

**LA COMPUTACIÓN EVOLUCIONARIA EN LA OPTIMIZACIÓN DEL DISEÑO  
MECÁNICO: APLICACIÓN DE ALGORITMOS DE OPTIMIZACIÓN  
MULTIOBJETIVO (MOEA) EN EL DISEÑO DE UN MOTOR WANKEL PARA  
MODELISMO**

**ANDRÉS ALONSO CASTILLO CASTELLANOS**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNICA  
BUCARAMANGA  
2.008**

**LA COMPUTACIÓN EVOLUCIONARIA EN LA OPTIMIZACIÓN DEL DISEÑO  
MECÁNICO: APLICACIÓN DE ALGORITMOS DE OPTIMIZACIÓN  
MULTIOBJETIVO (MOEA) EN EL DISEÑO DE UN MOTOR WANKEL PARA  
MODELISMO**

**ANDRÉS ALONSO CASTILLO CASTELLANOS**

**Trabajo de grado para optar al título de  
Ingeniero Mecánico**

**Director**

**JORGE LUIS CHACÓN  
Ingeniero Mecánico**

**Codirector**

**JORGE ENRIQUE MENESES  
Ingeniero Mecánico**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO – MECÁNICAS  
ESCUELA DE INGENIERÍA MECÁNICA  
BUCARAMANGA**

**2.008**

## RESUMEN

**TÍTULO:** APLICACIÓN DE ALGORITMOS DE OPTIMIZACIÓN MULTI OBJETIVO (MOEA) EN EL DISEÑO DE UN MOTOR WANKEL PARA MODELISMO\*

**AUTOR:** ANDRÉS ALONSO CASTILLO CASTELLANOS\*\*

### **PALABRAS CLAVE:**

Diseño Óptimo, Algoritmo Genético, Motor Wankel, NSGA-II, Sensibilidad de Parámetros, Análisis Termoelástico.

### **RESUMEN:**

En el presente trabajo se estudia la utilización de algoritmos genéticos multiobjetivo (MOOGA) en el proceso de optimización del diseño térmico y mecánico de un motor de combustión interna rotativo tipo Wankel para ser empleado en aeromodelos.

Para evaluar la calidad de las soluciones obtenidas se emplea un análisis termodinámico de primera ley, empleando un modelo de combustión cero-dimensional de dos zonas y una velocidad de aporte de calor empírica. Este modelo se complementa con un análisis multifísico que combina cargas térmicas y mecánicas (análisis termoelástico) y las evalúa mediante el método de elementos finitos con ayuda de un paquete de software FEM.

Para obtener el conjunto de soluciones más óptimas, o frente de Pareto, se emplea el algoritmo genético con sorteo no-dominado (NSGA-II) propuesto por (Deb, Pratap, Agarwal, & Meyarivan, 2002) y empleando un sistema de representación combinado compuesto por parámetros reales y parámetros con codificación binaria.

Para efectos del proceso de simulación, se elaboró un conjunto de funciones y rutinas programadas para el ambiente numérico MATLAB (Versión 7.4 – R2007a) y el análisis termoelástico programado para ejecución por lotes en entorno ANSYS-BATCH, que permiten al lector evaluar el desempeño del motor Wankel y obtener soluciones numéricas del problema de transferencia de calor en la carcasa.

---

\* Proyecto de Grado

\*\* Facultad de Ingenierías Fisicomecánicas, Escuela de Ingeniería Mecánica. Director: Jorge Luis Chacón.

## ABSTRACT

**TITLE:** IMPLEMENTATION OF MULTIOBJECTIVE OPTIMIZATION ALGORITHMS (MOEA) IN THE DESIGN OF A WANKEL ENGINE FOR AIRCRAFT MODELING\*

**AUTHOR:** ANDRÉS ALONSO CASTILLO CASTELLANOS \*\*

### KEYWORDS:

Optimum Design, Genetic Algorithm, Wankel engine, NSGA-II, Thermoelastic Analysis, Sensitivity Analysis.

### ABSTRACT:

This work studies the implementation of multiobjective genetic algorithms (MOOGA) in the optimization process of the thermal and mechanical design applied to a Wankel rotary internal combustion engine, to be used in aircraft modeling.

The assessment of the quality of the solutions obtained was carried out using a first-law of thermodynamics analysis, in conjunction with a zero-dimensional two-zone combustion model and an empiric mass burning rate. This model is complemented with a multi-physics analysis that combines mechanical and thermal loads (Thermoelastic Analysis), and solved using the finite elements method with help of a FEM software package.

In order to obtain the non-dominated solutions set, or Pareto Front, the non-dominated sorting genetic algorithm (NSGA-II) proposed by (Deb, Pratap, Agarwal, & Meyarivan, 2002) is used, with a combined representation system composed of real parameters and binary encoded parameters.

