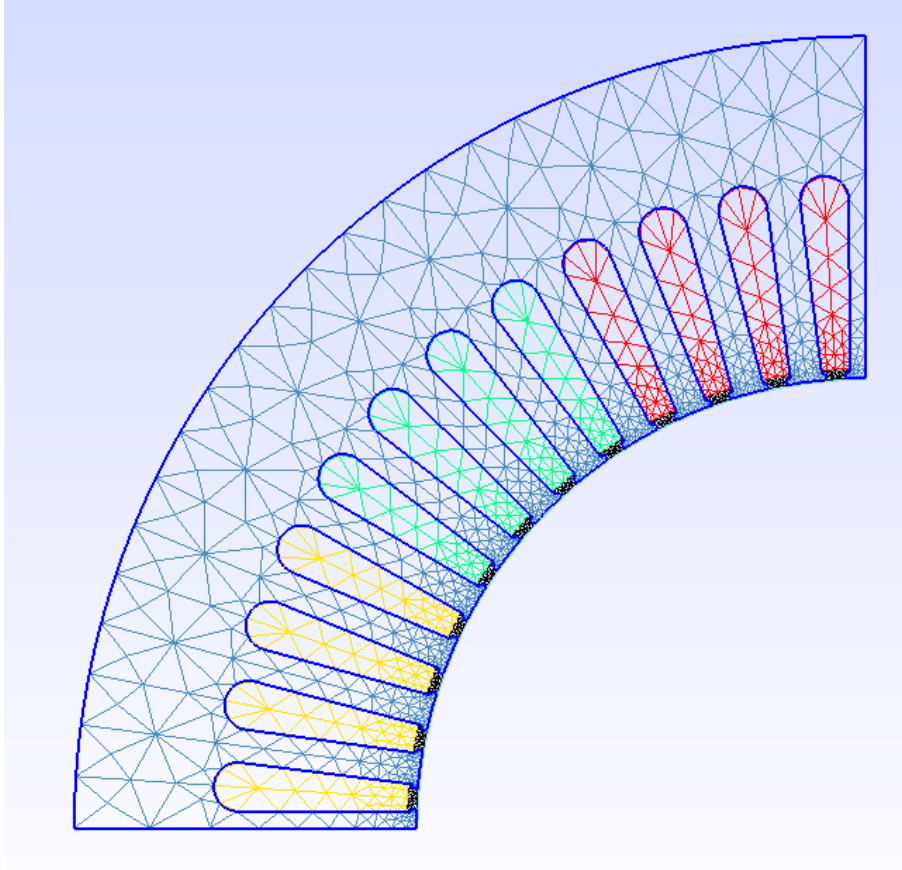
**Tabla 4.2:** Parámetros de la geometría del estátor.

<b>Parámetro</b>	<b>Descripción</b>	<b>Valor</b>
$Q_s$	Número de dientes del estátor	48
$N_{ss}$	Número de ranuras modeladas	96
$R_{sin}$	Radio del agujero del estátor	62.5
$R_{sout}$	Radio del agujero exterior	110
$R_g$	Radio de la mitad del espacio de aire	62.25
$b_5$	Ancho de la ranura inferior	6.8
$b_1$	Ancho de apertura de la ranura	2.8
$h_1$	Altura de las puntas de los dientes	0.7
$h_2$	Altura de las puntas levantadas	0.2
$b_4$	Ancho de la parte superior de la ranura	4
$h_5$	Altura de la sección de la ranura principal	24



**Figura 4.4:** Dimensiones de la ranura del estator. [6]

La creación de la geometría se fundamentó en el modelo de la máquina eléctrica de inducción trifásica [6], como se observa en la Fig. 4.5, para el caso de estudio, se modificó el modelo original transformándolo de trifásico a hexafásico. En la geometría de estátor se dividen las ranuras en dos partes iguales, como se puede apreciar en la Fig. 4.7 y a continuación se procede a realizar los cálculos del modelo matemático.

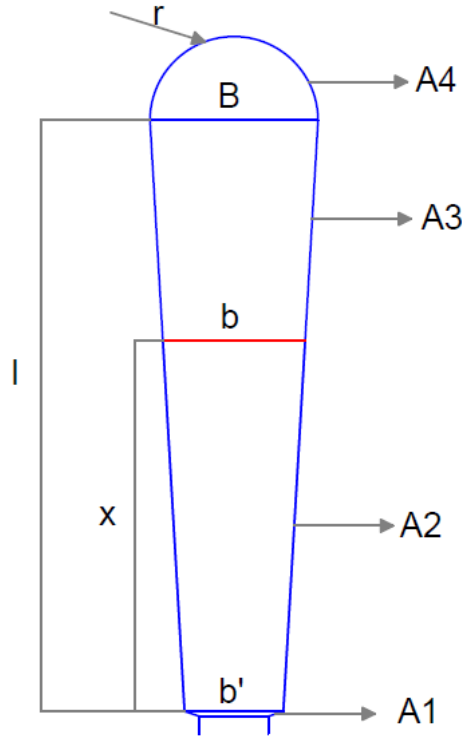


**Figura 4.5:** Geometría de una máquina eléctrica de inducción trifásica [6]

#### 4.2.1.1. Cálculo del centro de la ranura del estátor

En la Fig. 4.6 se presenta la geometría de la ranura del estátor y los principales datos para realizar el cálculo de centro de la ranura aplicando el método del trapecio.





**Figura 4.6:** Geometría de la ranura del estátor

Sistema de ecuaciones:

$$A_1 + A_2 = A_3 + A_4 \rightarrow \text{Condición} \quad (4.1)$$

$$A_2 - A_3 = A_4 - A_1 \quad (4.2)$$

$$A_2 + A_3 = A_C \quad (4.3)$$

Aplicando el método del trapecio, se determinan las ecuaciones de las áreas A2 y A3:

$$A_2 = \frac{(b + b') * x}{2} \quad (4.4)$$

$$A_3 = \frac{(B + b) * (l - x)}{2} \quad (4.5)$$

Reemplazando (4.4) y (4.5) en (4.1):

$$\frac{(b + b') * x}{2} - \frac{(B + b) * (l - x)}{2} = A_4 - A_1 \quad (4.6)$$

$$xb + xb - lB + xB - lb + xb' = 2A_4 - 2A_1 \quad (4.7)$$

$$2xb - lb = 2A_4 - 2A_1 - xb' + lB - Xb \quad (4.8)$$

$$b = \frac{2A_4 - 2A_1 - xb' + lB - xB}{2x - l} \quad (4.9)$$

Reemplazando (4.4) y (4.5) en (4.2):

$$\frac{(b + b') * x}{2} + \frac{(B + b') * (l - x)}{2} // \text{numberofrotorteeth} = A_c \quad (4.10)$$

$$xb + xb' + lB - xB + lb - xb = 2A_c \quad (4.11)$$

$$b = \frac{2A_c + xB - lB - xb'}{l} \quad (4.12)$$

Igualando las ecuaciones (4.9) y (4.12):

$$\frac{2A_4 - 2A_1 - xb' + lB - xB}{2x - l} = \frac{2A_c + xB - lB - xb'}{l} \quad (4.13)$$

$$(l)(2A_4 - 2A_1 - xb' + lB - xB) = (2x - l)(2A_c + xB - lB - xb') \quad (4.14)$$

Se resuelve el sistema de ecuaciones, igualando a cero, donde la única incógnita es x, el resto de variables son conocidos,

$$A_4 = 0.68[cm^2], A_1 = 18.15[cm^2], B = 6.8[cm], l = 24[cm], \quad (4.15)$$

$$b' = 4[cm], A_c = 129,6[cm^2], x = ? \quad (4.16)$$

Resolviendo la ecuación, se obtiene una ecuación de segundo orden, que se puede solucionar mediante la ecuación general

$$-5.6x^2 - 384x + 7059.36 = 0 \quad (4.17)$$

Formula general:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (4.18)$$

$$x' = \frac{384 + \sqrt{(-384)^2 - 4(-5.6)(7059.36)}}{2(-5.6)} = 83.64[cm] \quad (4.19)$$

$$x = \frac{384 - \sqrt{(-384)^2 - 4(-5.6)(7059.36)}}{2(-5.6)} = 15.07[cm] \quad (4.20)$$

Una vez determinado el valor de  $x=15.07[\text{cm}]$ , se obtiene el valor de la dimensión  $b$ ,

$$b = \frac{2A_c + xB - lB - xb'}{l} \quad (4.21)$$

$$b = \frac{2(129.6) + (15.07)(6.8) - (24)(6.8) - (15.07)(4)}{24} = 5.758[\text{cm}] \quad (4.22)$$

Se determinan las áreas  $A_2$  y  $A_3$ :

$$A_2 = \frac{(b + b') * x}{2} = \frac{(5.758 + 4) * 15.07}{2} = 73.526[\text{cm}^2] \quad (4.23)$$

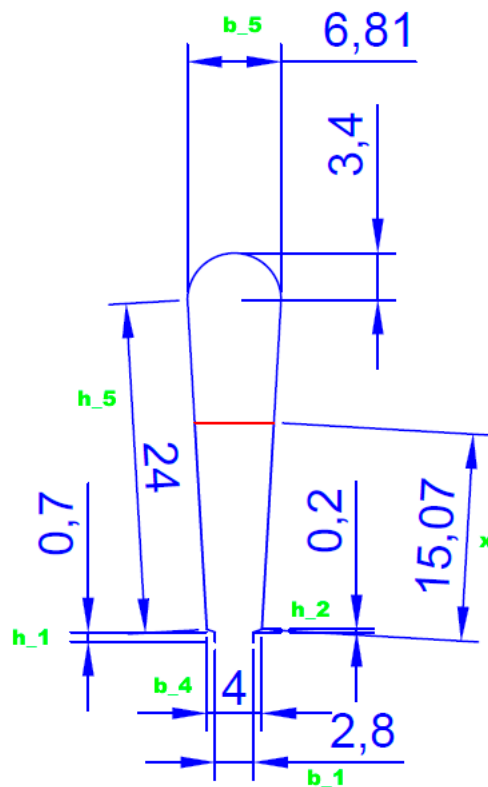
$$A_3 = \frac{(B + b) * (l - x)}{2} = \frac{(6.8 + 5.758) * (24 - 15.07)}{2} = 56.071[\text{cm}^2] \quad (4.24)$$

Condición inicial del sistema, para que cumpla el objetivo:

$$A_1 + A_2 = A_3 + A_4 \rightarrow \text{Condición} \quad (4.25)$$

$$0.68 + 73.526 = 56.071 + 18.15 \rightarrow \text{Condición} \quad (4.26)$$

$$74.206[\text{cm}^2] = 74.221[\text{cm}^2] \rightarrow \text{Cumple... Condición} \quad (4.27)$$



**Figura 4.7:** Geometría de la ranura del estátor

#### 4.2.1.2. Creación de la geometría del estátor mediante GMSH

Con el conocimiento de las dimensiones de la máquina eléctrica se procede a crear la geometría mediante la plataforma de GMSH, primeramente se ingresan los datos para la construcción de la geometría del estátor mencionado anteriormente:

```
Geometry.AutoCoherence = 0;
//ESTATOR
Qs = 48; // |
N_ss = 12;
//PARÁMETROS DEL ESTATOR
R_sin = 62.5; //
R_sout = 110; //
R_g = 62.25; //
//DIMENSIONES DEL SLOT
b_5 = 6.8;
b_1 = 2.8;
h_1 = 0.7;
h_2 = 0.2;
b_4 = 4;
h_5 = 24;

//CENTRO DE LAS AREAS
xcentro=15.07;
```

**Figura 4.8:** Ingreso de datos

GMSH elimina todas las entidades de la geometría con las mismas coordenadas automáticamente, para este caso es necesaria esta opción desactivada porque la geometría se construye por propagación [6]. Las alturas de las ranuras se definen haciendo referencia al eje de la ranura, por lo tanto, hay que tener en cuenta la curvatura de la superficie interna del estátor y un desplazamiento  $s$  debe calcularse como:

$$s = R_{sin} - \sqrt{(R_{sin})^2 - \left(\frac{b_1}{2}\right)^2} \quad (4.28)$$

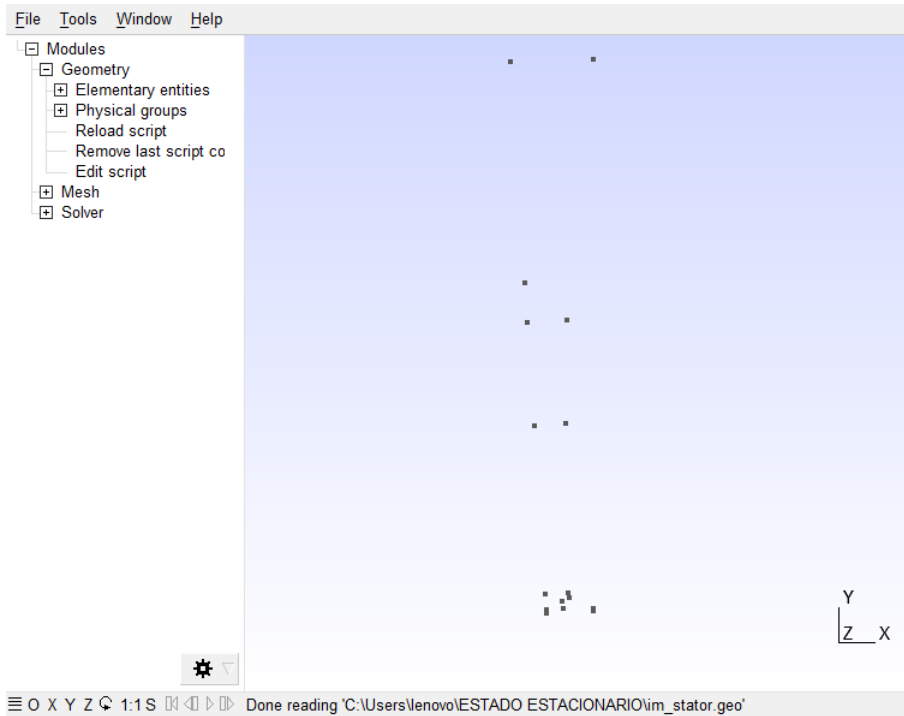
Ahora, se ingresan los puntos que describen la geometría de la ranura del estátor como se observa en la Fig. 4.10, los cuales son 17 puntos que describen la geometría del estátor y  $dP$  es una variable que permite realizar un seguimiento de la construcción de la geometría de todos los demás sectores modelados en un bucle *for*. En la Fig. 4.10, se visualiza los principales datos.

```

dP=newp;
Point(dP+0) = {0,0,0,m_coarse};
Point(dP+1) = {b_1/2, R_sin-s, 0, 2*m_gap};
Point(dP+2) = {b_1/2, R_sin-s+h_1, 0, m_sl_bot};
Point(dP+3) = {0, R_sin-s+h_1+h_2+h_5, 0, m_sl_top};
Point(dP+4) = {b_4/2, R_sin-s+h_1+h_2, 0, m_sl_bot};
Point(dP+5) = {b_5/2, R_sin-s+h_1+h_2+h_5, 0, m_sl_bot};
Point(dP+6) = {b_4/2, R_sin-s+h_1+h_2+0.5, 0, m_sl_bot/2};
Point(dP+7) = {0, R_sin-s+h_1+h_2+0.5, 0, m_sl_bot};
Point(dP+8) = {0, R_sin-s+h_1+h_2+h_5+b_5/2, 0, m_sl_bot};
Point(dP+9) = {R_sout*Sin(Pi/Qs), R_sout*Cos(Pi/Qs), 0, m_s_out};
Point(dP+10) = {0, R_sout, 0, m_s_out};
Point(dP+11) = {R_sin*Sin(Pi/Qs), R_sin*Cos(Pi/Qs), 0, 2.5*m_gap};
Point(dP+12) = {R_g*Sin(Pi/Qs), R_g*Cos(Pi/Qs), 0, 2.4*m_gap};
Point(dP+13) = {0, R_sin, 0, 1.5*m_gap};
Point(dP+14) = {0, R_g, 0, 1.5*m_gap};
// punto medio de ranura IZQUIERDO
Point(dP+15) = {0, R_sin-s+h_1+h_2+xcentro, 0,m_sl_top};|
// punto medio de ranura DERECHO
Point(dP+16) = {((b_5/2)+(b_4/2))/2, R_sin-s+h_1+h_2+xcentro, 0, m_sl_top};

```

**Figura 4.9:** Ingreso de puntos de la geometría del estátor



**Figura 4.10:** Puntos de la geometría del devanado del estátor.

En la Fig. 4.9, la tercera coordenada es 0 ya que se está construyendo un modelo 2D y el cuarto parámetro es de la densidad de la malla. A continuación, en la Fig. 4.11, se ingresan los valores de densidad de la malla, para producir una malla precisa. Como también la malla debe ser bastante densa en los lugares que tienen espacio de aire, GMSH tiene varias opciones para controlar esta densidad [33, 24], y uno de ellos es el de agregar puntos adicionales a la geometría.

```

//DENSIDAD DE LA MALLA
m_coarse = 20; // GRUESA
m_normal = 12; // NORMAL
m_gap = 0.55; // BRECHA
m_sl_top = 7; // CIMA
m_sl_bot = 2; //
m_s_out = 12; // FUERA

```

**Figura 4.11:** Densidad de la malla del estátor

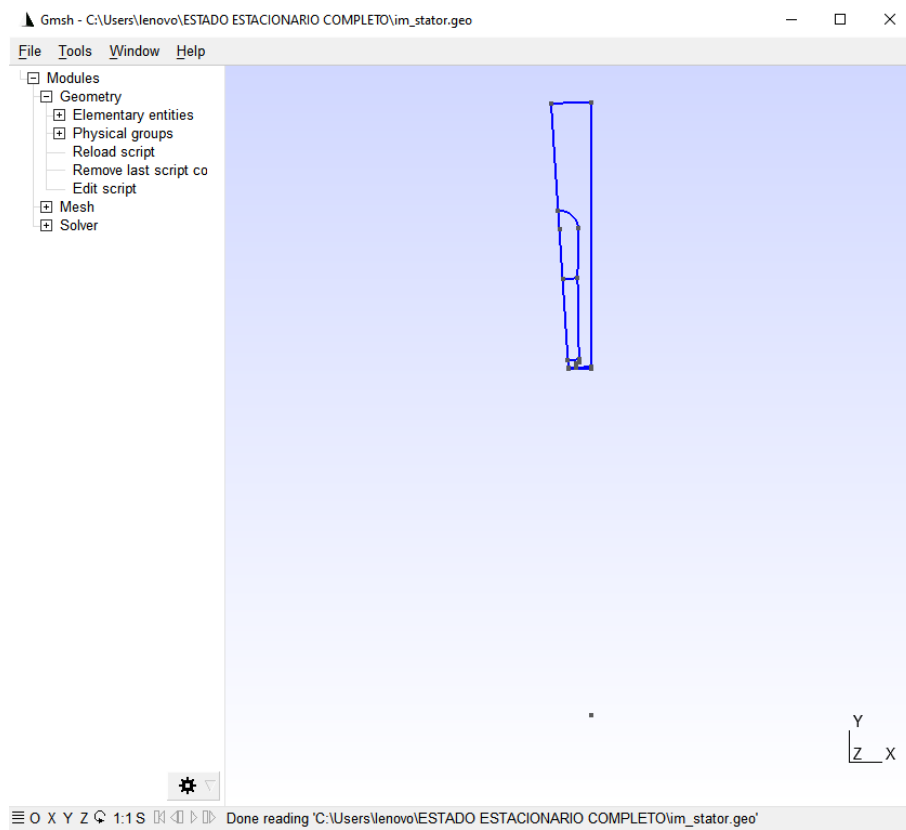
A continuación, en la Fig. 4.12, se generan líneas y curvas para unir los puntos y construir la geometría. En la Fig. 4.13 se puede visualizar el resultado obtenido.

```

// UNION DE LINEAS Y CURVAS
dR=new1-1;
Line(dR+1) = {dP+1,dP+2};
Line(dR+2) = {dP+4,dP+6};
Line(dR+3) = {dP+6,dP+16};
Line(dR+4) = {dP+6,dP+7};
Line(dR+5) = {dP+2,dP+4};
Circle(dR+6) = {dP+5,dP+3,dP+8};
Circle(dR+7) = {dP+12,dP+0,dP+14};
Circle(dR+8) = {dP+11,dP+0,dP+1};
Circle(dR+9) = {dP+1,dP+0,dP+13};
Circle(dR+10) = {dP+9,dP+0,dP+10};
Line(dR+11) = {dP+13,dP+7};
Line(dR+12) = {dP+7,dP+15};
Line(dR+13) = {dP+8,dP+10};
Line(dR+14) = {dP+11,dP+9};
Line(dR+15) = {dP+12,dP+11};
Line(dR+16) = {dP+14,dP+13};
Line(dR+17) = {dP+16,dP+15};
Line(dR+18) = {dP+16,dP+5};
Line(dR+19) = {dP+15,dP+3};
Line(dR+20) = {dP+3,dP+8};

```

**Figura 4.12:** Líneas y curvas de la geometría del estátor



**Figura 4.13:** Unión de líneas y puntos del estátor

En la Fig. 4.14, se realiza toda la geometría modelada, anidando las líneas definidas y creando dos bucle *for*, unos para las partes de la simetría de la ranura y el otro para todas las ranuras restantes. Todos los puntos deben girarse a la enésima posición de la ranura formado toda la geometría del estátor como muestra en la Fig. 4.15

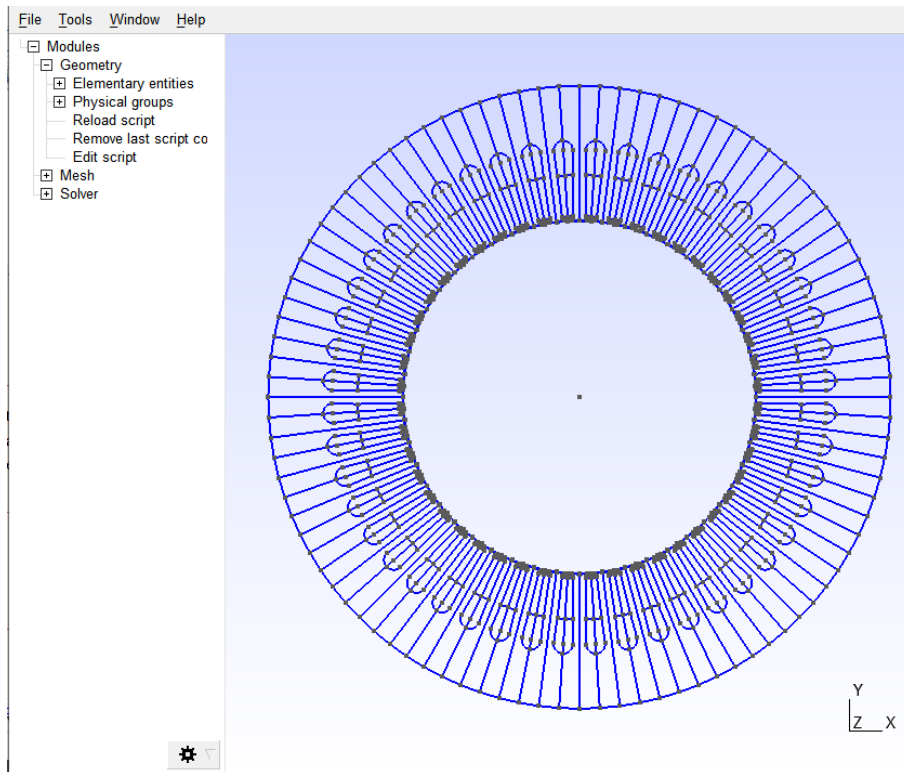
```

For t In {dP+0:dP+16}
  Rotate {{0,0,1},{0,0,0}, 2*Pi*i/Qs+2*Pi/Qs/2} {Point{t};}
EndFor

If (half==1) //second half
  For t In {dP+0:dP+16}
    Symmetry {Cos(2*Pi*i/Qs+2*Pi/Qs/2),Sin(2*Pi*i/Qs+2*Pi/Qs/2),0,0} { Point{t}; }
  EndFor
EndIf

```

**Figura 4.14:** Bucle para formar toda la geometría



**Figura 4.15:** Geometría completa del estátor

Ahora, para agilizar en tiempo de programación mediante la plataforma Elmer FEM, es necesario identificar las líneas que constituyen las entidades físicas. Por lo tanto, en la Fig. 4.16, se ingresa el código y se identifican los límites de las regiones, límites periódicos y límites deslizantes. Luego, las superficies planas se definen para construir regiones físicas.

```
// Llenando las listas de límites
EstatorExterior_[] += dR+10;
LimiteEstator_[] += {dR+10,dR+6,dR+18,dR+17,dR+3,dR+2,dR+5,dR+1,dR+8};
CorredizoEstator_[] += {dR+7};
Separar_[] += {dR+17};
If (Qs != N_ss)
  If (i==0 && half==0)
    EstatorPeriod_Derecha_[] = {dR+14,dR+15};
  EndIf
  If (i == N_ss-1 && half==1)
    EstatorPeriod_Izquierda_[] = {dR+14,dR+15};
  EndIf
EndIf
rev = (half ? -1 : 1);
Line Loop(new11) = {dR+19,dR+20,-dR-6,-dR-18,dR+17};
dH = news; Plane Surface(news) = -rev*{new11-1};
EstatorConductor1_[] += dH;

Line Loop(new11) = {dR+12,-dR-17,-dR-3,dR+4};
dH = news; Plane Surface(news) = -rev*{new11-1};
EstatorConductor2_[] += dH;

Line Loop(new11) = {dR+1,dR+5,dR+2,dR+3,dR+18,dR+6,dR+13,-dR-10,-dR-14,dR+8};
dH = news; Plane Surface(news) = -rev*{new11-1};
AceroEstator_[] += dH;
Line Loop(new11) = {dR+11,-dR-4,-dR-2,-dR-5,-dR-1,dR+9};
dH = news; Plane Surface(news) = -rev*{new11-1};
Wedge_Estator_[] += dH;
Line Loop(new11) = {-dR-16,-dR-7,dR+15,dR+8,dR+9};
dH = news; Plane Surface(news) = rev*{new11-1};
AireEstator_[] += dH;
EndFor
EndFor
```

**Figura 4.16:** Entidades físicas de la geometría del estátor

El siguiente paso consiste en asignar las superficies físicas o regiones a cada una



de las superficies planas. En la Fig. 4.17, se colocan los nombres a cada una de las regiones.

```

For f In {0:5}
  Con[]={};
  For i In {0:N_ss/qq-1}
    If (Fmod(i,6) == f)
      For j In {0:qq-1}
        Con[] += EstatorConductor1_[{2*i*qq+2*j, 2*i*qq+2*j+1}];
      EndFor
    EndIf
  EndFor
//colorear la fase y asignarla a una región física
If (#Con[] > 0)
  If (f == 0)
    Color Red {Surface{Con[]}};
    Physical Surface("U_plus") = {Con[]};
  EndIf
  If (f == 1)
    Color Blue {Surface{Con[]}};
    Physical Surface("W_minus") = {Con[]};
  EndIf
  If (f == 2)
    Color DarkGreen {Surface{Con[]}};
    Physical Surface("V_plus") = {Con[]};
  EndIf
  If (f == 3)
    Color Pink {Surface{Con[]}};
    Physical Surface("U_minus") = {Con[]};
  EndIf
  If (f == 4)
    Color Yellow {Surface{Con[]}};
    Physical Surface("W_plus") = {Con[]};
  EndIf
  If (f == 5)
    Color Cyan {Surface{Con[]}};
    Physical Surface("V_minus") = {Con[]};
  EndIf
EndIf
EndFor

```

**Figura 4.17:** Nombres de las regiones del estátor

A continuación, en la Fig. 4.18, se definen todas las líneas y superficies físicas restantes para el estátor:

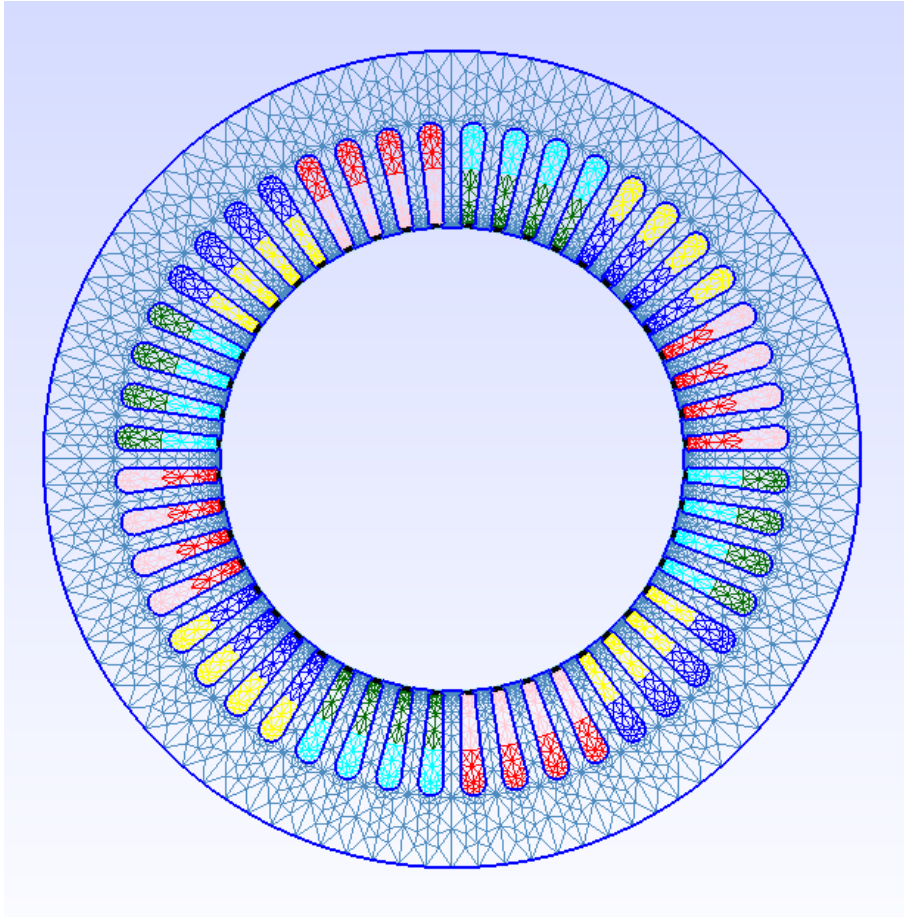
```

Physical Surface("AceroEstator") = {StatorIron_[]};
Physical Surface("Wedges_-Estator") = {StatorWedge_[]};
Physical Surface("AireEstator") = {StatorAingapLayer_[]};
Color SteelBlue {Surface{AceroEstator_[]}};
Color Black {Surface{Wedge_Estator_[]}};
Color SkyBlue {Surface{AireEstatorLayer_[]}};
Physical Line("EstatorExterior") = {OuterStator_[]};
Physical Line("EstatorDerecha") = {StatorPeriod_Right_[]};
Physical Line("EstatorIzquierda") = {StatorPeriod_Left_[]};
Physical Line("EstatorCorredizo") = {Sliding_[]};
Physical Line("Separar") = {Separar_[]};

```

**Figura 4.18:** Regiones del estátor

El resultado de la malla del estátor se muestra en la Fig. 4.19, esta malla 2D contiene 7585 elementos triangulares y el número total de elementos es de 18721.



**Figura 4.19:** Malla resultante de la geometría del estátor

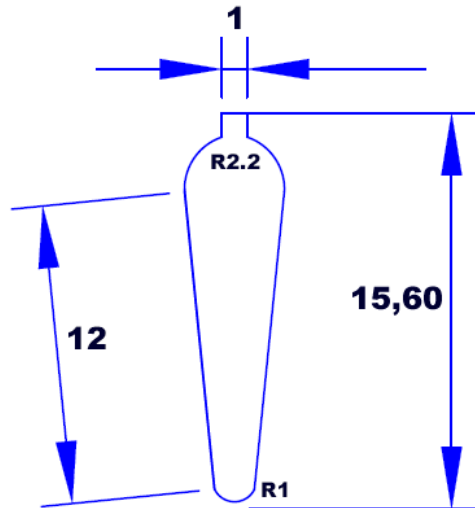
Por último, en la interface de usuario de GMSH, es necesario exportar el documento en formato *geo* al formato *msh*.

#### **4.2.2. Geometría del rotor**

En el cuadro 4.3, se presentan los principales parámetros para la creación de la geometría del rotor y en la Fig. 4.20 se muestra la gráfica del modelo de la ranura del rotor que es la misma notación que la ranura del estátor.

**Cuadro 4.3:** Parámetros de la geometría del rotor [6]

<b>Parámetro</b>	<b>Descripción</b>	<b>Valor</b>
Q <sub>r</sub>	Número de dientes del rotor	40
N <sub>rs</sub>	Número de ranuras del rotor modeladas	40
R <sub>rin</sub>	Radio interior del rotor	28
R <sub>rout</sub>	Radio exterior del rotor	62
R <sub>g</sub>	Radio de la mitad del espacio de aire	62.25
b <sub>5</sub>	Ancho de la ranura inferior	2
b <sub>1</sub>	Ancho de apertura de la ranura	1
h <sub>1</sub>	Altura de las puntas de los dientes	0.4
h <sub>5</sub>	Altura de la sección de la ranura principal	12
b <sub>4</sub>	Ancho de la parte superior de la ranura	4.4



**Figura 4.20:** Dimensiones de la ranura del rotor

#### 4.2.2.1. Creación de la geometría del rotor mediante GMSH

Mediante las dimensiones anteriormente presentadas en el cuadro 4.3, se procede a crear la geometría del rotor mediante la plataforma GMSH. De igual forma como se creó en la sección anterior la geometría del estátor. En la Fig. 4.21, se ingresan los datos principales de la geometría del rotor.

```

Qr = 40;
N_rs = 10;

R_rin = 27;
R_rout = 62;
R_g = 62.25;

b_1 = 1;
h_1 = 0.4;
b_4 = 4.4;
b_5 = 2;
h_5 = 12;

```

**Figura 4.21:** Ingreso de datos del rotor

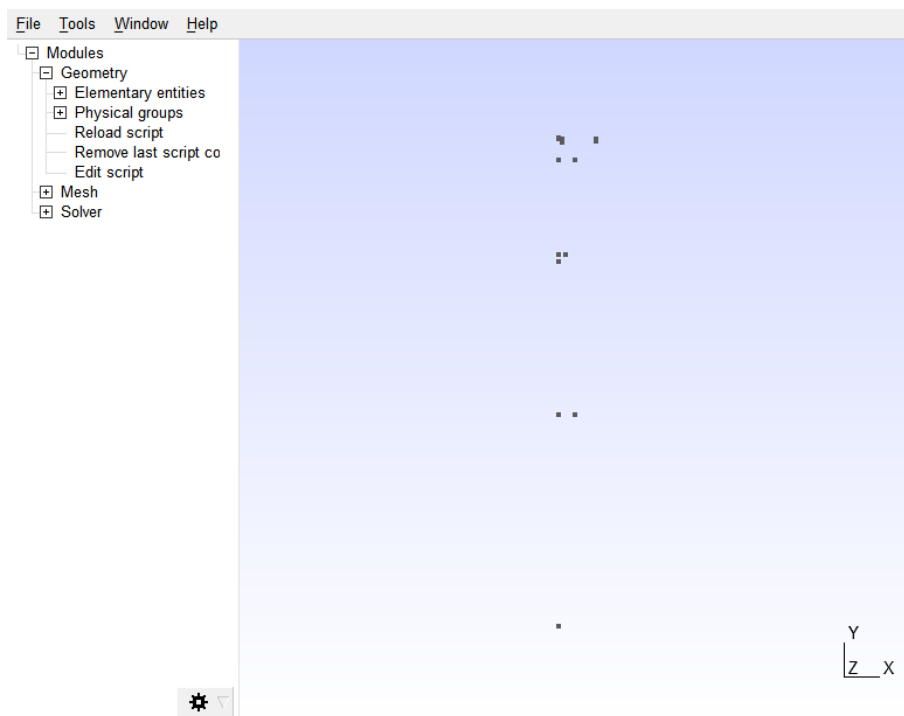
A continuación en la Fig. 4.22 se definen los puntos de la mitad la ranura del rotor al igual que lo hicimos con la ranura del estátor. En la Fig. 4.23 podemos apreciar todos los puntos generados.

```

dP=newp;
Point(dP+0) = {0,0,0,m_coarse};
Point(dP+1) = {b_1/2, R_rout-s, 0, 1.4*m_gap};
Point(dP+2) = {b_1/2, R_rout-h_1-s2, 0, m_sl_top};
Point(dP+3) = {0, R_rout-h_1-b_4/2-h_5, 0, m_sl_bot};
Point(dP+4) = {b_4/2, R_rout-h_1-b_4/2, 0, 2*m_sl_top};
Point(dP+5) = {b_5/2, R_rout-h_1-b_4/2-h_5, 0, m_sl_bot};
Point(dP+6) = {0, R_rout-h_1-b_4/2, 0, m_sl_top};
Point(dP+7) = {0, R_rout-h_1-b_4/2-h_5-b_5/2, 0, m_sl_bot};
Point(dP+8) = {R_rin*Sin(Pi/Qr), R_rin*Cos(Pi/Qr), 0, m_r_in};
Point(dP+9) = {0, R_rin, 0, m_r_in};
Point(dP+10) = {R_rout*Sin(Pi/Qr), R_rout*Cos(Pi/Qr), 0, 2.5*m_gap};
Point(dP+11) = {R_g*Sin(Pi/Qr), R_g*Cos(Pi/Qr), 0, 2.8*m_gap};
Point(dP+12) = {0, R_rout, 0, 1.5*m_gap};
Point(dP+13) = {0, R_g, 0, 1.4*m_gap};