For the sake of the simulation process, a toolbox of functions and programmatic sub-routines were developed for the MATLAB numerical environment (Version 7.4 – R2007a), while the thermoelastic analysis was programmed and deployed for batch execution in ANSYS-BATCH, which allows the reader or potential users to evaluate the performance of the Wankel engine and obtain numerical solutions to the studied heat transfer problem in the shell.

---

\* Undergraduate Degree Work

\*\* Physical-Mechanical Engineering Faculty, Department of Mechanical Engineering. Director: Jorge Luis Chacón

*A mis papás.*

*A mis hermanos.*

*A mis amigos.*

*'Si dejas obra,  
muere tranquilo,  
confiando en unos pocos  
buenos amigos.'*

*Andrés Caicedo, 1975*

## **AGRADECIMIENTOS**

A PhD. Jorge Luis Chacón, director del GIEMA y director del proyecto aportar su visión, por la confianza depositada en el proyecto y su incondicional apoyo.

Al profesor Jorge Enrique Meneses, codirector del proyecto por dar un rumbo al proyecto, por la colaboración brindada y promover esa necesidad de leer e investigar.

A la Universidad Industrial de Santander, centro cultural de la región y espacio de formación personal y profesional, y a los docentes que comprometidos con la academia promueven la inventiva, creatividad y el amor por la carrera.

A Raúl Fernando Castro, Diego Rivera, Yoediver Meneses por la colaboración brindada.

A mis papás y mis hermanos por brindarme su incondicional apoyo.

## CONTENIDO

<b>CONTENIDO</b> .....	<b>9</b>
<b>LISTADO DE FIGURAS</b> .....	<b>12</b>
<b>LISTADO DE TABLAS</b> .....	<b>14</b>
<b>DESCRIPCIÓN DEL DOCUMENTO</b> .....	<b>15</b>
<b>1. INTRODUCCIÓN</b> .....	<b>17</b>
1.1. ANTECEDENTES .....	19
1.2. ALCANCE DEL PROYECTO .....	20
1.2.1. <i>Objetivos Generales</i> .....	20
1.2.2. <i>Objetivos Específicos</i> .....	20
<b>2. MODELO MATEMATICO DEL MOTOR ROTATIVO</b> .....	<b>22</b>
2.1. VENTAJAS Y DESVENTAJAS DEL MOTOR ROTATIVO .....	24
2.2. PARTES CONSTITUTIVAS DE UN MOTOR ROTATIVO TIPO WANKEL.....	24
2.2.1. <i>Conjunto Rotor</i> .....	25
2.2.2. <i>Estator Periférico</i> .....	26
2.2.3. <i>Estator Lateral</i> .....	27
2.3. GEOMETRÍA DEL MOTOR WANKEL .....	27
2.3.1. <i>Forma General de la Carcasa</i> .....	28
2.3.2. <i>Forma General del Rotor</i> .....	31
2.3.3. <i>Ángulo de Oscilación del Sello</i> .....	32
2.3.4. <i>Relación de Compresión</i> .....	33
2.3.5. <i>Engranajes de Fase</i> .....	35
2.4. EFICIENCIA TÉRMICA – MODELO TERMODINAMICO .....	37
2.4.1. <i>Consideraciones y Simplificaciones</i> .....	37
2.4.2. <i>Modelo de Compresión-Expansión</i> .....	39
2.4.3. <i>Ley de Liberación de Calor</i> .....	41
2.4.4. <i>Resultados del modelo de análisis termodinámico</i> .....	45
2.5. DISEÑO DE LA CARCASA Y EL SISTEMA DE REFRIGERACIÓN. ....	50
2.5.1. <i>Análisis Termoelástico</i> .....	52
2.5.2. <i>Selección del elemento PLANE223</i> .....	55
2.5.3. <i>Perfil de Temperaturas y Estado de Esfuerzos</i> .....	57
2.5.4. <i>Diseño de las Aletas Radiales</i> .....	61