```

**Figura 4.22:** Puntos de la ranura del rotor



**Figura 4.23:** Puntos de la ranura del rotor

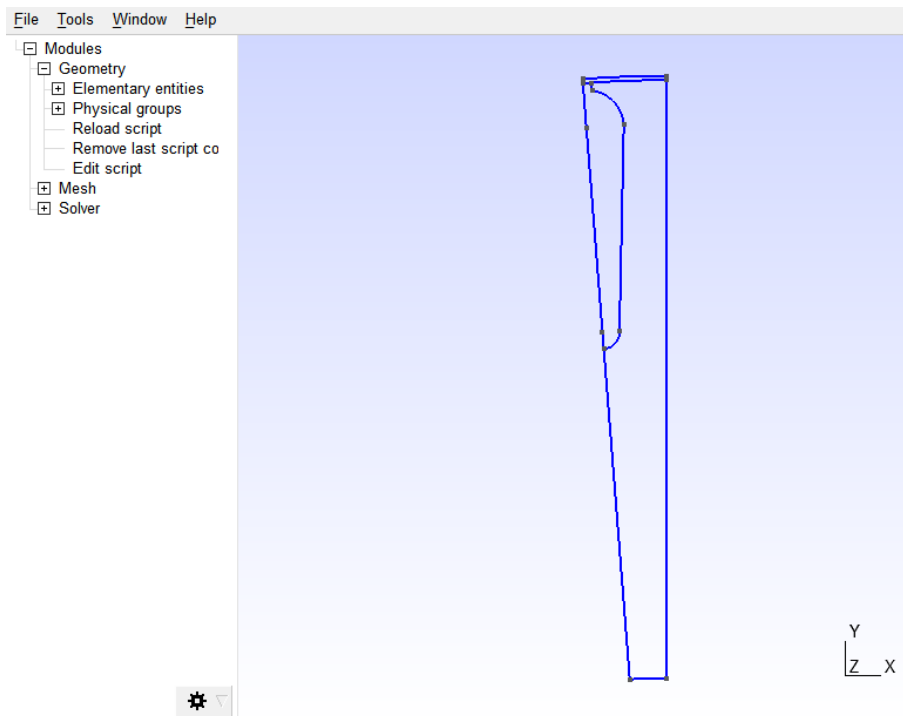
Ahora, en la Fig. 4.24, se generan líneas y curvas para poder conectar los puntos de la ranura. En la Fig. 4.25, se puede observar en forma gráfica como se forma la ranura del rotor.

```

dR=newl-1;
Line(dR+1) = {dP+1,dP+2};
  Circle(dR+2) = {dP+2,dP+6,dP+4};
Line(dR+3) = {dP+4,dP+5};
Circle(dR+4) = {dP+5,dP+3,dP+7};
Circle(dR+5) = {dP+11,dP+0,dP+13};
Circle(dR+6) = {dP+10,dP+0,dP+1};
Circle(dR+7) = {dP+1,dP+0,dP+12};
Circle(dR+8) = {dP+8,dP+0,dP+9};
Line(dR+9) = {dP+12,dP+7};
Line(dR+10) = {dP+7,dP+9};
Line(dR+11) = {dP+10,dP+8};
Line(dR+12) = {dP+11,dP+10};
Line(dR+13) = {dP+13,dP+12};

```

**Figura 4.24:** Líneas y curvas de la geometría del rotor



**Figura 4.25:** Unión de líneas y puntos del rotor

Luego, nuevamente se crea dos bucles *for* para crear la geometría completa del rotor. En la Fig. 4.26, ingresando el código para la rotación de los puntos definidos que se deben implementar a cada iteración y en la Fig. 4.27, se observa como se crea la geometría completa del rotor.

```

For t In {dP+0:dP+13}
  Rotate {{0,0,1},{0,0,0}, 2*Pi*i/Qr+2*Pi/Qr/2} {Point{t};}
EndFor
If (half==1)
  For t In {dP+0:dP+13}
    Symmetry {Cos(2*Pi*i/Qr+2*Pi/Qr/2),Sin(2*Pi*i/Qr+2*Pi/Qr/2),0,0} { Point{t}; }
  EndFor
EndIf

```

Figura 4.26: Rotación de puntos definidos del rotor

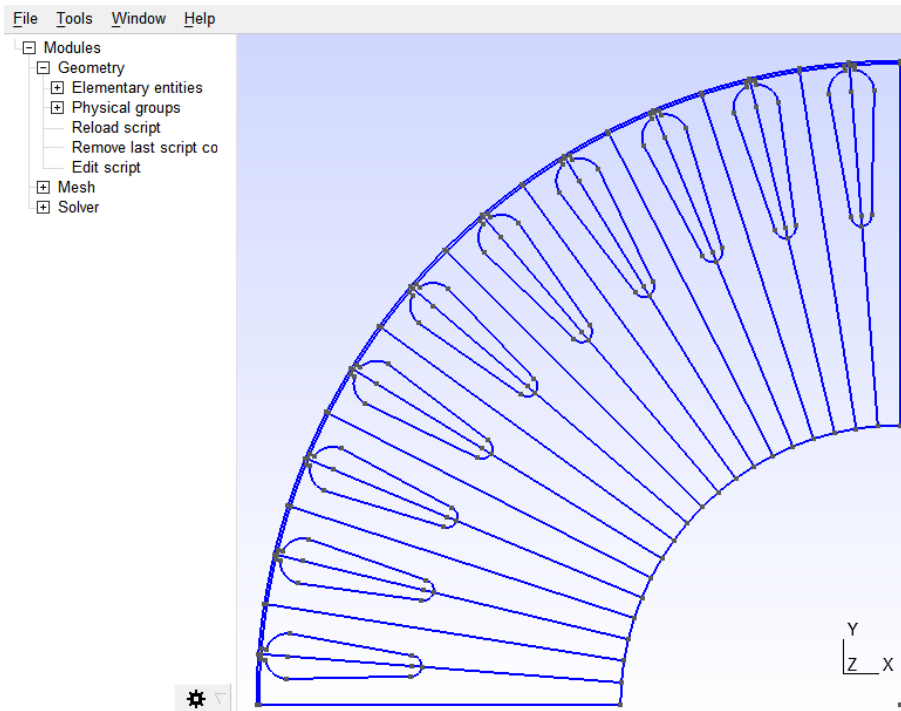


Figura 4.27: Geometría de un polo del rotor

A continuación, en la Fig. 4.28, se ingresan todas las entidades físicas que constituye la geometría del rotor y en la Fig. 4.29, se genera la malla para la geometría del rotor.

```

Color Yellow {Surface{RotorConductor_[]};}
Physical Surface("Cuerpo1") = {RotorConductor_{0,1}};
Physical Surface("Cuerpo2") = {RotorConductor_{2,3}};
Physical Surface("Cuerpo3") = {RotorConductor_{4,5}};
Physical Surface("Cuerpo4") = {RotorConductor_{6,7}};
Physical Surface("Cuerpo5") = {RotorConductor_{8,9}};
Physical Surface("Cuerpo6") = {RotorConductor_{10,11}};
Physical Surface("Cuerpo7") = {RotorConductor_{12,13}};
Physical Surface("Cuerpo8") = {RotorConductor_{14,15}};
Physical Surface("Cuerpo9") = {RotorConductor_{16,17}};
Physical Surface("Cuerpo10") = {RotorConductor_{18,19}};

Physical Surface("AceroRotor") = {AceroRotor_[]};
Physical Surface("AireRotor") = {AireRotorLayer_[]};

Color SteelBlue {Surface{AceroRotor_[]};}
Color SkyBlue {Surface{AireRotorLayer_[]};}

Physical Line("RotorInterno") = {RotorInterno_[]};
Physical Line("RotorDerecho") = {RotorPeriod_Derecho_[]};
Physical Line("RotorIzquierdo") = {RotorPeriod_Izquierdo_[]};
Physical Line("Corredizo_Rotor") = {CorredizoR_[]};

Coherence;

show_rotor[] = CombinedBoundary{Surface{AceroRotor_[]};};
Hide{ Point{Point '*'};};
Hide{ Line{Line '*'};};
Show{ Line{show_rotor[]};};

Mesh 2;

```

**Figura 4.28:** Entidades físicas de la geometría del rotor

```

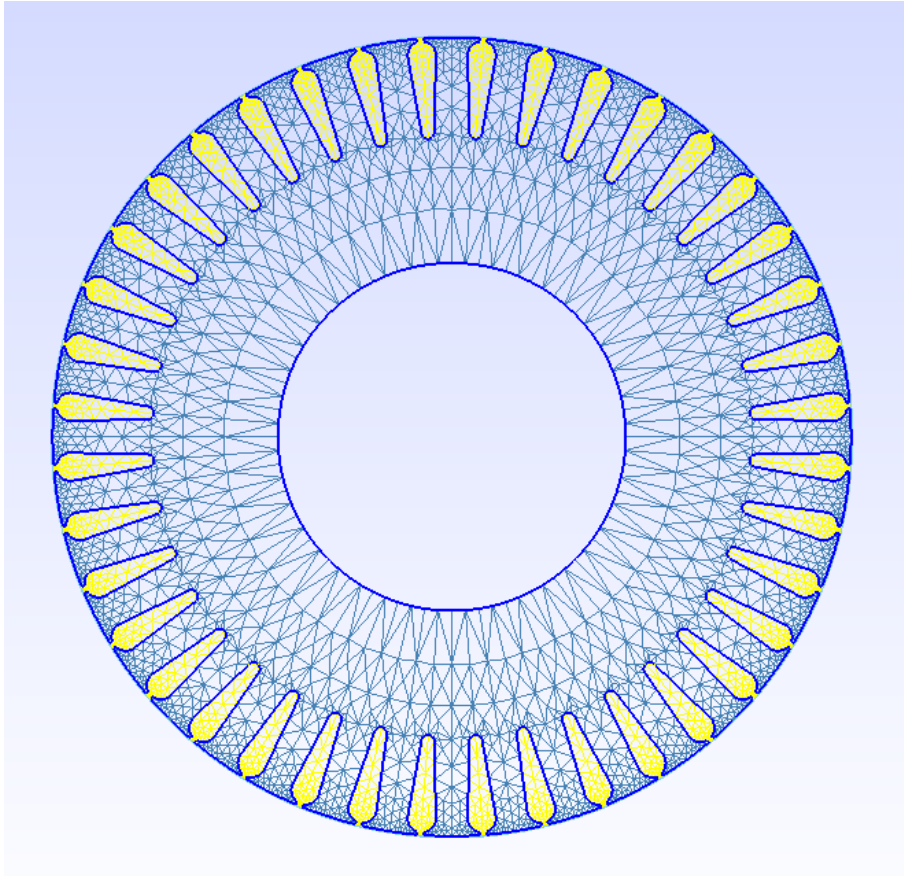
show_rotor[] = CombinedBoundary{Surface{AceroRotor_[]};};
Hide{ Point{Point '*'};};
Hide{ Line{Line '*'};};
Show{ Line{show_rotor[]};};

Mesh 2;

```

**Figura 4.29:** Generación de la malla del rotor

En la Fig. 4.30, se presenta el resultado de la generación de la malla del rotor, que contiene: 2220 elementos triangulares y el número total de elementos es de 3094.



**Figura 4.30:** Malla resultante de la geometría del rotor

Por último, es necesario exportar el documento de formato *.geo* a un formato *.msh*.

### 4.2.3. Importación a Elmer FEM

En este segmento se realiza la exportación de la malla generada en formato *.msh* de GMSH a un formato de Elmer. Esto se puede obtener mediante líneas de comando ElmerGrid [24]. Este admite varios formatos de malla y se exportarán todas las regiones físicas nombradas en GMSH. Mediante la herramienta Símbolo del sistema, en la Fig. 4.31, se ingresa el código para la importación del archivo a Elmer FEM.

```
ElmerGrid 14 2 Estator.msh -2d -autoclean -names  
ElmerGrid 14 2 Rotor|.msh -2d -autoclean -names
```

**Figura 4.31:** Comando de importación

donde;

- 14 representa el formato de malla de entrada.
- 2 representa el formado de salida de la malla.
- 2D ignora la tercera coordenada.
- autoclean limpia los nodos no utilizados y renumera las entidades físicas.



- `names` conserva los nombres de las regiones físicas.

Ahora, las dos mallas generadas se integran en una sola malla. En la Fig. 4.32, se ingresa el código y se obtiene una malla resultante con 13844 elementos.

```
ElmerGrid 2 2 estator - in rotor -unite -autoclean -names -out -MaquinaCompleta
```

**Figura 4.32:** Integración de las mallas

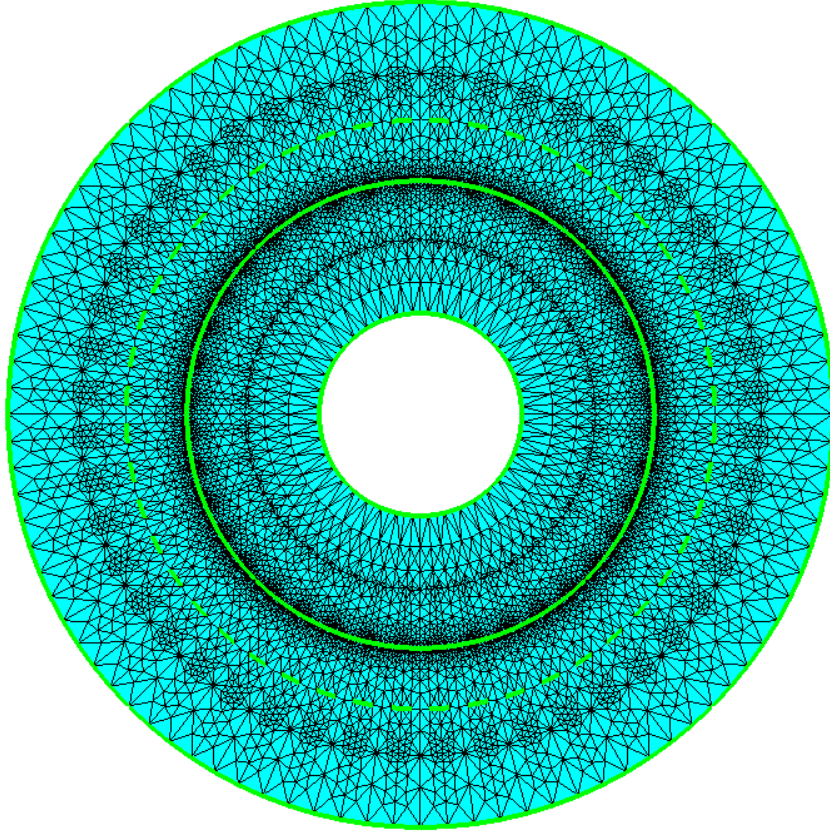
## 4.3. Simulación mediante Elmer FEM

En esta sección, se describe la generación de un archivo SIF usando ElmerGUI, la interfaz de usuario de ElmerGUI que permite desarrollar la programación de forma interactiva o de forma manual. Para este caso de estudio, se realizará manualmente, primeramente se simula el modelo en estado estacionario para su verificación y con el objetivo de comprobar que las condiciones de contorno estén correctamente establecidas. Por último se simula el modelo en estado transitorio en el tiempo y se analiza cada uno de los resultados obtenidos tales como: par eléctrico, corrientes del estátor y voltajes del estátor.

### 4.3.1. Simulación en estado estacionario

Este proceso comienza preparando el archivo SIF, el cual contendrá configuraciones básicas de simulación, descripción de las regiones físicas, condiciones de contorno, definición de los materiales, configuraciones específicas del solucionador y configuraciones para los solucionadores numéricos.

Para el caso del estado estacionario, se simula la distribución de densidad de flujo bajo la excitación DC para verificar que las condiciones de contorno estén correctamente establecidas por el modelo. En la Fig. 4.33 se puede apreciar la vista principal del modelo importado a Elmer FEM de la máquina de inducción hexafásica. Las ranuras del rotor se simulan utilizando aire como material, debido a que la excitación del estátor es constante y no requiere modelación del circuito del rotor.



**Figura 4.33:** Modelo de la MI Hexafásica importada en ElmerGUI

#### 4.3.1.1. Estructura SIF

Al momento de abrir el archivo creado geoméricamente, con la malla correspondiente, se procede a verificar los nombres de las entidades físicas y límites que se utilizan en la geometría, tanto para el rotor, como para el estátor. En la estructura del archivo SIF en primer lugar se debe ingresar la sección del encabezado, en el cual se definen todos los directorios utilizados en el modelo. En la Fig. 4.34, se puede observar toda la sección del encabezado, donde, la malla está localizada en un directorio "*im*", los resultados en "*results*". Los directorios se utilizarán para almacenar los resultados de la simulación. Adicionalmente, se ingresan los materiales extras que no existen en el paquete de ElmerSolver y el programa realizará la búsqueda de los materiales dentro de dicha carpeta "*materials*".

```

include transient_params.dat

Header
  Mesh DB "MaquinaCompleta"
  Results Directory "resultados"
  Include Path "materiales"
End

Simulation
  Max Output Level = 3
  Coordinate System = Cartesian 2D
  Coordinate Scaling = 0.001

  Circuit Model Depth = Real $ I

  Simulation Type = Transient
  Timestepping Method = BDF
  BDF Order = 2

  Output Intervals = 1
  Timestep Sizes = $ 1/f/15
  Timestep Intervals = $ 25*15

  Use Mesh Names = Logical True
End

```

**Figura 4.34:** Sección del encabezado del SIF

En la siguiente sección se describe los parámetros generales de simulación. *MaxOutput Level*, define cuanta información imprime cada solucionador en la interface de ElmerSolver, *Coordinate Scaling* es necesario para escalar las unidades de malla en  $[mm]$  a  $[m]$  utilizado por Elmer Fem, *SimulationType* selecciona el tipo de simulación que puede ser transitoria o estable, *Use Mesh Names* permite el uso de etiquetas de entidades físicas para cuerpos separados dentro de la SIF.