2.6. DEFINICIÓN DE LA FUNCIÓN PESO .....	65
2.6.1. Dimensiones del Rodamiento del Rotor .....	66
2.6.2. Dimensiones del engranaje interno .....	68
2.7. REPRESENTACIÓN DE LAS FUNCIONES OBJETIVO .....	69
<b>3. ALGORITMO DE OPTIMIZACIÓN .....</b>	<b>71</b>
3.1. INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS.....	72
3.1.1. Principio de Funcionamiento .....	72
3.1.2. Funcionalidad del Algoritmo Genético .....	73
3.2. ALGORITMOS DE OPTIMIZACIÓN MULTI-OBJETIVO (MOO) .....	76
3.3. NON-DOMINATED SORTING GENETIC ALGORITHM – NSGA-II .....	80
3.3.1. Estructura De Datos .....	81
3.3.2. Operadores de Cruce ó Recombinación .....	84
3.3.3. Operador De Mutación .....	87
3.3.4. Mecanismo De Selección .....	88
3.3.5. Función de Penalización .....	90
3.4. DOMINIO DE LA FUNCIÓN APTITUD .....	92
3.4.1. Dominio de las Variables Continuas .....	92
3.4.2. Dominio de las Variables Discretas .....	93
3.4.3. Conjunto de Restricciones .....	94
<b>4. ANÁLISIS DE RESULTADOS .....</b>	<b>96</b>
<b>5. CONCLUSIONES.....</b>	<b>110</b>
<b>6. RECOMENDACIONES .....</b>	<b>112</b>
<b>BIBLIOGRAFÍA.....</b>	<b>114</b>
<b>APÉNDICES.....</b>	<b>117</b>
A.1. ALGORITMO DE OPTIMIZACIÓN – (NSGA-II-WNKL.M) .....	117
- Inicializar Población (+GA/inicializar_poblacion.m) .....	117
- Ordenamiento y Distancia de Separación (fast_sorting_algorithm.m) ..	118
- Selección por Torneo (+GA/seleccion_torneo.m) .....	121
- Operadores Genéticos (+GA/genetic_operator.m) .....	122
- Mecanismo de Selección (+GA/crowded_comparison_operator.m) .....	125
- Función de Penalización (+GA/penalty.m) .....	125
- Función Objetivo (+GA/funcion_objetivo.m) .....	127
- Sub-Rutinas empleadas por la Función Objetivo.....	130

A.2. IMPLEMENTACIÓN EN ANSYS-BATCH ( <i>FROID.INP</i> ) .....	136
- <i>Implementación en Linux (GLNX86 &amp; GLNXA64)</i> .....	138
- <i>Ejecución del Análisis Termoelástico (/thermoelastic_fem.m)</i> .....	139
- <i>Lectura y Obtención de Parámetros</i> .....	140
- <i>Generación de la Geometría</i> .....	141
- <i>Enmallado</i> .....	143
- <i>Restricciones y Condiciones de Frontera</i> .....	144
- <i>Solución, Post-Proceso y Resultados</i> .....	146
A.3. SIMULACIÓN DEL CICLO EN MATLAB ( <i>SIMT.M</i> ) .....	150
- <i>Información Termodinámica Básica (airdata.m &amp; fueldata.m)</i> .....	150
- <i>Fuel, Air &amp; Residual Gases (farg.m)</i> .....	150
- <i>Productos de la Combustión (ecp.m)</i> .....	152
- <i>Calculo de la Temperatura Adiabática (Tadiabatic.m)</i> .....	156
- <i>Etapas de Funcionamiento del Motor</i> .....	157
- <i>Rutina Principal (SimT.m)</i> .....	160
A.4. SOPORTE TEÓRICO COMPLEMENTARIO .....	163
- <i>Aleta Longitudinal de Perfil Trapecial (Fernández, 2002)</i> .....	163
- <i>Distribución de los Pernos en las Tapas Laterales (epic.m)</i> .....	165
- <i>Diseño del Engranaje Interno</i> .....	167