#### 4.3.1.2. Materiales y fuerzas corporales

Para la creación de las propiedades del material de todas las regiones, se crea una sección separada para cada material. Al momento de realizar la simulación en estado estacionario se requiere dos materiales: aire y acero eléctrico laminado. En la Fig. 4.35, se ingresan los respectivos materiales y sus propiedades.

```

! Aire
Material 1
  Relative Permeability = 1
  Electric Conductivity = 0
End

! Acero eléctrico laminado
Material 2
  Name = "Acero"
  INCLUDE el_steel_M800_65A
  Electric Conductivity = 0
End

! Aluminio
Material 3
  Relative Permeability = 1
  ! Electric Conductivity = 36e6 ! [1/(Ohm*m)] rotor frio
  Electric Conductivity = 24e6 ! [1/(Ohm*m)] rotor caliente
End

! Cobre
Material 4
  Relative Permeability = 1
  Electric Conductivity = 48e6
End

```

**Figura 4.35:** Ingreso de materiales

Como se mencionó en la sección anterior, se pueden ingresar archivos externos de materiales con el comando *INCLUDE*, como muestra en la Fig. 4.35, para ingresar

el material de acero eléctrico laminado. Este es un archivo comprimido que contiene información de la curva BH requerida para el modelo del material, en la Fig. 4.36, se muestra toda la información de dicho material.

```

Electric Conductivity = 0
H-B Curve = Variable coupled iter
Real Monotone Cubic
0 0
0.10 28.98
0.20 40.29
0.30 46.63
0.40 52.23
0.50 58.57
0.60 66.02
0.70 74.90
0.80 85.01
0.90 99.45
1.00 121.1
1.10 153.8
1.20 211.9
1.30 343.7
1.40 754.8
1.50 2062
1.60 4485
1.70 7899
1.718 8779
1.746 10000
1.883 20000
1.947 30000
2.018 50000
2.117 100000
2.262 200000
2.394 300000
4.000 1000000
End

```

**Figura 4.36:** Ingreso del material acero eléctrico

Ahora, se requiere describir todas las fuerzas externas que actúan sobre los cuerpos de la malla. En la Fig. 4.37, se puede visualizar el código ingresado. En este sentido, *Body Force* 1 se utiliza para verificar que el límite entre el estátor y el rotor se resuelva correctamente. Contiene un comando que hace rotar todos los cuerpos 10 grados a lo largo del eje  $Z$  [24], mediante el cual todos los cuerpos del rotor se verán afectados por esta fuerza corporal. Las fuerzas corporales desde la dos hasta la siete, imponen densidades de corriente en las ranuras de las seis fases de la máquina eléctrica en estudio.

La corriente pico de fase es de  $15,56 \text{ A rms}$  y la ranura contiene 32 conductores, con un área de aproximadamente  $1,4E - 4 \text{ m}^2$ . Por lo tanto, la densidad de corriente máxima en la ranura es aproximadamente de  $3E6 \text{ A/m}^2$ .

```

Body Force 1
Mesh Rotate 3 = Real MATC "10"
End
! Para A+
Body Force 2
Current Density = 3e6
End
! Para B+
Body Force 3
Current Density = 3e6
End
! Para C+
Body Force 4
Current Density = 3e6
End
! Para d-
Body Force 5
Current Density = -3e6
End
! Para e-
Body Force 6
Current Density = -3e6
End
! Para f-
Body Force 7
Current Density = -3e6
End

```

**Figura 4.37:** Fuerzas externas

En cada cuerpo se tiene que asignar el material y la fuerza corporal. Además, se coloca la *Ecuación1* para indicar a ElmerSolver que los solucionadores de dicha ecuación están activos en los cuerpos. En la Fig. 4.38, se puede apreciar el ingreso de dos de los 27 cuerpos que tiene el modelo. En la sección de Anexos es posible visualizar completamente el desarrollo.

```

Body 1
Name = U_plus
Equation = 1
Material = 4
End
Body 2
Name = W_minus
Equation = 1
Material = 4
End
Body 3
Name = V_plus
Equation = 1
Material = 4
End
...

```

**Figura 4.38:** Ingreso de los cuerpos

#### 4.3.1.3. Solucionadores

El siguiente paso es definir los solucionadores que se utilizan en el modelo, ingresando los solucionadores que estarán activos. ElmerSolver es un módulo que realiza cálculos sobre la malla o los datos de campo, en el mismo se pueden simular diferentes solucionadores, como en este caso:

Los solucionadores 2, 3 y 4 que muestra en la Fig. 4.39, calcularán y cambiarán las variables de campo sobre la malla.

Solver 1, se utiliza para hacer girar la malla del rotor, esto es para verificar que la superficie deslizante estén correctamente definidos en el modelo.

Solver 2, este es el solucionador principal, se calcula el potencial del vector magnético  $A$ , este se calcula en  $2D$ . La tolerancia de convergencia del sistema no lineal se establece en  $1E - 16$ , donde la tolerancia de convergencia del sistema lineal. En el caso de solucionadores iterativos, debe ser siempre menor que la tolerancia de convergencia no lineal. Luego, se ingresan todas las líneas del sistema lineal de manera iterativa para los solucionadores 2, 3 y 4.

Solver 3, en esta sección se realiza la operación de cálculo de la densidad de flujo y los campos de densidad de corriente. Esto a partir de la solución del campo magnético  $A$ , proporcionada a partir del solucionador anterior.

```

Equation 1 :: Active Solvers(3) = 2 3 4
Solver 1
Exec Solver = Before timestep
Equation = MeshDeform
Procedure = "RigidMapper" "RigidMapper"
End
Solver 2
Exec Solver = Always
Equation = Circuits
Variable = X
Procedure = "CircuitsAndDynamics" "CircuitsAndDynamics"
No Matrix = Logical True
End
Solver 3
Exec Solver = Always
Equation = MgDyn2D
Variable = A
Procedure = "MagnetoDynamics2D" "MagnetoDynamics2D"
Stabilize = False
Partition Local Constraints = Logical True
Nonlinear System Compute Change in Scaled System = Logical True
Nonlinear System Convergence Measure = Residual
Nonlinear System Convergence Tolerance = 1e-6
Nonlinear System Max Iterations = 20
Nonlinear System Min Iterations = 1
Nonlinear System Relaxation Factor = 1.0
Nonlinear System Newton After Iterations = 7
Export Lagrange Multiplier = Logical True
Linear System Abort Not Converged = False
Linear System Solver = Iterative
Linear System Iterative Method = GCR
Linear System GCR Restart = 500
Bicgstabl Polynomial Degree = 4
Linear System Preconditioning = ILU4
Linear System Max Iterations = 1500
Linear System Residual Output = Integer 23
Linear System Convergence Tolerance = 1e-7 ! 2.0e-6
Mortar BCs Additive = Logical True
End
Solver 4
Exec Solver = Always
Equation = CalcFields
Potential Variable = "A"
Procedure = "MagnetoDynamics" "MagnetoDynamicsCalcFields"
Calculate Nodal Forces = Logical True
Calculate Magnetic Torque = Logical True
Calculate Current Density = Logical True
Calculate Magnetic Vector Potential = Logical True
End
Solver 5
Exec Solver = Always
Equation = CircOutput
Procedure = "CircuitsAndDynamics" "CircuitsOutput"
End

```

**Figura 4.39:** Solucionadores

#### 4.3.1.4. Condiciones de contorno

Para el modelo, se tiene el límite del estátor externo y el límite del rotor interno. Se asume la condición de límite de Dirichlet [6, 26] de flujo magnético cero que fluye a través del límite. Esto como se aprecia en la Fig. 4.40,  $A = 0$  en esos límites.

```

Boundary Condition 1
Name = OuterStator
A = Real 0
End
Boundary Condition 6
Name = InnerRotor
A = Real 0
End

```

**Figura 4.40:** Límite interno y externo

Los lados del modelo de la máquina representan una condición de límite anti-periódico, haciendo esto un límite periódico. En la Fig. 4.41, se muestra las palabras clave que definen esta condición de límite tanto para el estátor y el rotor.

```
Boundary Condition 2
Name = EstatorDerecha
Mortar BC = Integer 3
Mortar BC Static = Logical True
Anti Radial Projector = Logical True
Galerkin Projector = Logical True
End
Boundary Condition 3
Name = EstatorIzquierda
End
Boundary Condition 4
Name = Corredizo_Estator
Mortar BC = Integer 9
Anti Rotational Projector = Logical True
Galerkin Projector = Logical True
End
```

**Figura 4.41:** Límites periódicos

### 4.3.2. Simulación transitoria en el tiempo

En esta sección se realiza una simulación no lineal y transitoria en el tiempo, con ecuaciones de circuitos. El modelo ahora opera con fuentes de voltaje. La MI hexafásica se simula a una velocidad nominal constante del rotor, con el objetivo de analizar el rendimiento nominal del motor.

#### 4.3.2.1. Parámetros y encabezado de la máquina de inducción hexafásica

En primer lugar, se incluye el documento *parametros\_trasendiente.dat* donde está la información de los parámetros de la MI hexafásica y los circuitos de voltaje del estátor. En la Fig. 4.42, se puede verificar parte de dicha información, también disponible en los Anexos.

```
$ f = 60 ! [Hz]
$ U = 440 ! [V] voltaje
$ pp = 2
$ n = 1740 ! [rpm]

$ l = 8.160 ! [m]
$ Rs = 0.353 ! [Ohm]
$ Nph = 256 !
$ w_syn = 1800/60*2*pi ! [rad/s]

!analytically estimated values:
$ L_ew = 0.06747 ! [H]
$ R_er = 0.424 ! [Ohm]
$ L_er = 0.012 ! [H]
```

**Figura 4.42:** Ingreso de los parámetros de la MI hexafásica

Ahora, en la Fig. 4.43, se realiza el cambio del tipo de simulación para este modelo. Se puede observar que ahora la simulación es de régimen transitorio. *BDF* es el método de pasos de tiempo que para este modelo se realiza de segundo orden. El tamaño del paso de tiempo es de  $1/f/40$ , lo que representa 40 pasos de tiempo durante un periodo eléctrico, en [24] se recomienda que los pasos de tiempo no sean menores a 20.



```

include resultados_transitoria.dat
Header
  Mesh DB "MaquinaCompleta"
  Results Directory "resultados"
  Include Path "materiales"
End

Simulation
  Max Output Level = 3
  Coordinate System = Cartesian 2D
  Coordinate Scaling = 0.001
  Circuit Model Depth = Real $ I
  Simulation Type = Transient
  Timestepping Method = BDF
  BDF Order = 2
  Output Intervals = 1
  Timestep Sizes = $ 1/f/15
  Timestep Intervals = $ 1000
  Use Mesh Names = Logical True
End

```

**Figura 4.43:** Encabezado de la simulación transitoria

#### 4.3.2.2. Materiales para la simulación transitoria en el tiempo

Para esta simulación, se ingresan dos materiales más que en la sección anterior de la simulación en estado estacionario, estos son para el devanado de cobre del estátor y aluminio para el caso de los devanados del rotor. En la Fig. 4.44, se muestran los materiales y sus características para la simulación.

```

! Aire
Material 1
  Relative Permeability = 1
  Electric Conductivity = 0
End
! Acero lamimado
Material 2
  Name = "Acero"
  INCLUDE el_steel_M800_65A
  Electric Conductivity = 0
End
! Aluminio
Material 3
  Relative Permeability = 1
  ! Electric Conductivity = 36e6 ! [1/(Ohm*m)] rotorfrío
  Electric Conductivity = 24e6 ! [1/(Ohm*m)] rotorcaliente
End
! Cobre
Material 4
  Relative Permeability = 1
  Electric Conductivity = 48e6
End

```

**Figura 4.44:** Materiales para la simulación transitoria en el tiempo

#### 4.3.2.3. Fuerzas corporales para la simulación transitoria en el tiempo

En el *Body Force* 1, se define las fuentes de los circuitos eléctricos, la tensión nominal línea–línea es de 440 V. Se supone que la conexión del devanado es simétrico, por lo tanto en la Fig. 4.45, se muestra la fuerza corporal de un polo de la MI hexafásica y se definen los voltajes pico de fase.

```

Body Force 1
Name = "Circuit"
A_a = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f)"
B_b = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f-pi/3)"
C_c = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f-2*pi/3)"
D_d = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f-pi)"
E_e = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f-4*pi/3)"
F_f = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f-5*pi/3)"
End
Body Force 7
Mesh Rotate 3 = Variable time
Real MATC "180/pi*2*pi*f/pp*(1-slip)*tx"
End

```

**Figura 4.45:** Fuerzas corporales de la simulación transitoria en el tiempo

#### 4.3.2.4. Cuerpos para la simulación transitoria en el tiempo

En este segmento se ingresan todos los cuerpos de la geometría de la MI hexafásica, las ranuras del rotor están nombradas como *Cuerpo1* hasta el *Cuerpo40*, que representa las 40 ranuras del rotor. La parte del estátor y sus ranuras están representadas como se aprecia en la Fig. 4.46 En este apartado se muestran tres fases de las seis de la MI. Se destinan los cuerpos a las diferentes ecuaciones, materiales, fuerzas corporales y grupos de par en el caso del rotor.

```

Body 1
  Name = Bar1
  Equation = 1
  Material = 3
  Body Force = 7
  Torque Groups = Integer 1
End
Body 11
  Name = A_plus
  Equation = 1
  Material = 4
End
Body 12
  Name = B_minus
  Equation = 1
  Material = 4
End
Body 13
  Name = C_plus
  Equation = 1
  Material = 4
End
Body 14
  Name = D_minus
  Equation = 1
  Material = 4
End
Body 15
  Name = F_plus
  Equation = 1
  Material = 4
End
Body 16
  Name = E_minus
  Equation = 1
  Material = 4
End
...

```

**Figura 4.46:** Cuerpos para la simulación transitoria en el tiempo

#### 4.3.2.5. Componentes

Cada cuerpo de modelo se asocia con los circuitos utilizando componentes, estos tienen su propio índice corporal. Los componentes son bobinas trenzadas, esto significa que las corrientes parásitas se desprecia en los cables de la bobina y la corriente fluye solo a lo largo del eje de la máquina.

En la Fig. 4.47, se muestra la numeración de los componentes desde el 41. Esto es porque los 40 primeros están reservados para las cuerpos del rotor y se ingresan en el documento *parametros\_trasendiente.dat*. Todos los componentes tienen su propia resistencia y número de vueltas.

```

Component 41
  Name = String Aplus
  Body = Integer 11
  Coil Type = String Stranded
  Number of Turns = Real $ Nph/2
  Resistance = Real $ Rs
End
...

```

**Figura 4.47:** Componentes

#### 4.3.2.6. Solucionadores para el modelo en estado transitorio

El solucionador *CircuitsAndDynamics* discretiza las ecuaciones que se adjuntan a la matriz principal de la solución de campo [24] y se utiliza la palabra en la sección del solucionador *MagnetoDynamics2D*. En la Fig. 4.48, se adicionan dos solucionadores más para inicializar y realizar el proceso de los grados de libertad del circuito.

```
Solver 2
Exec Solver = Always
Equation = Circuits
Variable = X
Procedure = "CircuitsAndDynamics" "CircuitsAndDynamics"
No Matrix = Logical True
End
Solver 5
Exec Solver = Always
Equation = CircOutput
Procedure = "CircuitsAndDynamics" "CircuitsOutput"
End
```

Figura 4.48: Solucionadores del estado transitorio

#### 4.3.2.7. Definición de los circuitos eléctricos

Todas las ecuaciones del circuito del estátor se definen en el documento incluido *parametros\_trasendiente.dat*. Los circuitos son similares para todas las fases, en la Fig. 4.49, se muestra el circuito para la fase A. El número de circuitos son 25, de los cuales 24 son para representar las bobinas del estátor y uno para el rotor.

Esto se realiza asumiendo que el devanado es simétrico [6], y se alimenta por voltajes sinusoidales simétricos.

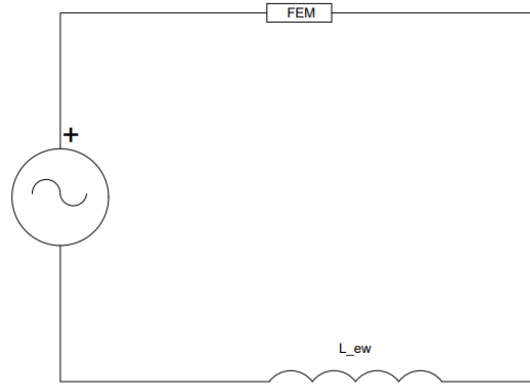
El diagrama del circuito se presenta en la Fig. 4.50. Este circuito consiste de una fuente de voltaje, el componente FEM y la inductancia de la bobina del devanado final.

```
$ Circuits = 25

! Ax' + Bx = fuente

$ C.1.variables = 6
$ C.1.perm = zeros(C.1.variables)
$ C.1.A = zeros(C.1.variables, C.1.variables)
$ C.1.B = zeros(C.1.variables, C.1.variables)
$ C.1.Mre = zeros(C.1.variables, C.1.variables)
$ C.1.Mim = zeros(C.1.variables, C.1.variables)
```

Figura 4.49: Circuitos



**Figura 4.50:** Diagrama de circuito de fase [6]

Los circuitos se definen mediante un sistema de ecuaciones diferenciales de primer orden con coeficientes constantes:

$$Ax' + Bx = F(t) \quad (4.29)$$

donde,  $x$  es el vector de variables de circuito,  $A$  es la matriz de coeficientes del componente derivado del tiempo,  $B$  es la matriz de coeficientes proporcionales y  $F$  es la parte de origen que se establece en la sección *Body Force* 1.

Las matrices  $A$  y  $B$ , se ingresan en la Fig. 4.51, utilizando las leyes de Kirchhoff 1 y Kirchhoff 2. A continuación se ingresa en (4.29) y las dos ecuaciones se resuelven juntas como un solo sistema.

El diagrama de fase tiene como modelo seis circuitos conectados en serie, como se puede apreciar en la Fig. 4.50. En la Fig. 4.51, se puede visualizar que para cada elemento se definen dos variables de circuito, 1 y 2 para corriente y voltaje, 3 y 4 para bobinado conectado a FEM, 5 y 6 para el devanado final fuera del dominio FEM. Las matrices se ingresan considerando cada ley y elemento del circuito por separado.

```

§ C.1.name.1 = "corr_sa"
§ C.1.name.2 = "volt_sa"
§ C.1.name.3 = "corr_componente(11)"
§ C.1.name.4 = "volt_componente(11)"
§ C.1.name.5 = "corr_ewa"
§ C.1.name.6 = "volt_ewa"
§ C.1.B(0,1) = 1
§ C.1.source.1 = "A_a"

```

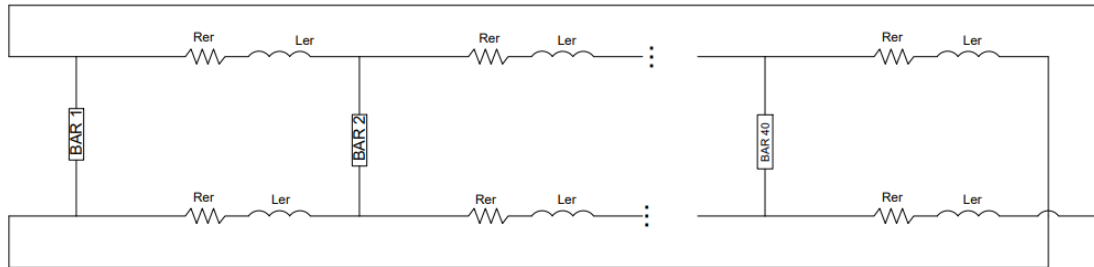
**Figura 4.51:** Kirchhoff 1 y Kirchhoff 2

#### 4.3.2.8. Bobinado del rotor tipo jaula de ardilla

En la Fig. 4.52, se puede observar el esquema del bobina tipo jaula de ardilla. Se utilizan en el modelo de la MI, las condiciones de contorno antiperiódicas establecidas mediante los cuerpos de contorno de conexión cruzada 1 y 40. Donde,  $R_{er}$  y  $L_{er}$  son la resistencia y la inductancia de las secciones de los anillos del rotor.

Para un polo de la MI hexafásica, el número de elementos rotor sería de 60, incluyendo  $R_{er}$  y  $L_{er}$  dentro de las variables de circuito. Lo cual, sería demasiadas variables

para calcular manualmente. Por esta razón se crea un script, que automatice una ecuación de circuito para los devanados del rotor. Los parámetros de entrada incluyen el número de cuerpos, las condiciones de simetría, el número de cortes y el tipo de conductores.

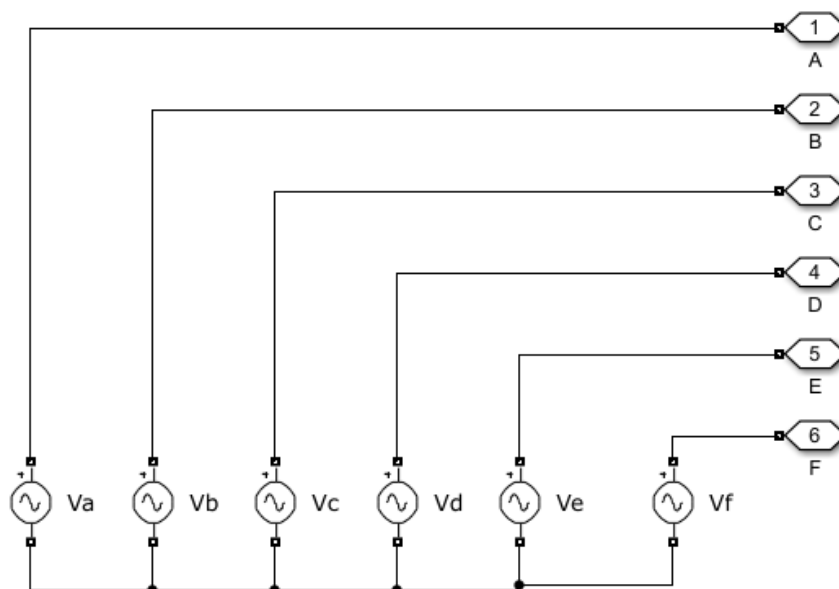


**Figura 4.52:** Diagrama de bobinado del rotor tipo jaula de ardilla

# 5 Simulación mediante el método de vectores espaciales

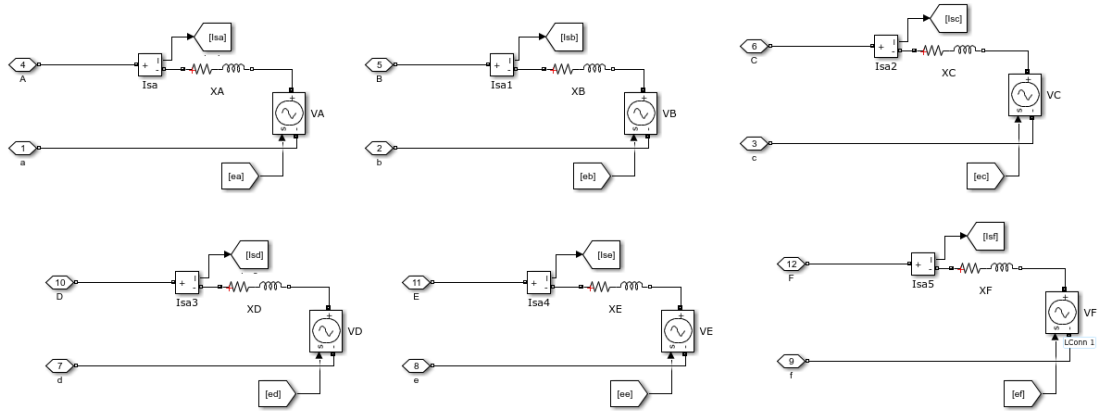
## 5.1. Modelo circuital

Para la elaboración del modelo la máquina de inducción hexafásica mediante el método de vectores espaciales, se utiliza el software Matlab y su herramienta Simulink . En la Fig. 5.1 se muestra la fuente de alimentación que se utiliza en el modelo, cada fuente está desfasada  $\frac{\pi}{3}$  una de otra.



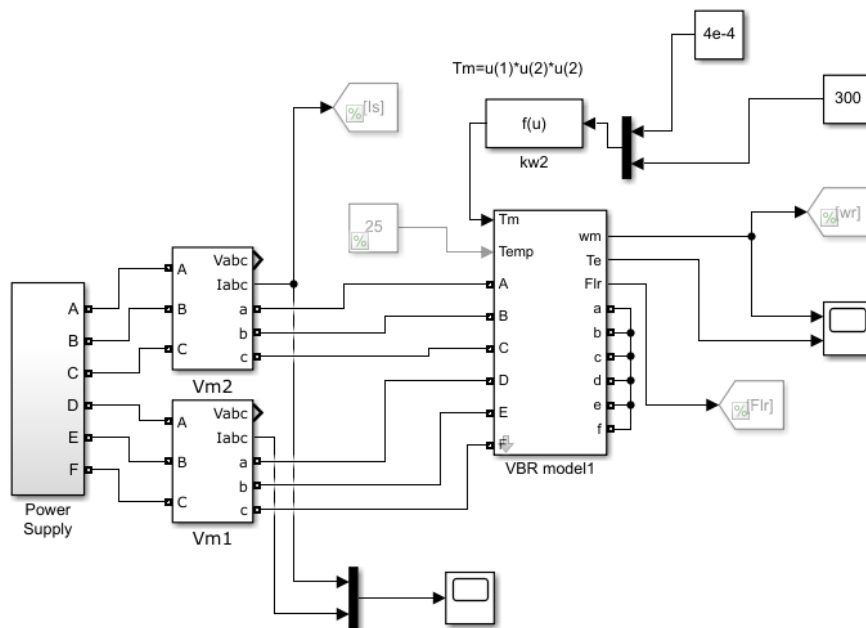
**Figura 5.1:** Fuente de alimentación del modelo circuital

El modelo dinámico de la máquina de inducción, se desarrolla mediante los vectores espaciales de flujo en el estator y rotor. El desarrollado completo se puede visualizar en la Sección 2.6. En la Fig. 5.2 se observa el modelo dinámico de la máquina de inducción hexafásica realizado en Simulink, donde se utiliza 6 fuentes de tensión detrás de la reactancia VBR.



**Figura 5.2:** Modelo dinámico

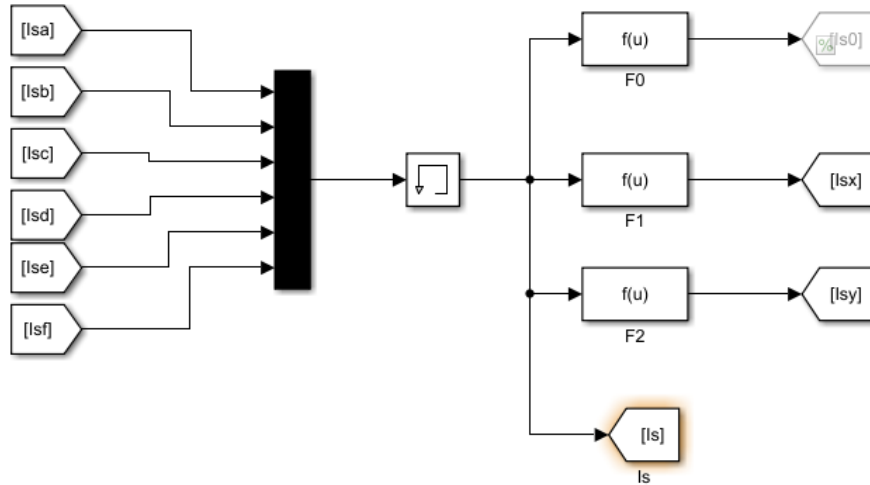
En la Fig. 5.3 se puede visualizar el diagrama completo del modelo en vectores espaciales para la simulación en Simulink, así como las variables de salida, que son las corrientes y el par eléctrico.



**Figura 5.3:** Modelo completo de la máquina de inducción en Simulink

En la Fig. 5.4 se visualiza el módulo de cálculo de las corrientes ( $i_{sx}$ ,  $i_{sy}$ ), esto a partir del ingreso de las 6 corrientes de estátor.

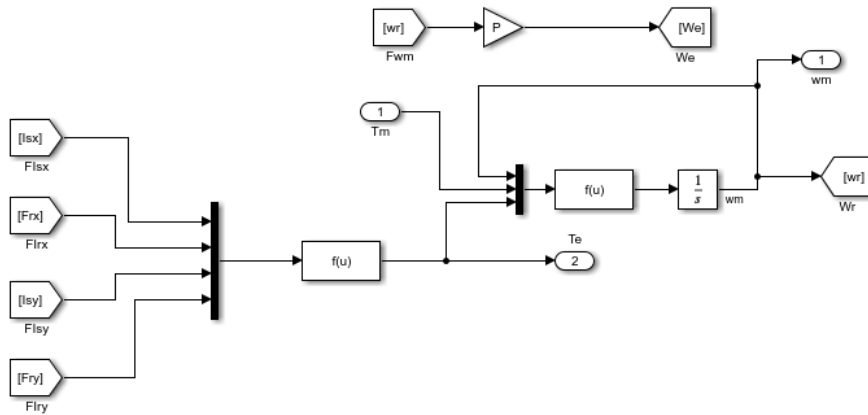




**Figura 5.4:** Diagrama de las corrientes del modelo dinámico

En la Fig. 5.5 se observa el módulo de cálculo, para obtener el par eléctrico de la máquina de inducción hexafásica  $T_e$ . De tal manera que se puede observar la velocidad de la máquina mediante la siguiente Ecuación. 5.1,

$$T_e - T_m = J\ddot{\theta} + p\dot{\theta} \quad (5.1)$$



**Figura 5.5:** Diagrama del cálculo del par eléctrico

En la Fig. 5.6 se visualiza el módulo principal del modelo VBR, de la MI, donde, la función fcn permite calcular las 6 fuerzas electromotrices ( $e_a, e_b, e_c, e_d, e_f$ ) y los enlaces de flujo ( $F_{rx}, F_{ry}$ ) de la MI hexafásica.

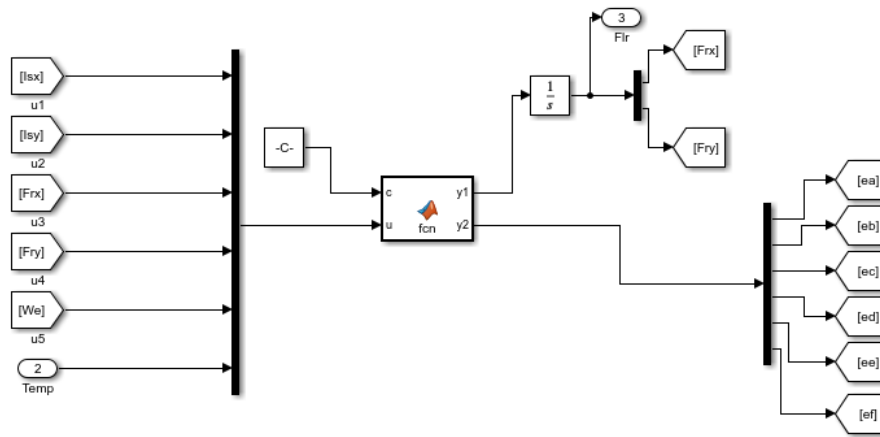
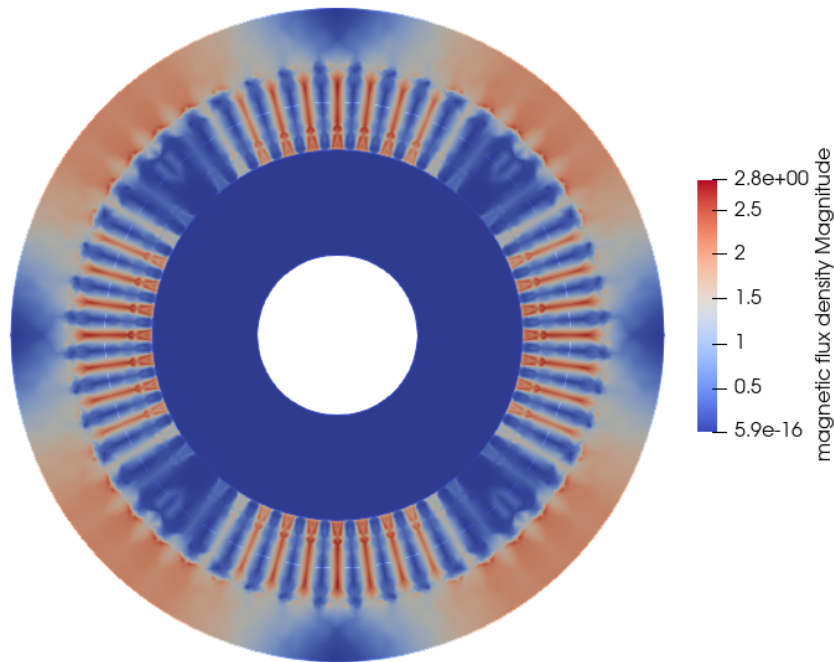


Figura 5.6: Diagrama de enlace de flujo y la fem

## 6 Análisis de Resultados.

### 6.1. Análisis en estado estacionario

Mediante la plataforma paralela a Elmer FEM que es Paraview, se pueden visualizar los resultados en forma gráfica, como es el flujo magnético y las líneas de flujo magnético. En la Fig. 6.1, se puede observar el flujo magnético del modelo de MI hexafásica, donde, los valores de densidad de flujo están dentro de los valores esperados para esta simulación.



**Figura 6.1:** Densidad de flujo en estado estacionario

La Fig. 6.2, muestra las líneas de flujo magnético, donde las líneas son continuas sobre la superficie deslizante y tangenciales en los límites interior y exterior. La periodicidad y la simetría de las líneas permiten verificar que se han resuelto correctamente.

Por lo tanto, se puede concluir que este modelo está configurado correctamente y es posible realizar el siguiente paso consistente en la simulación transitoria en el tiempo.

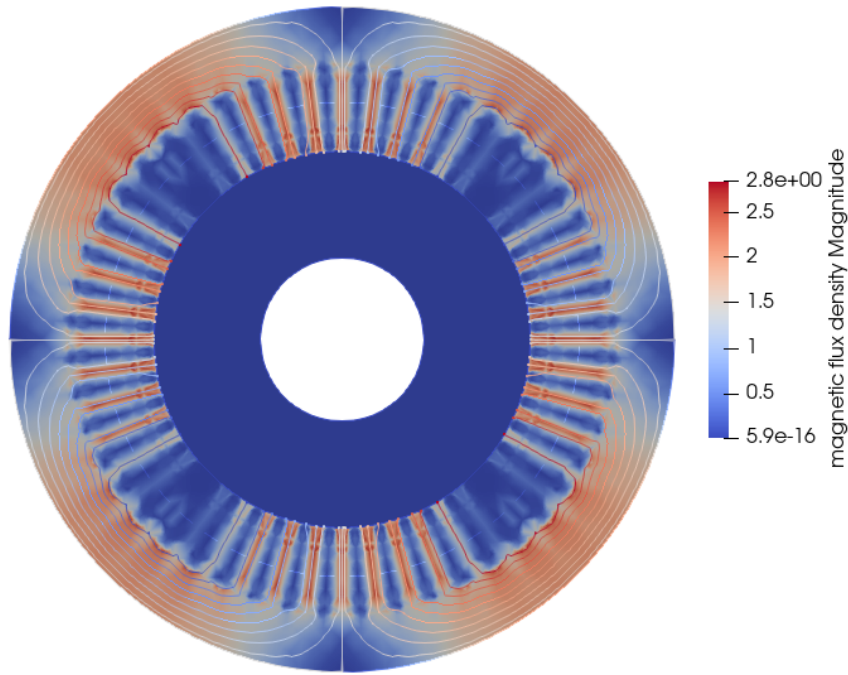
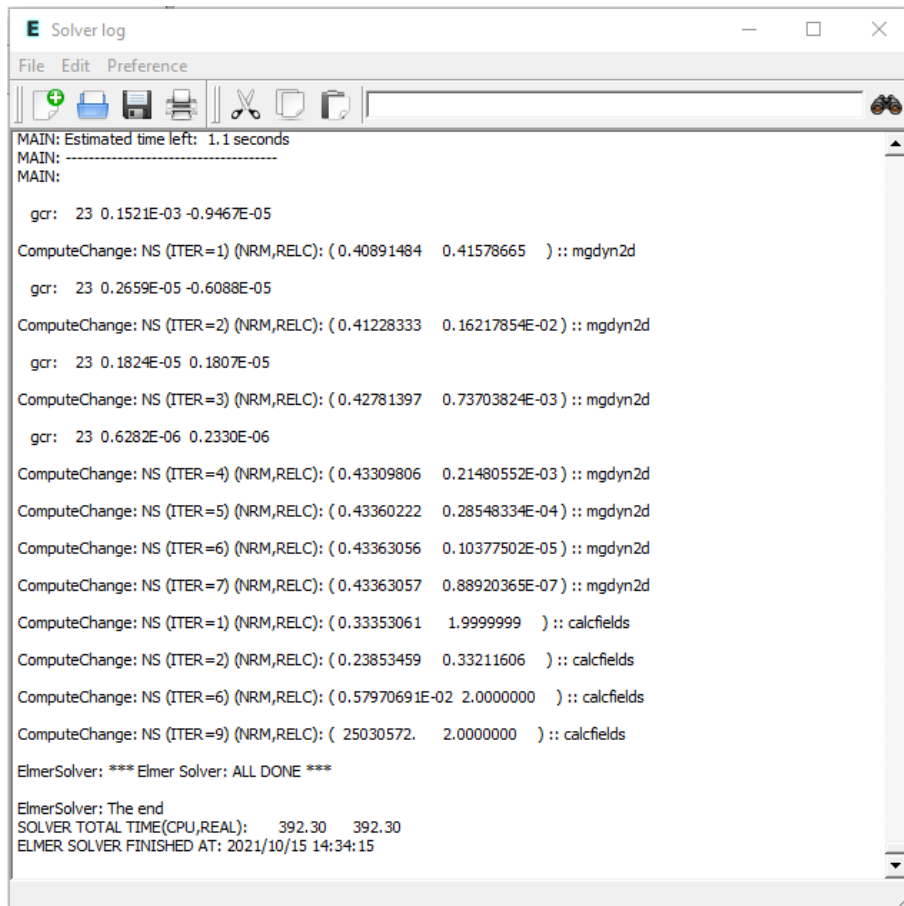