## LISTADO DE FIGURAS

<b>Figura 1.</b> Metodología Propuesta .....	18
<b>Figura 2.</b> Representación MIMO de la evaluación de aptitud. ....	22
<b>Figura 3.</b> Motor Mazda 13B, Imagen tomada por Sfoskett bajo licencia GFDL.....	23
<b>Figura 4.</b> Vista Explosionada del Motor Rotativo .....	25
<b>Figura 5.</b> Elementos del Conjunto Rotor .....	26
<b>Figura 6.</b> Conjunto Estator-Rotor .....	26
<b>Figura 7.</b> Estator Lateral con engranaje estacionario .....	27
<b>Figura 8.</b> Geometría característica del conjunto Estator-Rotor .....	28
<b>Figura 9.</b> Generación de la peritrocoide .....	29
<b>Figura 10.</b> Forma general de la carcasa para diferentes valores de “ <b>K</b> ” .....	30
<b>Figura 11.</b> Forma general del rotor para diferentes valores de <b>K</b> .....	32
<b>Figura 12.</b> Ángulos de oscilación y diagrama de cuerpo libre en el sello .....	33
<b>Figura 13.</b> Ángulo de Oscilación en función de “ <b>θ</b> ” y “ <b>K</b> ” .....	33
<b>Figura 14.</b> Volúmenes Máximo y Mínimo de la cámara de combustión .....	34
<b>Figura 15.</b> Procesos que interactúan en el motor .....	37
<b>Figura 16.</b> Primera Ley aplicada al volumen de control .....	39
<b>Figura 17.</b> Relación de entradas y salidas para el análisis termodinámico .....	46
<b>Figura 18.</b> Trabajo vs. Ángulo del Cigüeñal .....	46
<b>Figura 19.</b> Temperatura de los gases vs. Ángulo del cigüeñal .....	48
<b>Figura 20.</b> Resultados de la rutina .....	49
<b>Figura 21.</b> Distribución de presiones y temperaturas en la carcasa .....	50
<b>Figura 22.</b> Motor 49PI tipo II de la O.S [osengines.com] .....	50
<b>Figura 23.</b> Sistema de refrigeración propuesto.....	51
<b>Figura 24.</b> Elementos que interactúan en el análisis termoelástico .....	52
<b>Figura 25.</b> Elemento PLANE13, imagen tomada de (ANSYS, Inc., 2005) .....	56
<b>Figura 26.</b> Elemento PLANE223 imagen tomada de (ANSYS, Inc., 2005) .....	56
<b>Figura 27.</b> Distribución de temperaturas en la carcasa, datos en grados Kelvin...59	
<b>Figura 28.</b> Esfuerzo equivalente en la carcasa, datos en Pascales .....	60
<b>Figura 29.</b> Temperatura en la base de las aletas .....	61
<b>Figura 30.</b> Longitud de las aletas para diferentes escenarios propuestos .....	63
<b>Figura 31.</b> Distribución de las aletas a lo largo de la carcasa .....	64
<b>Figura 32.</b> Validación del diseño, perfil de temperaturas. ....	64
<b>Figura 33.</b> Forma simplificada del rotor. Dimensiones del Rodamiento del Rotor.66	
<b>Figura 34.</b> Dimensiones del rodamiento de agujas.....	68

<b>Figura 35.</b> Función Objetivo 1 – Eficiencia Térmica .....	69
<b>Figura 36.</b> Diagrama I/O para la Función Objetivo 1 .....	70
<b>Figura 37.</b> Función Objetivo 2 – Masa del Conjunto Rotor-Estator .....	70
<b>Figura 38.</b> Mecanismo de Búsqueda Genética.....	73
<b>Figura 39.</b> Representación esquemática de un AG típico .....	75
<b>Figura 40.</b> Frente de Pareto para el ejemplo 1 .....	78
<b>Figura 41.</b> Esquema de Funcionamiento Propuesto para el Algoritmo NSGA-II...	80
<b>Figura 42.</b> Representación binaria de una característica .....	83
<b>Figura 43.</b> Ordenamiento por Predominancia.....	89
<b>Figura 44.</b> Población Inicial – Tamaño de la Población 150 .....	98
<b>Figura 45.</b> 6 <sup>a</sup> Generación – Tamaño de la Población 150 .....	99
<b>Figura 46.</b> 12 <sup>a</sup> Generación – Tamaño de la Población 150 .....	99
<b>Figura 47.</b> 18 <sup>a</sup> Generación – Tamaño de la Población 150 .....	100
<b>Figura 48.</b> 24 <sup>a</sup> Generación – Tamaño de la Población 150 .....	101
<b>Figura 49.</b> 30 <sup>a</sup> Generación – Tamaño de la Población 150 .....	101
<b>Figura 50.</b> Variación de la función (1) a lo largo de 36 generaciones .....	103
<b>Figura 51.</b> Variación de la función (2) a lo largo de 36 generaciones .....	103
<b>Figura 52.</b> Comparación de Resultados .....	104
<b>Figura 53.</b> Grafica de R versus Función Masa versus Eficiencia Térmica .....	107
<b>Figura 54.</b> Grafica de K versus Función Masa versus Eficiencia Térmica .....	107
<b>Figura 55.</b> Grafica de NA versus Función Masa versus Eficiencia Térmica.....	108
<b>Figura 56.</b> Grafica del ángulo de encendido versus Funciones Objetivo .....	108
<b>Figura 57.</b> Grafica de la relación b/e versus Funciones Objetivo.....	109
<b>Figura 58.</b> Modificar variables del sistema .....	136
<b>Figura 59.</b> Generación de la Geometría .....	142
<b>Figura 60.</b> Enmallado por Barrido .....	143
<b>Figura 61.</b> Restricciones en los pernos y Presión al interior de la carcasa .....	144
<b>Figura 62.</b> Transferencia de Calor por Convección. ....	145
<b>Figura 63.</b> Generación de la Geometría para el modelo de verificación .....	148
<b>Figura 64.</b> Condiciones de frontera para el modelo de verificación. ....	149
<b>Figura 65.</b> Aleta Recta de Perfil Trapecial.....	163
<b>Figura 66.</b> Factor de Geometría J .....	168