Figura 6.2: Líneas de flujo magnético en estado estacionario

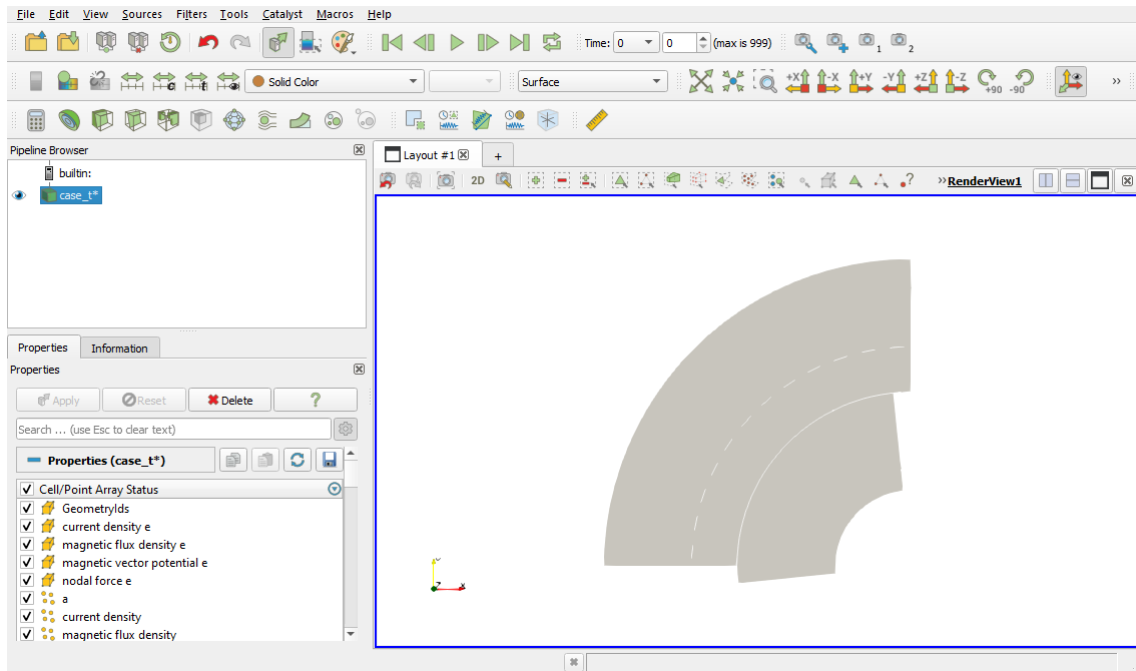
## 6.2. Análisis de simulación transitoria en el tiempo

En esta sección, una vez realizada la compilación del programa, se tiene que esta modelación demoró 392 segundos. Los solucionadores del modelo simulado se guardan en el documento de salida, tipo *vtu*, en la carpeta llamada *resultados*.



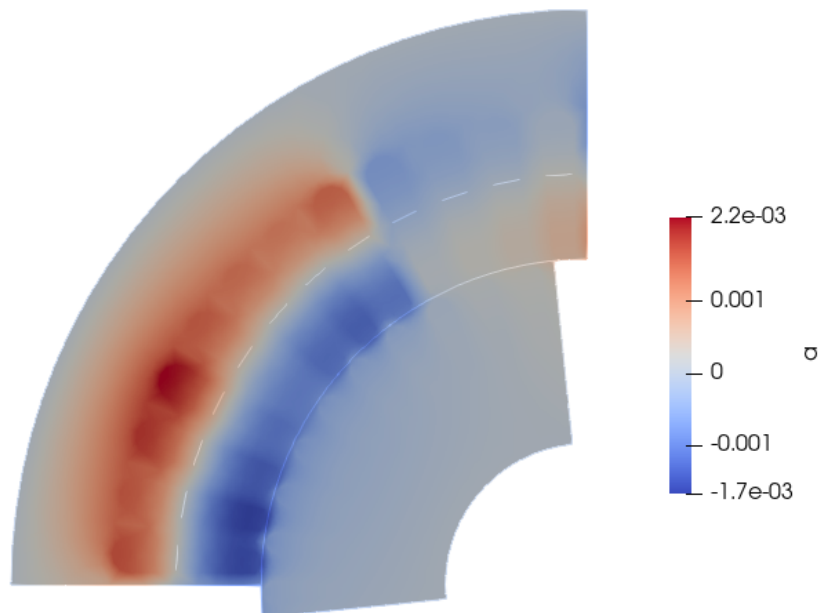
**Figura 6.3:** Tiempo de simulación en estado estacionario

Mediante el software Paraview se visualizan los resultados. Se realiza la apertura del archivo *vtu* y luego se presiona el botón *Apply*. Desde el menú desplegable "Solid Color" como muestra en la Fig. 6.4, se selecciona el campo que se desea mostrar. Las flechas de color verde, permiten el paso del tiempo de la simulación.

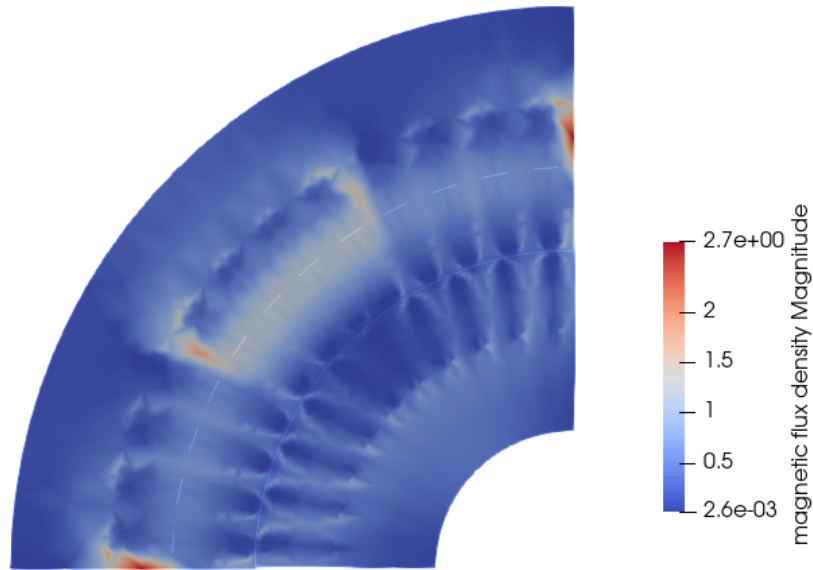


**Figura 6.4:** Interface de usuario principal de Paraview

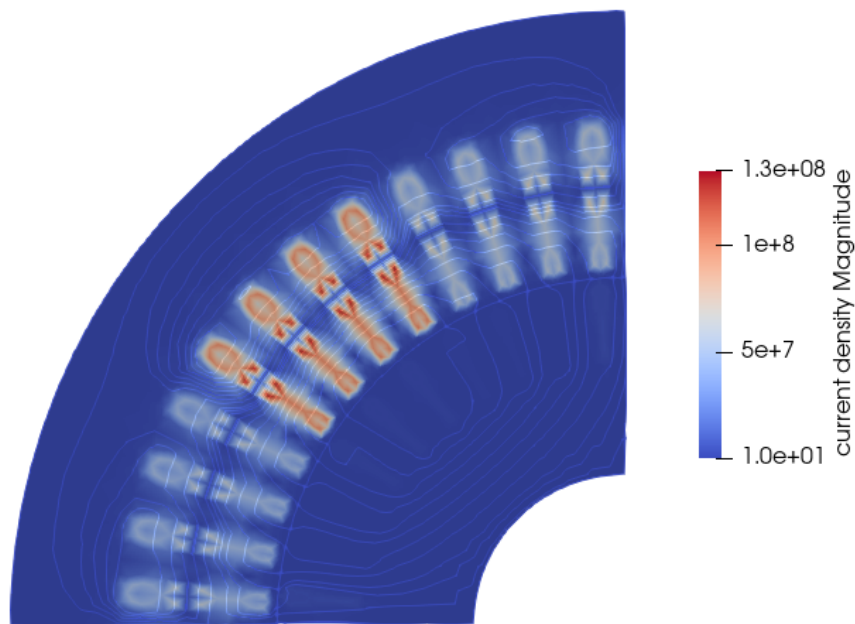
La Fig. 6.5, muestra el potencial del vector magnético  $A$ , en Fig. 6.6 se puede visualizar la densidad de flujo  $B$  y en la Fig. 6.7, la densidad de corriente  $J$ . Todos estos resultados son guardados por el solucionador 6.



**Figura 6.5:** Potencial del vector magnético



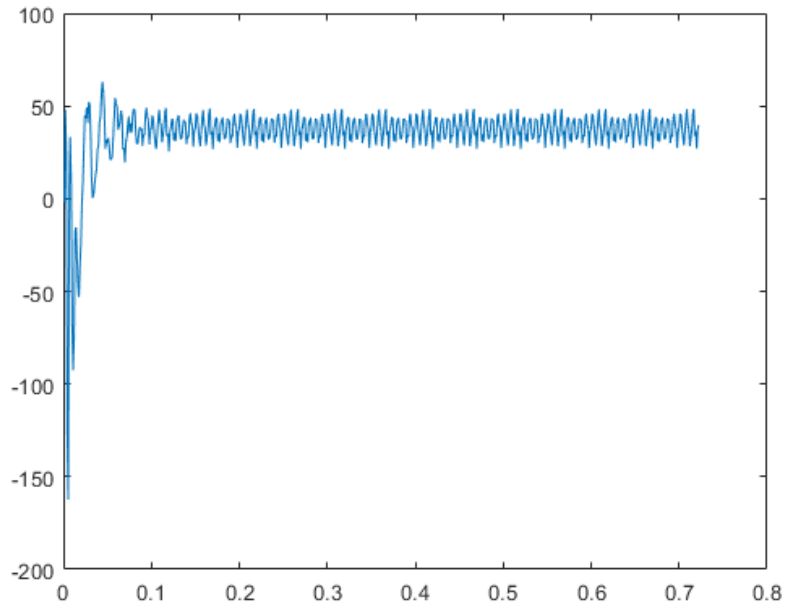
**Figura 6.6:** Densidad de flujo B



**Figura 6.7:** Densidad de corriente

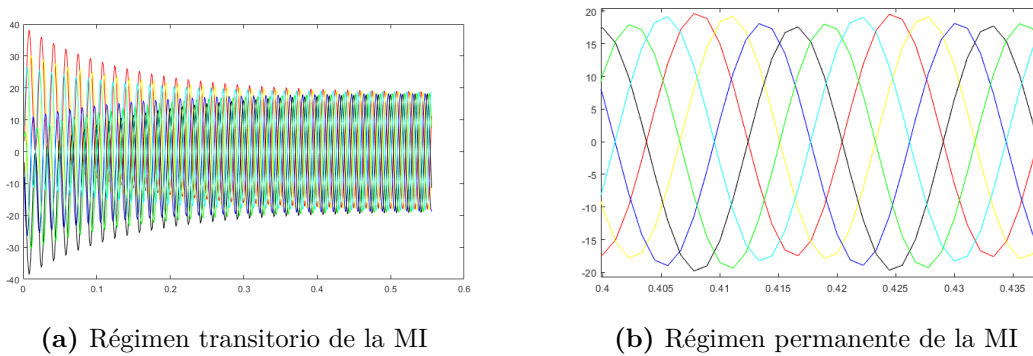
El solucionador 7 almacena todos los datos escalares en el archivo *transient\_result.dat* como se expuso en el capítulo anterior, en el formato ASCII txt. Cada paso de tiempo se guarda en su propia fila del documento y los nombres de las variables se muestra en el archivo *transient\_results.dat.names*. Estos resultados se pueden visualizar con el software Matlab, para lo cual es necesario realizar la importación en formato ASCII.

En la Fig. 6.8, se puede apreciar la curva del par de la MI. El estado estacionario se alcanza a los 0,15 segundos aproximadamente. El par promedio es de 35 Nm.



**Figura 6.8:** Par eléctrico de la MI con el FEM

En la Fig. 6.9 se puede visualizar las curvas de las corrientes del estátor. La corriente de fase máxima es de aproximadamente 18,5 A, lo que es 13,08 A rms.



**(a)** Régimen transitorio de la MI

**(b)** Régimen permanente de la MI

**Figura 6.9:** Corrientes del estátor de la MI mediante la FEM

En la Fig. 6.10, se muestran los voltajes del estátor inicializados en la simulación para las seis fases de la MI.



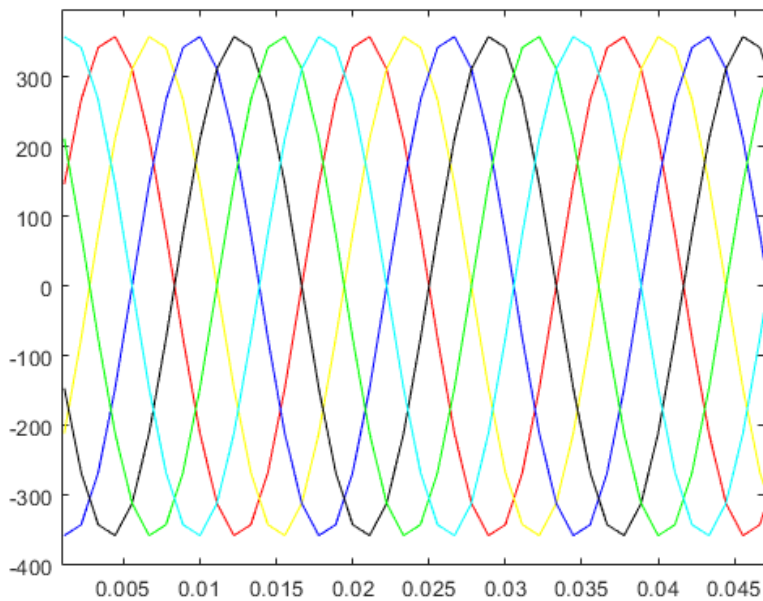
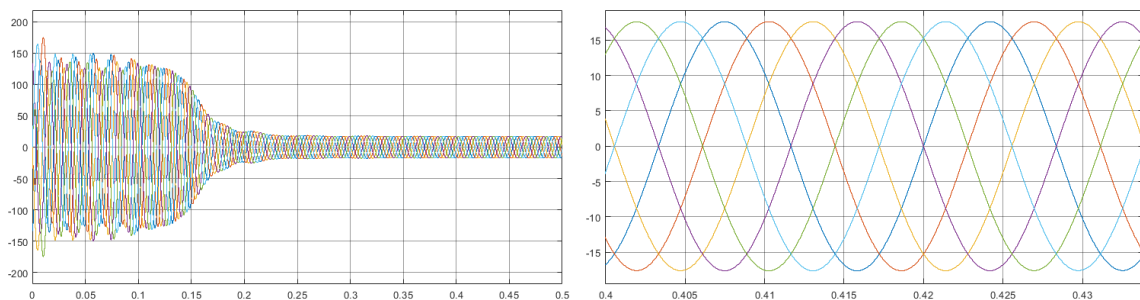


Figura 6.10: Voltajes del estátor de la MI

### 6.3. Simulación de la MI hexafásica con el método de vectores espaciales

#### 6.3.1. Corrientes del estátor

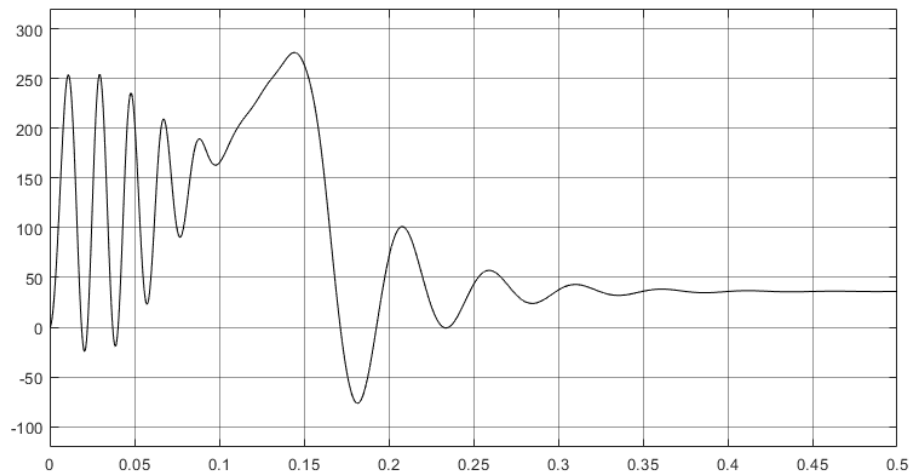
En la Fig. 6.11 se visualiza las corrientes del estátor ( $i_a, i_b, i_c, i_d, i_e, i_f$ ) del modelo de la MI hexafásica, mediante el análisis de vectores espaciales de la máquina de referencia. En la Fig. 6.11a se observa las corrientes en régimen transitorio y en la Fig. 6.11b las corrientes en régimen permanente.



(a) Régimen transitorio del modelo circuital      (b) Régimen permanente del modelo circuital

Figura 6.11: Corrientes del estátor con el método circuital

En la Fig. 6.12 se observa el comportamiento del par eléctrico  $T_e$  de la máquina de inducción hexafásica, donde, el programa Matlab realiza una integración mediante el método de Euler a un paso de  $dt = 10\mu s$ .