## LISTADO DE TABLAS

<b>Tabla 1.</b> Valores de la excentricidad propuestos. ....	36
<b>Tabla 2.</b> Propiedades Térmicas y Mecánicas de las aleaciones contempladas. ....	57
<b>Tabla 3.</b> Rodamientos contemplados para el rotor, tabla tomada de SKF .....	67
<b>Tabla 4.</b> Ejemplo del Operador de Recombinación .....	86
<b>Tabla 5.</b> Ejemplo del operador de mutación .....	88
<b>Tabla 6.</b> Aptitud media de la población a lo largo de 36 generaciones .....	102
<b>Tabla 7.</b> Individuos del Frente de Pareto .....	106
<b>Tabla 8.</b> Dimensiones de los Pernos .....	166

## DESCRIPCIÓN DEL DOCUMENTO

Este trabajo de grado se desarrolló con la intención de incentivar el desarrollo de la investigación en el campo de los motores, no solo como un ejercicio académico sino que a futuro permita entender mejor las relaciones existentes entre los parámetros y transformar este conocimiento en soluciones al problema que rodea a estos sistemas de generación de potencia.

A su vez es la continuación de una serie de proyectos orientados a la investigación y estudio del **Proceso de Optimización del Diseño Mecánico**; incorporando avances recientes en el campo de los algoritmos genéticos y evolucionarios (GAE). En este informe se pretende mostrar los resultados al incorporar estos elementos de una manera práctica al diseño mecánico. Para nuestro caso en particular se busca incorporar esta metodología al diseño de una máquina térmica rotativa tipo Wankel destinada a ser utilizada en aeromodelos.

En este informe se profundiza en el proceso de optimización del diseño mecánico con criterios u objetivos múltiples que a menudo compiten entre sí. Tanto el procedimiento como los conceptos básicos de su aplicación al diseño óptimo de máquinas son objeto de estudio por parte del autor, a una máquina de una complejidad razonable.

Para facilitar la lectura del documento se ha decidido presentar en este los aportes realizados por el autor y los fundamentos teóricos presentarlos aparte en los anexos, de modo que el lector pueda profundizar en el tema si resulta de su interés.

El documento se ha decidido presentarlo en cuatro capítulos:

**Capítulo 1:** En este capítulo se determinan los alcances del proyecto y se da una breve inducción al lector para que se entere de los aspectos más relevantes y de los objetivos planteados en el desarrollo de este trabajo.

**Capítulo 2:** En esta sección se presentan los fundamentos teóricos y el desarrollo de los modelos matemáticos que permitan caracterizar el problema, en función de los dos criterios a evaluar como son *la eficiencia térmica y el peso*.

**Capítulo 3:** En este capítulo se presenta al lector con algún grado de familiaridad con el tema, el resultado del estudio bibliográfico de los temas desarrollados y el desarrollo teórico que llevo a la selección del método empleado en la optimización del diseño, y se muestran las herramientas requeridas para la implementación de la optimización.

**Capítulo 4:** Se presentan los datos obtenidos y se hace un análisis de los resultados obtenidos.

Finalmente el autor plantea las conclusiones del presente trabajo y hace las respectivas recomendaciones para trabajos futuros.

Los capítulos que le siguen permiten al lector un acceso fácil a las bases teóricas sobre las cuales se fundamenta el proyecto sin entorpecer al lector que se encuentre familiarizado con los tópicos respectivos.

**Apéndice 1:** Se presenta la implementación del algoritmo de optimización NSGA-II, una breve descripción de su funcionamiento y el código en MATLAB, de modo que facilite al lector su uso y su posterior modificación.

**Apéndice 2:** En este capítulo se encuentra en detalles la implementación en APDL del análisis termoelástico en ANSYS-BATCH.

**Apéndice 3:** En este capítulo se encuentran las rutinas de la simulación del ciclo termodinámico en detalle programadas en MATLAB.

**Apéndice 4:** Por último se presentan algunos de los conceptos teóricos fundamentales que fueron aplicados de manera complementaria.

## 1. INTRODUCCIÓN

El diseño mecánico es en última instancia producto de la toma de decisiones por parte del diseñador, quien es el encargado de elegir la mejor de las soluciones planteadas e inclusive proponer mejoras, luego depende en gran medida de la experiencia del diseñador. Este no cuenta con los parámetros o la capacidad suficiente para asegurar que la propuesta seleccionada sea la óptima, ya que le es físicamente imposible evaluar todo el espectro de soluciones posibles.

Luego surge la necesidad de emplear una herramienta que permita obtener soluciones de diseño que se acerquen, en la medida de lo posible, a una solución óptima y cuyo desempeño no dependa de la experiencia del diseñador.