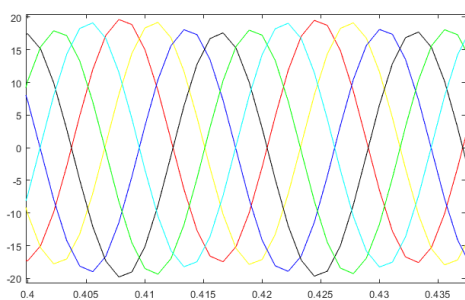


**Figura 6.12:** Par eléctrico con el método de vectores espaciales

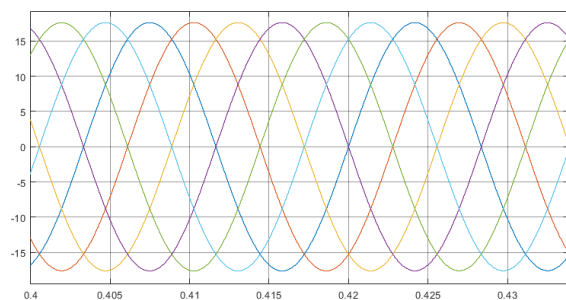
## 6.4. Comparación mediante el modelo de elementos finitos y el modelo circuital de vectores espaciales

En esta sección se desarrolló la comparación de los resultados de los dos métodos analizados en el proyecto. En la Fig. 6.13 se visualiza las corrientes del estátor, donde, para los casos de estudio las corrientes sobrepasan los valores nominales de la máquina eléctrica hexafásica.

Para el modelo de elementos finitos mediante ELMER FEM, el valor de las corrientes de fase máxima en régimen permanente es de aproximadamente 18,5 A - 13,08 A rms. En el modelo circuital de vectores espaciales mediante el software Matlab-Simulink, el valor de corrientes de fase máxima en régimen permanente es de aproximadamente 17,5 A - 12,37 A rms.



(a) Corrientes del estátor mediante la FEM



(b) Corrientes del estátor en Matlab

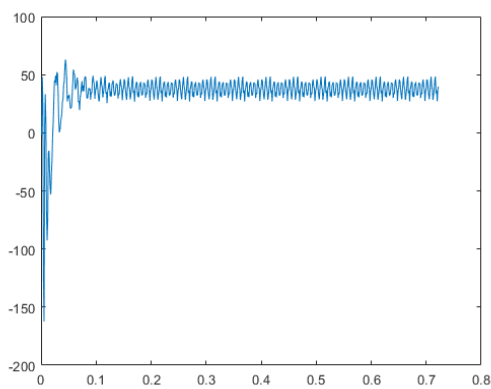
**Figura 6.13:** Gráfico de las corrientes del estátor mediante la FEM

Los valores simuladores son más altos que los parámetros de la MI hexafásica, la corriente tiene un valor de 11 A. Esto debido a que no se toman en cuenta las

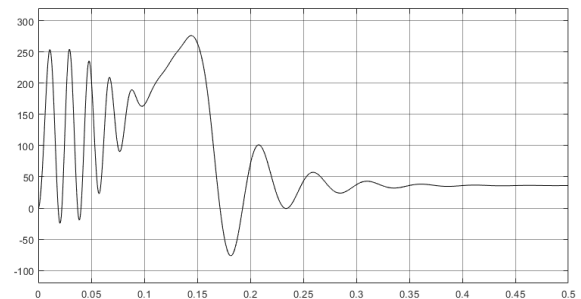
pérdidas de la MI como: pérdidas de hierro, pérdidas por viento, pérdidas mecánicas y pérdidas por corrientes parásitas.

En la Fig. 6.14a, se visualiza el par eléctrico de la MI con el método de elementos finitos. La curva es muy ruidosa por el tiempo de resolución, es recomendable simular con una computadora de nivel industrial y subir el tiempo de resolución para obtener mejores resultados. El régimen permanente se alcanza cerca de los 0,35 s y el par promedio es de 35 Nm.

En la Fig. 6.14b se observa el par eléctrico con el método circuital de vectores espaciales, donde, se visualiza que el régimen permanente se alcanza a los 0,4 s y el par promedio es de 35 Nm. En los gráficos de la Fig. 6.14 existe una diferencia en la constante de tiempo, esto es debido a la discretización del tiempo en las dos simulaciones.



(a) Par mediante la FEM



(b) Gráfica del Par en Matlab

**Figura 6.14:** Comparación de par eléctrico

## 7 Conclusiones y Recomendaciones

### 7.1. Conclusiones

El modelo de la máquina de inducción hexafásica desarrollado en este proyecto fue desarrollado mediante varios programas de código abierto y con licencia GNU. Empezando con el programa GMSH utilizado como pre-procesador, en el cual se construyó toda la geometría de la MI hexafásica. El programa Elmer, realizó la simulación de elementos finitos y dentro del post-procesamiento se utilizaron los programas Paraview y el compilador de Python. De esta forma fue posible visualizar y validar todo el modelo de la máquina de inducción hexafásica.

El mallado de la MI tiene que tener una alta densidad, para poder definir un campo magnético prácticamente continuo. Sin embargo, en este proyecto se orientó a una malla más simple puesto que esto define significativamente el tiempo de simulación.

Elmer Fem tiene una extensa documentación, lo cual ofrece facilidades para la comunidad de usuarios. Un ejemplo de esto es el documento *ElmerTutorials* que es una guía de aplicación de diferentes problemas multifísicos. Estos problemas se desarrollan mediante la interfaz gráfica de Elmer, pero también es posible realizarlo iterativamente. Los tutoriales disponibles contienen ejemplos para áreas tan diversas como: mecánica, electrostática, campos magnéticos, flujo térmico, flujo de fluidos, etc.

En este proyecto técnico, se detalló y explicó cada proceso de la modelación y simulación de la máquina eléctrica de inducción hexafásica. Esto a partir de la modelación de una máquina eléctrica trifásica comercial, obteniendo resultados coherentes de la simulación en estado estacionario y en la simulación transitoria en el tiempo.

Los resultados obtenidos del análisis de la simulación transitoria en el tiempo son algo superiores a los valores nominales de la MI. Esto es debido a que no se han considerado las corrientes parásitas, las pérdidas en el hierro y las pérdidas por ventilación o fricción mecánica. Sin embargo, Elmer FEM está preparado para realizar de forma efectiva estas simulaciones. Su código abierto es una herramienta de trabajo en la que se pueden desarrollar pruebas con los modelos y determinar las pérdidas de las máquinas eléctricas.

Para la simulación transitoria en el tiempo, solo se desarrolla un polo de la MI hexafásica. Esto debido a que Elmer FEM es un programa que exige una extensa demanda computacional. Con lo cual se pudieron obtener en menos tiempo los mismos resultados de la máquina con sus cuatro polos. La simulación completa de la máquina aumenta significativamente los tiempos computacionales y los requerimientos de máquinas más poderosas.

El modelo de la máquina de inducción hexafásica, con el método de vectores espaciales utilizando la herramienta simulink, permitió cumplir el objetivo de comparar

y verificar positivamente el modelo de la MI. El modelo circuital y la herramienta Simulink se desarrollaron en base a un MI trifásica, donde, se demostró ser una herramienta rápida y precisa de simulación. La comparación de los resultados tuvieron errores uno de otro, ya que con la FEM se desarrolla la simulación ingresando valores tales como: Materiales, fuerzas corporales, fuerzas externas, condiciones de contorno, etc. Haciendo que la simulación sea extensa, pero proyectando resultados reales de una máquina eléctrica.

## 7.2. Recomendaciones

Para realizar un proyecto mediante Elmer, es necesario tener un conocimiento previo, desarrollando múltiples ejemplos propuestos por los autores en el tutorial. Esto conduce a solucionar las preguntas que se presentarán en el desarrollo del proyecto.

Mediante la experiencia obtenida al desarrollar la modelación, para trabajos futuros se recomienda que la densidad de la malla sea superior al 95 %. Esto generará resultados más precisos. Esto requiere la utilización de computadores de alta gama, tanto en procesador, como en la memoria RAM.

Para futuras investigaciones con la herramienta Elmer Fem, se puede desarrollar el proyecto tanto de la geometría y el mallado en 3D, para esto se recomienda utilizar herramientas CAD más desarrolladas y personalizadas.

# A Anexos

## A.1. Código de la simulación en estado estacionario

```
Mesh DB "MaquinaCompleta"
Results Directory "resultados"
Include Path "materiales"
End
Simulation Max Output Level = 3
Coordinate System = Cartesian
Coordinate Scaling = 0.001
Steady State Max Iterations = 1
Simulation Type = Steady
Use Mesh Names = Logical True
End
! Aire
Material 1
Relative Permeability = 1
End
! Acero
Material 2
Name = "Acero"
INCLUDE el_steel_M800_65A
End
Body Force 1
Mesh Rotate 3 = Real MATC "10"
End
! A+
Body Force 2
Current Density = 3e6
End
! B+
Body Force 3
Current Density = 3e6
End
! C+
Body Force 4
Current Density = 3e6
End
! d-
Body Force 5
Current Density = -3e6
End
! e-
Body Force 6
Current Density = -3e6
End
! f-
Body Force 7
Current Density = -3e6
End
Body 1
Name = A_plus
Equation = 1
Material = 1
Body Force = 2
End
Body 2
Name = B_plus
Equation = 1
Material = 1
Body Force = 3
End
Body 3
Name = C_plus
Equation = 1
Material = 1
Body Force = 4
End
Body 4
Name = A_minus
Equation = 1
Material = 1
Body Force = 5
End
Body 5
Name = B_minus
Equation = 1
Material = 1
Body Force = 6
End
Body 6
Name = C_minus
Equation = 1
Material = 1
Body Force = 7
End
Body 7
Name = d_plus
Equation = 1
Material = 1
Body Force = 5
End
Body 8
Name = e_plus
Equation = 1
Material = 1
Body Force = 6
End
Body 9
Name = f_plus
Equation = 1
Material = 1
Body Force = 7
End
Body 10
Name = d_minus
Equation = 1
Material = 1
Body Force = 2
End
Body 11
Name = e_minus
Equation = 1
Material = 1
Body Force = 3
End
Body 12
Name = f_minus
Equation = 1
```

Figura A.1: Estado estacionario 1

```

Material = 1
Body Force = 4
End
Body 13
Name = AceroEstator
Equation = 1
Material = 2
End
Body 14
Name = Wedge_Estator
Equation = 1
Material = 1
End
Body 15
Name = Aire_Estator
Equation = 1
Material = 1
End
Body 16
Name = cuerpo1
Equation = 1
Material = 1
Body Force = 1
End
Body 17
Name = AceroRotor
Equation = 1
Material = 2
Body Force = 1
End
Body 18
Name = AireRotor
Equation = 1
Material = 1
Body Force = 1
End
|----- SLOT DE STATOR-----
Body 19; Name = cuerpo2; Equation = 1; Material = 1; Body
Force = 1; End;
Body 20; Name = cuerpo3; Equation = 1; Material = 1; Body
Force = 1; End;
Body 21; Name = cuerpo4; Equation = 1; Material = 1; Body
Force = 1; End;
Body 22; Name = cuerpo5; Equation = 1; Material = 1; Body
Force = 1; End;
Body 23; Name = cuerpo6; Equation = 1; Material = 1; Body
Force = 1; End;
Body 24; Name = cuerpo7; Equation = 1; Material = 1; Body
Force = 1; End;
Body 25; Name = cuerpo8; Equation = 1; Material = 1; Body
Force = 1; End;
Body 26; Name = cuerpo9; Equation = 1; Material = 1; Body
Force = 1; End;
Body 27; Name = cuerpo10; Equation = 1; Material = 1;
Body Force = 1; End;
Equation 1 :: Active Solvers(2) = 2 3 4
Solver 1
Exec Solver = Before simulation

Equation = MeshDeform
Procedure = "RigidMeshMapper" "RigidMeshMapper"
End
Solver 2
Exec Solver = Always
Equation = MgDyn2D
Variable = A
Procedure = "MagnetoDynamics2D"
"MagnetoDynamics2D"
Nonlinear System Convergence Tolerance = 1.0e-16
Nonlinear System Max Iterations = 60
Linear System Solver = Iterative
Linear System Iterative Method = Bicgstabl
Linear System Symmetric = True
Linear System Max Iterations = 200
Linear System Preconditioning = ILU2
Linear System Convergence Tolerance = 1e-8
Linear System Residual Output = 50
End
Solver 3
Exec Solver = Always
Equation = CalcFields Potential
Variable = "A"
Procedure = "MagnetoDynamics"
"MagnetoDynamicsCalcFields"
Calculate Current Density = Logical True
Calculate Magnetic Vector Potential = Logical True
Linear System Solver = Iterative
Linear System Iterative Method = Bicgstabl
Linear System Symmetric = True
Linear System Max Iterations = 200
Linear System Preconditioning = ILU2
Linear System Convergence Tolerance = 1e-8
Linear System Residual Output = 50
End
Solver 4
Exec Solver = after simulation
Equation = "ResultOutput"
Procedure = "ResultOutputSolve" "ResultOutputSolver"
Output File Name = Estacionario
Output Directory = resultados
Save Geometry Ids = Logical True
Vtu Format = Logical True
End
Boundary Condition 1
Name = EstatorExterior
A = Real 0
End
Boundary Condition 2
Name = DerechaEstator
Mortar BC = Integer 3
Mortar BC Static = Logical True
Anti Radial Projector = Logical True
Galerkin Projector = Logical True
End
Boundary Condition 3
Name = IzquierdaEstator

```

**Figura A.2:** Estado estacionario 2

```
End
Boundary Condition 4
Name = CorredizoEstator
Mortar BC = Integer 8
Anti Rotational Projector = Logical True
Galerkin Projector = Logical True
End
!-----CONTORTO SEPARAR-----
Boundary Condition 5
Name = Separar
A = Real 0
End
Boundary Condition 6
Name = Corredizo_Rotor
End
Boundary Condition 7
Name = RotorInterno
A = Real 0
End
Boundary Condition 8
Name = RotorDerecho
Mortar BC = Integer 20
Mortar BC Static = Logical True
Anti Radial Projector = Logical True
Galerkin Projector = Logical True
End
Boundary Condition 9
Name = RotorIzquierdo
End
```

**Figura A.3:** Estado estacionario 3



## A.2. Código para simulación transitoria en el tiempo

```
1.2 Código para simulación transitoria en el tiempo
include parametros_transitoria.dat
Header
Mesh DB "MaquinaCompleta"
Results Directory "resultados"
Include Path "materiales"
End
Simulation
Max Output Level = 3
Coordinate System = Cartesian 2D
Coordinate Scaling = 0.001
Circuit Model Depth = Real $ I
Simulation Type = Transient
Timestepping Method = BDF
BDF Order = 2
Output Intervals = 1
Timestep Sizes = $ 1/f/40
Timestep Intervals = $ 1000
Use Mesh Names = Logical True
End
! Aire
Material 1
Relative Permeability = 1
Electric Conductivity = 0
End
! Acero
Material 2
Name = "Acero"
INCLUDE curva_acero
Electric Conductivity = 0
End
! Aluminio
Material 3
Relative Permeability = 1
Electric Conductivity = 24e6 ! [1/(Ohm*m)] rotor caliente
End
! Cobre
Material 4
Relative Permeability = 1
Electric Conductivity = 48e6
End
Body Force 1
Name = "Circuito"
A_a = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f)"
B_b = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f-pi/3)"
C_c = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f-2*pi/3)"
D_d = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f-pi)"
E_e = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f-4*pi/3)"
F_f = Variable time
Real MATC "440*sqrt(2)/sqrt(3)*sin(tx(0)*2*pi*f-5*pi/3)"
End
Body Force 7
Mesh Rotate 3 = Variable time
```

Figura A.4: Simulación transitoria en el tiempo 1

```

Real MATC "180/pi*2*pi*f/pp*(1-slip)*tx"
End
Body 1 Name = Cuerpo1
Equation = 1
Material = 3
Body Force = 7
Torque Groups = Integer 1
End
Body 2
Name = cuerpo2
Equation = 1
Material = 3
Body Force = 7
Torque Groups = Integer 1
End
Body 3
Name = cuerpo3
Equation = 1
Material = 3
Body Force = 7
Torque Groups = Integer 1
End
Body 4
Name = cuerpo4
Equation = 1
Material = 3
Body Force = 7
Torque Groups = Integer 1
End
Body 5
Name = cuerpo5
Equation = 1
Material = 3
Body Force = 7
Torque Groups = Integer 1
End
Body 6
Name = cuerpo6
Equation = 1
Material = 3
Body Force = 7
Torque Groups = Integer 1
End
Body 7
Name = cuerpo7
Equation = 1
Material = 3
Body Force = 7
Torque Groups = Integer 1
End
Body 8
Name = cuerpo8
Equation = 1
Material = 3
Body Force = 7
Torque Groups = Integer 1
End
Body 9

```

**Figura A.5:** Simulación transitoria en el tiempo 2

```
Name = cuerpo9
Equation = 1
Material = 3
Body Force = 7
Torque Groups = Integer 1
End
Body 10
Name = cuerpo10
Equation = 1
Material = 3
Body Force = 7
Torque Groups = Integer 1
End
Body 11
Name = A_plus
Equation = 1
Material = 4
End
Body 12
Name = C_minus
Equation = 1
Material = 4
End
Body 13
Name = B_plus
Equation = 1
Material = 4
End
Body 14
Name = D_minus
Equation = 1
Material = 4
End
Body 15
Name = F_plus
Equation = 1
Material = 4
End
Body 16
Name = E_minus
Equation = 1
Material = 4
End
Body 17
Name = AceroEstator
Equation = 1
Material = 2
End
Body 18
Name = Wedge_Estator
Equation = 1
Material = 1
End
Body 19
Name = Aire_Estator
Equation = 1
Material = 1
End
```

**Figura A.6:** Simulación transitoria en el tiempo 3

```

Body 20
Name = AceroRotor
Equation = 1
Material = 2
Body Force = 7
Torque Groups = Integer 1
End
Body 21
Name = AireRotor
Equation = 1
Material = 1
Body Force = 7
R Inner = Real 0.0620
R Outer = Real 0.0625
End
include "encapsulado/definiciones.encapsulado"
Component 11
Name = StringRB
Aplus Body = Integer 11
Coil Type = String Stranded
Number of Turns = Real $ Nph/2
Resistance = Real $ Rs
End
Component 12
Name = String Bplus
Body = Integer 12
Coil Type = String Stranded
Number of Turns = Real $ Nph/2
Resistance = Real $ Rs
End
Component 13
Name = String Cminus
Body = Integer 13
Coil Type = String Stranded
Number of Turns = Real $ Nph/2
Resistance = Real $ Rs
End
Component 14
Name = String Dplus
Body = Integer 14
Coil Type = String Stranded
Number of Turns = Real $ Nph/2
Resistance = Real $ Rs
End
Component 15
Name = String Eminus
Body = Integer 15
Coil Type = String Stranded
Number of Turns = Real $ Nph/2
Resistance = Real $ Rs
End
Component 16
Name = String Fplus
Body = Integer 16
Coil Type = String Stranded
Number of Turns = Real $ Nph/2
Resistance = Real $ Rs
End