En general cualquier labor de optimización significa hallar la mejor solución a través de una búsqueda en el espacio de soluciones posibles. Para espacios de relativa simplicidad es suficiente la exploración con los métodos exhaustivos clásicos, pero espacios más grandes requieren de la utilización de una herramienta computacional robusta. La capacidad combinada de exploración y explotación del espacio de búsqueda hace de la incorporación de los algoritmos genéticos y evolucionarios (GEA) al diseño mecánico, una alternativa viable para el desarrollo de una herramienta evolutiva.

Para la optimización del diseño mecánico, se tiene un espacio de búsqueda bastante complejo debido a la naturaleza de su función objetivo. La función objetivo es la expresión matemática de los criterios de optimalidad en función de las variables de diseño. Esta función viene acompañada por una serie de restricciones que condicionan el espacio de búsqueda para las variables de diseño como por ejemplo las características de los materiales o condiciones de funcionamiento.

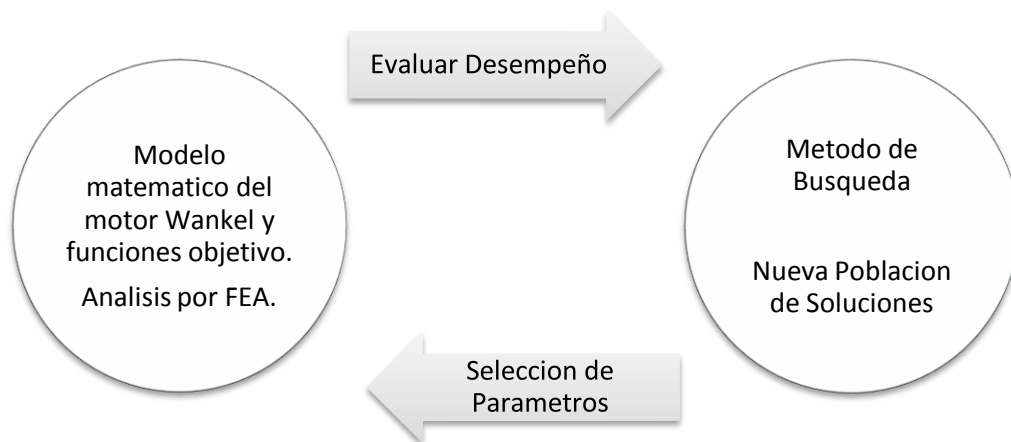
La **optimización del diseño mecánico** tiene importancia en la medida que consiga:

- Una reducción en los costos de fabricación, bien sea minimizando el consumo de materia prima o mejorando la eficiencia global de los procesos y por ende el aprovechamiento energético.
- Aporte mejoras en la funcionalidad del producto, como mayor capacidad, menor ruido, menor mantenimiento, etc.
- Reducir el tiempo de desarrollo de equipos nuevos.
- Mejorar el flujo de trabajo en el proceso de fabricación, con la utilización de procesos cada vez más sencillos.

Se ha escogido el motor Wankel por poseer un diseño innovador que ha sido probado en numerosas ocasiones con resultados satisfactorios, además de poseer un número de piezas significativamente menor a los de un motor alternativo, poseen un alto *froid.inp*, una excelente relación de peso/potencia y tener un funcionamiento suave.

Para este fin buscamos una integración de los métodos de búsqueda por algoritmo genético (GA) con el análisis por elementos finitos (FEM), siendo necesaria la implementación en el lenguaje paramétrica APDL propietario de ANSYS. Luego algunas de las evaluaciones de la función objetivo serán efectuadas de manera analítica y el resto de manera numérica. En la Figura 1 se muestra una representación esquemática de la metodología propuesta.

**Figura 1.** Metodología Propuesta



Además buscamos una mayor comprensión de la forma como se relacionan los distintos parámetros en el interior del motor rotativo, de una manera que resulte sencilla y conveniente y que además nos permita explorar soluciones que no hayan sido contempladas con anterioridad por el diseñador. Los criterios de optimalidad aplicados al proceso de toma de decisiones son la confiabilidad, el peso y el rendimiento del combustible, por ser destinado a una rama de la aeronáutica, aunque para este trabajo de grado en cuestión solo contemplamos la eficiencia térmica y el peso del conjunto rotor-estator.

### **1.1. ANTECEDENTES**

Los estudios que dieron origen a esta rama de investigación en la Universidad Industrial de Santander a mediados de los años 90, cubren de una manera muy completa los aspectos básicos de la metodología del diseño óptimo y su implementación a diversos problemas que se presentan en la ingeniería. Y si bien son un excelente punto de partida para el lector, al ser un sujeto de estudio ligado a las ciencias de la computación estos trabajos requieren de cierto grado de continuidad y actualización periódica. Con este trabajo de grado se busca reiniciar la línea de estudios en diseño óptimo y servir de plataforma para nuevos trabajos de grado relacionados.