```

**Figura A.7:** Simulación transitoria en el tiempo 4

```

Equation 1 :: Active Solvers(3) = 2 3 4
Solver 1
Exec Solver = Before timestep
Equation = MeshDeform
Procedure = "RigidMeshMapper" "RigidMeshMapper"
End
Solver 2
Exec Solver = Always
Equation = Circuits
Variable = X
Procedure = "CircuitsAndDynamics" "CircuitsAndDynamics"
No Matrix = Logical True
End
Solver 3
Exec Solver = Always
Equation = MgDyn2D
Variable = A
Procedure = "MagnetoDynamics2D" "MagnetoDynamics2D"
Stabilize = False
Partition Local Constraints = Logical True
Nonlinear System Compute Change in Scaled System = Logical True
Nonlinear System Convergence Measure = Residual
Nonlinear System Convergence Tolerance = 1e-6
Nonlinear System Max Iterations = 20
Nonlinear System Min Iterations = 1
Nonlinear System Relaxation Factor = 1.0
Nonlinear System Newton After Iterations = 7
Export Lagrange Multiplier = Logical True
Linear System Abort Not Converged = False
Linear System Solver = Iterative
Linear System Iterative Method = GCR
Linear System GCR Restart = 500
Bicgstabl Polynomial Degree = 4
Linear System Preconditioning = ILU4
Linear System Max Iterations = 1500
Linear System Residual Output = Integer 23
Linear System Convergence Tolerance = 1e-7
Mortar BCs Additive = Logical True
Linear System Symmetric = True
End
Solver 4
Exec Solver = Always
Equation = CalcFields
Potential Variable = "A"
Procedure = "MagnetoDynamics" "MagnetoDynamicsCalcFields"
Calculate Nodal Forces = Logical True
Calculate Magnetic Torque = Logical True
Calculate Current Density = Logical True
Calculate Magnetic Vector Potential = Logical True
End
Solver 5
Exec Solver = Always
Equation = CircOutput
Procedure = "CircuitsAndDynamics" "CircuitsOutput"
End
Solver 6
Exec Solver = after timestep
Equation = "ResultOutput"

```

**Figura A.8:** Simulación transitoria en el tiempo 5

```

Procedure = "ResultOutputSolve" "ResultOutputSolver"
Output File Name = estacionario
Output Directory = resultados
Save Geometry Ids = Logical True
Vtu Format = Logical True
Binary Output = Logical True
Single Precision = Logical True
End
Solver 7
Exec Solver = After timestep
Equation = scalars
Procedure = "SaveData" "SaveScalars"
Filename = resultados_transitoria.dat
Output Directory = resultados
End
Boundary Condition 1
Name = EstatorExterior
A = Real 0
End
Boundary Condition 2
Name = DerechaEstator
Mortar BC = Integer 3
Mortar BC Static = Logical True
Anti Radial Projector = Logical True
Galerkin Projector = Logical True
End
Boundary Condition 3
Name = IzquierdaEstator
End
Boundary Condition 4
Name = CorredizoEstator
Mortar BC = Integer 9
Anti Rotational Projector = Logical True
Galerkin Projector = Logical True
End
Boundary Condition 5
Name = Separar
End
Boundary Condition 6
Name = Rotor_Interno
A = Real 0
End
Boundary Condition 7
Name = Rotor_Derecho
Mortar BC = Integer 8
Mortar BC Static = Logical True
Anti Radial Projector = Logical True
Galerkin Projector = Logical True
End
Boundary Condition 8
Name = Rotor_Izquierdo
End
Boundary Condition 9
Name = Corredizo_Rotor
End

```

**Figura A.9:** Simulación transitoria en el tiempo 6

## A.3. Parámetros de la MI

```
1.3 Parámetros de la MI
$ slip = 0.03333
$ f = 60 ! [Hz] Frecuencia eléctrica
$ U = 440 ! [V] voltaje
$ pp = 2 ! número de pares de polos
$ n = 1740 ! [rpm] velocidad
$ l = 0.160 ! [m]
$ Rs = 0.94/2 ! [Ohm]
$ Nph = 256
$ w_syn = 1800/60*2*pi ! [rad/s]
$ L_ew = 0.0062 ! [H] inductancia de fase de devanado final
$ R_er = 0.0000016 ! [Ohm] Resistencia del rotor del anillo final de una sección entre 2 barras
$ L_er = 0.00000012 ! [H] inductancia del rotor del anillo final de una sección entre 2 barras
! Circuitos impulsados por voltaje del estator para 6 fases
$ Circuits = 7
! FASE A_plus
$ C.1.variables = 6
$ C.1.perm = zeros(C.1.variables)
$ C.1.A = zeros(C.1.variables, C.1.variables)
$ C.1.B = zeros(C.1.variables, C.1.variables)
$ C.1.Mre = zeros(C.1.variables, C.1.variables)
$ C.1.Mim = zeros(C.1.variables, C.1.variables)
$ C.1.name.1 = "corr_su"
$ C.1.name.2 = "volt_su"
$ C.1.name.3 = "corr_componente(11)"
$ C.1.name.4 = "volt_componente(11)"
$ C.1.name.5 = "corr_ewu"
$ C.1.name.6 = "volt_ewu"
$ C.1.B(0,1) = 1
$ C.1.source.1 = "A_a"
$ C.1.B(2,0) = -0.5
$ C.1.B(2,2) = 1
$ C.1.B(4,0) = -0.5
$ C.1.B(4,4) = 1
$ C.1.B(1,1) = -0.5
$ C.1.B(1,3) = 1
$ C.1.B(1,5) = 1
$ C.1.A(5,4) = L_ew/4
$ C.1.B(5,5) = -1
! FASE B_plus
$ C.2.variables = 6
$ C.2.perm = zeros(C.2.variables)
$ C.2.A = zeros(C.2.variables, C.2.variables)
$ C.2.B = zeros(C.2.variables, C.2.variables)
$ C.2.Mre = zeros(C.2.variables, C.2.variables)
$ C.2.Mim = zeros(C.2.variables, C.2.variables)
$ C.2.name.1 = "corr_sv"
$ C.2.name.2 = "volt_sv"
$ C.2.name.3 = "corr_componente(12)"
$ C.2.name.4 = "volt_componente(12)"
! $ C.2.name.5 = "corr_ewv"
$ C.2.name.6 = "volt_ewv"
$ C.2.B(0,1) = 1
$ C.2.source.1 = "B_b"
$ C.2.B(2,0) = -0.5
$ C.2.B(2,2) = 1
$ C.2.B(4,0) = -0.5
$ C.2.B(4,4) = 1
```

Figura A.10: Parámetros de la MI 1

```

$ C.2.B(1,1) = -0.5
$ C.2.B(1,3) = 1
$ C.2.B(1,5) = 1
$ C.2.A(5,4) = L_ew/4
$ C.2.B(5,5) = -1
! FASE C_minus
$ C.3.variables = 6
$ C.3.perm = zeros(C.3.variables)
$ C.3.A = zeros(C.3.variables, C.3.variables)
$ C.3.B = zeros(C.3.variables, C.3.variables)
$ C.3.Mre = zeros(C.3.variables, C.3.variables)
$ C.3.Mim = zeros(C.3.variables, C.3.variables)
$ C.3.name.1 = "corr_sw"
$ C.3.name.2 = "volt_sw"
$ C.3.name.3 = "corr_componente(13)"
$ C.3.name.4 = "volt_componente(13)"
$ C.3.name.5 = "corr_eww"
$ C.3.name.6 = "volt_eww"
$ C.3.B(0,1) = 1
$ C.3.source.1 = "C_c"
$ C.3.B(2,0) = -0.5
$ C.3.B(2,2) = 1
$ C.3.B(4,0) = -0.5
$ C.3.B(4,4) = 1
$ C.3.B(1,1) = -0.5
$ C.3.B(1,3) = 1
$ C.3.B(1,5) = 1
$ C.3.A(5,4) = L_ew/4
$ C.3.B(5,5) = -1
! FASE D_plus
$ C.4.variables = 6
$ C.4.perm = zeros(C.4.variables)
$ C.4.A = zeros(C.4.variables, C.4.variables)
$ C.4.B = zeros(C.4.variables, C.4.variables)
$ C.4.Mre = zeros(C.4.variables, C.4.variables)
$ C.4.Mim = zeros(C.4.variables, C.4.variables)
$ C.4.name.1 = "corr_sx"
$ C.4.name.2 = "volt_sx"
$ C.4.name.3 = "corr_componente(14)"
$ C.4.name.4 = "volt_componente(14)"
$ C.4.name.5 = "corr_ewx"
$ C.4.name.6 = "volt_ewx"
$ C.4.B(0,1) = 1
$ C.4.source.1 = "D_d"
$ C.4.B(2,0) = -0.5
$ C.4.B(2,2) = 1
$ C.4.B(4,0) = -0.5
$ C.4.B(4,4) = 1
$ C.4.B(1,1) = -0.5
$ C.4.B(1,3) = 1
$ C.4.B(1,5) = 1
$ C.4.A(5,4) = L_ew/4
$ C.4.B(5,5) = -1
! FASE E_minus
$ C.5.variables = 6
$ C.5.perm = zeros(C.5.variables)
$ C.5.A = zeros(C.5.variables, C.5.variables)
$ C.5.B = zeros(C.5.variables, C.5.variables)

```

**Figura A.11:** Parámetros de la MI 2



```

$ C.5.Mre = zeros(C.5.variables, C.5.variables)
$ C.5.Mim = zeros(C.5.variables, C.5.variables)
$ C.5.name.1 = "corr_sy"
$ C.5.name.2 = "volt_sy"
$ C.5.name.3 = "corr_component(15)"
$ C.5.name.4 = "volt_component(15)"
$ C.5.name.5 = "corr_ewy"
$ C.5.name.6 = "volt_ewy"
$ C.5.B(0,1) = 1
$ C.5.source.1 = "E_e"
$ C.5.B(2,0) = -0.5
$ C.5.B(2,2) = 1
$ C.5.B(4,0) = -0.5
$ C.5.B(4,4) = 1
$ C.5.B(1,1) = -0.5
$ C.5.B(1,3) = 1
$ C.5.B(1,5) = 1
$ C.5.A(5,4) = L_ew/4
$ C.5.B(5,5) = -1
!FASE F_plus
$ C.6.variables = 6
$ C.6.perm = zeros(C.6.variables)
$ C.6.A = zeros(C.6.variables, C.6.variables)
$ C.6.B = zeros(C.6.variables, C.6.variables)
$ C.6.Mre = zeros(C.6.variables, C.6.variables)
$ C.6.Mim = zeros(C.6.variables, C.6.variables)
$ C.6.name.1 = "corr_sz"
$ C.6.name.2 = "volt_sz"
$ C.6.name.3 = "corr_component(16)"
$ C.6.name.4 = "volt_component(16)"
$ C.6.name.5 = "corr_ewz"
$ C.6.name.6 = "volt_ewz"
$ C.6.B(0,1) = 1
$ C.6.source.1 = "F_f"
$ C.6.B(2,0) = -0.5
$ C.6.B(2,2) = 1
$ C.6.B(4,0) = -0.5
$ C.6.B(4,4) = 1
$ C.6.B(1,1) = -0.5
$ C.6.B(1,3) = 1
$ C.6.B(1,5) = 1
$ C.6.A(5,4) = L_ew/4
$ C.6.B(5,5) = -1

```

**Figura A.12:** Parámetros de la MI 3

# Bibliografía

- [1] J. Rodas Benítez, “Análisis y evaluación del funcionamiento del control predictivo de corriente en una máquina de 6 fases con doble devanado trifásico independiente y asimétrico utilizando estimadores de orden reducido,” in *Trabajo de Fín de Máster para optar al grado de Máster en Electrónica, Tratamiento de Señal y Comunicaciones por la Universidad de Sevilla - España*, vol. 1, Octubre 2013, pp. 29–54.
- [2] M. A. Gary Echague and J. Rodas, “Design, analysis and validation of a six-phase induction machine from a commercial three-phase for academic research,” *IEEE latin america transactions*, vol. 18, 2020.
- [3] J. M. Aller, “Máquinas eléctricas rotativas: Introducción a la teoría general,” *Editorial Equinoccio*, 2006.
- [4] P. Á. Lloret, “Caracterización del software libre de elementos finitos elmer en aplicaciones de ingeniería mecánica,” in *Trabajo de Fín de Máster para optar al grado de Máster en Métodos Numéricos para Diseño en Ingeniería por la Universidad Politécnica Salesiana - Ecuador*, vol. 1, Sep 2010, pp. 2–10.
- [5] U. Ayachit, *The ParaView Guide: A Parallel Visualization Application*. Clifton Park, NY, USA: Kitware, Inc., 2020.
- [6] P. Ponomarev, “Elmer fem induction machine tutorial,” *VTT Technical Research Centre of Finland LTD*, 2017.
- [7] M. K. Pinjala and R. Bhimasingu, “Improving the dc-link utilization of nine-switch boost inverter suitable for six-phase induction motor,” *IEEE Transactions on Transportation Electrification*, vol. 6, no. 3, pp. 1177–1187, 2020.
- [8] R. Karampuri, J. Prieto, F. Barrero, and S. Jain, “Extension of the dtc technique to multiphase induction motor drives using any odd number of phases,” in *2014 IEEE Vehicle Power and Propulsion Conference (VPPC)*, 2014, pp. 1–6.
- [9] M. I. Abdelwanis and F. Selim, “A sensorless six-phase induction motor driving a centrifugal pump system,” in *2017 Nineteenth International Middle East Power Systems Conference (MEPCON)*, 2017, pp. 242–247.
- [10] A. Vikram and V. Agarwal, “Asic based single-phase to six-phase conversion,” in *2010 International Conference on Power, Control and Embedded Systems*, 2010, pp. 1–4.
- [11] R. Hammad, S. M. Dabour, and E. M. Rashad, “Open-circuit fault detection of asymmetrical six-phase induction motor fed from z-source inverter,” in *2019 21st International Middle East Power Systems Conference (MEPCON)*, 2019, pp. 1216–1222.
- [12] A. M. Trzynadlowski, *The Field Orientation Principle in Control of Induction Motors*. Springer US, 1994.

- [13] P. C. Krause, O. Wasynczuk, S. D. Sudhoff, and S. Pekarek, *Analysis of electric machinery and drive systems*. Wiley Online Library, 2002, vol. 2.
- [14] J. M. Aller, “Máquinas eléctricas rotativas,” *Caracas, Equinoccio*, 2008.
- [15] S. Liu, Y. Cheng, and L. Li, “A voltage-behind-reactance model and validation for six-phase synchronous machines,” in *2014 IEEE Conference and Expo Transportation Electrification Asia-Pacific (ITEC Asia-Pacific)*. IEEE, aug 2014.
- [16] J. Aller, J. Viola, J. Restrepo, M. Fajardo, A. Ginart, J. Rengifo, J. D. Loja, and J. D. Ochoa, “Space vector control of asymmetrical single-phase induction motors,” in *2018 XIII International Conference on Electrical Machines (ICEM)*. IEEE, sep 2018.
- [17] J. M. Aller, J. A. Restrepo, J. C. Viola, and J. R. Mayor, “Simplified voltage behind reactance model for the six-phase open-end salient pole pmsm,” in *2020 International Conference on Electrical Machines (ICEM)*, vol. 1, Aug 2020, pp. 2216–2221.
- [18] B. S. Abdel-Mageed, S. Mohamed S.A., K. F. Ali, and M. A. Morsy, “Coupled finite element-circuit modelling and parameter estimation of six-phase induction motor,” in *2019 21st International Middle East Power Systems Conference (MEPCON)*, 2019, pp. 645–649.
- [19] D. G. Dorrell, C. Y. Leong, and R. A. McMahon, “Analysis and performance assessment of six-pulse inverter-fed three-phase and six-phase induction machines,” *IEEE Transactions on Industry Applications*, vol. 42, no. 6, pp. 1487–1495, 2006.
- [20] B. C. Ghosh, M. Habibullah, and M. E. Ali, “Performance comparison of five and six phase induction motors operating under normal and faulty conditions,” in *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, 2019, pp. 1–6.
- [21] M. F. Khan and R. Khan, “Modeling and analysis of a six-phase self excited induction generator feeding induction motors,” *IEEE Transactions on Energy Conversion*, pp. 1–1, 2020.
- [22] M. Y. Mohamed, M. Fawzi, S. A. A. Maksoud, and A. E. Kalas, “Finite element analysis of multi-phase squirrel cage induction motor to develop the optimum torque,” in *2019 IEEE Conference on Power Electronics and Renewable Energy (CPERE)*, 2019, pp. 504–510.
- [23] L. Zhao, Q. Ge, K. Wang, and Y. Li, “Research on the control technology of six-phase induction motor based with the novel flux observer speed sensorless,” in *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, 2019, pp. 1–4.
- [24] M. Lyly, “Elmergui manual,” *CSC-IT Center for Science*, 2015.
- [25] M. M. Peter Raback, “Overview of elmer,” *CSC-IT Center for Science*, 2019.
- [26] P. Ponomarev, “Fem modeling of pmsms using elmer,” 2014.
- [27] A. Villarejo-Galende and A. Herrero-San Martín, “Nikola tesla: relámpagos de inspiración,” *Rev Neurol*, vol. 56, pp. 109–14, 2013.

- [28] S. Suresh and R. P. P., “Virtual space vector-based direct torque control schemes for induction motor drives,” *IEEE Transactions on Industry Applications*, vol. 56, no. 3, pp. 2719–2728, 2020.
- [29] T. S. de Souza, R. R. Bastos, and B. J. Cardoso Filho, “Synchronous-frame modeling and dq current control of an unbalanced nine-phase induction motor due to open phases,” *IEEE Transactions on Industry Applications*, vol. 56, no. 2, pp. 2097–2106, 2020.
- [30] S. Sharma, M. V. Aware, and A. Bhowate, “Symmetrical six-phase induction motor-based integrated driveline of electric vehicle with predictive control,” *IEEE Transactions on Transportation Electrification*, vol. 6, no. 2, pp. 635–646, 2020.
- [31] H. R. H. Bhag S. Guru, “Máquinas eléctricas y transformadores,” *New York, Oxford University Press México*, 2003.
- [32] P. A. G. Espinoza, “Análisis, diseño y simulación de comportamiento mecánico de un motor asíncrono trifásico, utilizando el método de elementos finitos,” *Universidad politécnica salesiana*, 2018.
- [33] C. Geuzaine and J.-F. Remacle, “Gmsh,” in *Universidad de Liegia and Universidad católica de Louvain*, vol. 1, Abril 2021, pp. 2–57.