El uso de las técnicas de computación evolutiva, tales como los GEA, se han aplicado a la solución de problemas en el pasado, (Goldberg, 1989) aplico esta técnica inicialmente al diseño de armaduras planas, combinando elementos tanto de los algoritmos genéticos como del análisis por elementos finitos (FEM). Su objetivo fue disminuir la sección transversal de los miembros de la estructura de modo que pudiese reducir el consumo de materiales y el peso, con la restricción de un esfuerzo máximo, de lo cual obtuvo resultados satisfactorios.

En los últimos 20 años el interés en los métodos de búsqueda heurísticos para optimización ha aumentado de una manera significativa, en particular en lo que respecta a una de las vertientes más prolíficas en el campo de la optimización como son los **algoritmos genéticos y evolucionarios** (GEA). En los últimos 10 años se han

desarrollado diferentes perspectivas teóricas que nos permiten entender de una manera mejor los algoritmos genéticos, aunque si bien estas ideas se encuentran dispersas en la literatura, no existe un compendio que las reúna.

## **1.2. ALCANCE DEL PROYECTO**

### **1.2.1. Objetivos Generales**

Continuar la misión de la Universidad Industrial de Santander en la generación y adecuación de conocimientos mediante la aplicación de técnicas como los algoritmos genéticos en la optimización del diseño mecánico.

Continuar la visión de la Universidad Industrial de Santander, como institución inteligente capaz de adaptarse de manera eficiente a los cambios del entorno, al adoptar nuevas metodologías de diseño mediante la aplicación de la metodología de optimización del diseño mecánico en un motor rotativo.

### **1.2.2. Objetivos Específicos**

- Documentar con fundamentos teóricos una metodología de diseño de motores de combustión interna, rotativo tipo Wankel.
- Modelar, documentar y desarrollar el proceso de optimización tomando como funciones objetivo la maximización de la eficiencia térmica y minimización del peso, de un motor rotativo tipo Wankel que cumpla con las siguientes características:
  - Desplazamiento Volumétrico: 10 cc.
  - Potencia: 1.75 BHP @ 16.000 RPM
  - Rango de Velocidades: 2500-16,500 RPM
  - 1 Rotor
- Corroborar las ventajas obtenidas de la utilización de la metodología de optimización en las labores de ingeniería de manera cualitativa y cuantitativa mediante la comparación de resultados entre el diseño óptimo y el diseño base.

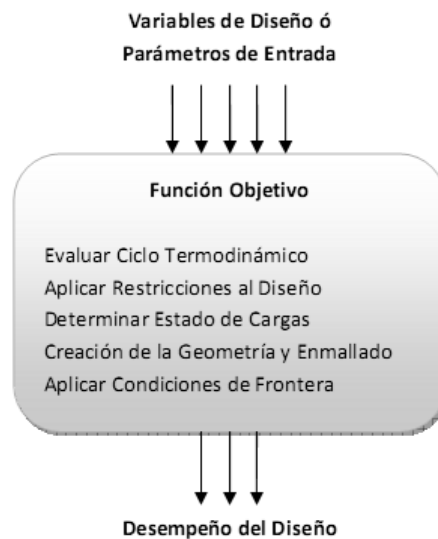
- Generar un módulo para aula virtual con los fundamentos teóricos que llevaron al desarrollo de la metodología de diseño óptimo, que mediante contenido interactivo contenga lo siguiente:
  - El pre-modelado del sistema, con identificación de las partes objetivo, diagramas cinemáticos, modelos geométricos y ecuaciones de cálculo.
  - La determinación de las funciones objetivo.
  - La determinación de las variables de optimización.
  - Los principios utilizados en la selección de las restricciones del diseño.
  - La determinación de las soluciones del problema y de los criterios de aptitud.
  - La verificación de resultados obtenidos.
- Escribir un artículo técnico sobre los resultados obtenidos en el proceso de optimización mediante utilización de algoritmos genéticos.

## 2. MODELO MATEMATICO DEL MOTOR ROTATIVO

El concepto presentado por (Rivera & Meneses, 2008) reúne la información disponible del motor rotativo tipo Wankel, y propone un diseño simplificado para un motor pequeño adecuado para un aeromodelo. Si bien esta propuesta inicial es para efectos prácticos funcional, solo se estudia una sola configuración de los múltiples parámetros geométricos que interactúan en el motor.

El enfoque propuesto por el autor consiste en simular el desempeño del motor empleando un algoritmo que emplea modelos analíticos y la metodología de diseño propuesta por los autores en conjunto con un paquete de análisis por elementos finitos – FEA<sup>1</sup>, por sus siglas en inglés – como una especie de ‘caja negra’ acoplada a un algoritmo de optimización multi-objetivo que de manera sistemática no-determinística busca las configuraciones más adecuadas para mejorar cierto criterio de diseño, en nuestro caso la eficiencia térmica y la masa del motor. Una forma más clara de ver cómo funciona el modelo es mediante una representación de entradas y salidas o diagrama MIMO, como se muestra en la Figura 2.

**Figura 2.** Representación MIMO de la evaluación de aptitud.

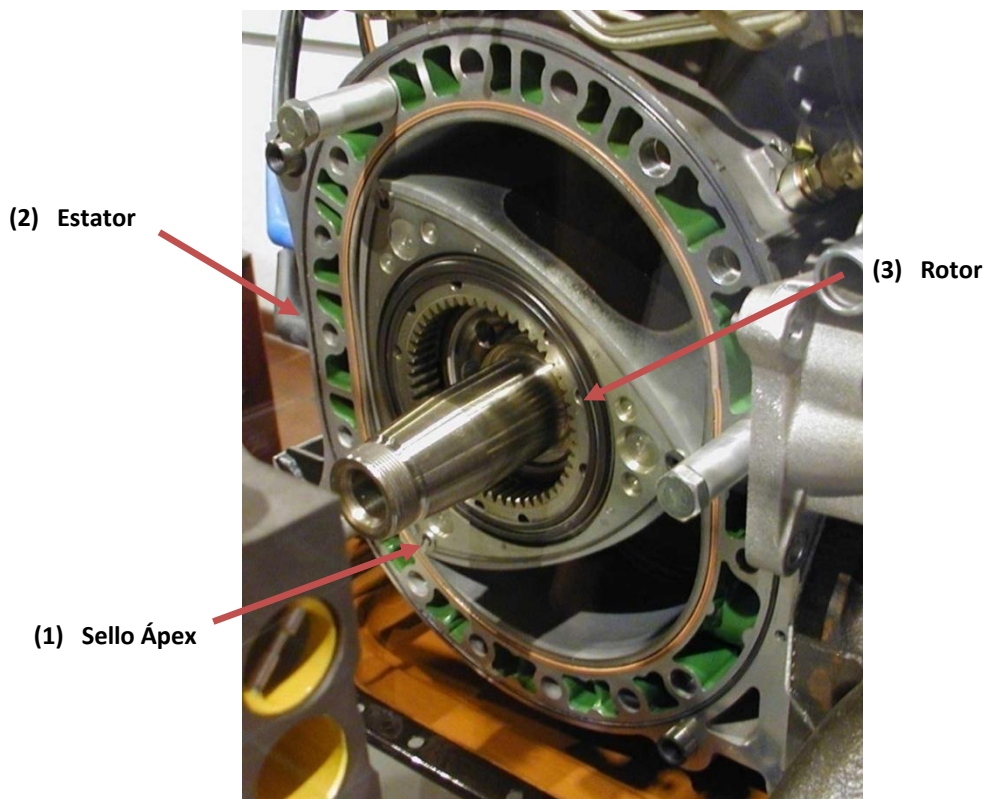


<sup>1</sup> Finite Element Analysis - FEA

El conjunto de parámetros resultante tiene una variedad de usos potenciales, como dar un valor explicativo al modelo al observar la influencia la variación de los parámetros en la salida del sistema, además es posible que resulte útil para explicar y posteriormente refinar el modelo.

Una serie de estudios realizados por la NASA, relacionados con el desempeño de los motores rotativos (Danieli, 1976)(Norman, 1983)(Raju & Willis, 1990) concluyen de manera efectiva que cerca del 25% de la potencia del motor se pierde por concepto de transferencia de calor al ambiente, y enfatiza la importancia de mejorar la eficiencia térmica de estos motores que en general poseen una área de transferencia de calor por unidad de volumen, mayor que su equivalente recíprocante. Para simplificar el problema nos enfocaremos exclusivamente en el conjunto Rotor-Estator, mostrado en la Figura 3.

**Figura 3.** Motor Mazda 13B, Imagen tomada por Sfoskett bajo licencia GFDL



## **2.1. VENTAJAS Y DESVENTAJAS DEL MOTOR ROTATIVO**

Entre las desventajas que presenta este motor el autor destaca la pobre economía del combustible, atribuida típicamente a la fuga de mezcla no quemada a través de los sellos ápex (1) durante la compresión y la combustión a las cámaras adyacentes (Yamamoto, 1981). También, se cree que estas fugas son las causantes de las altas emisiones de hidrocarburos. Los altos costes de fabricación radican en las aleaciones y los diferentes recubrimientos que son requeridos para soportar el desgaste del rotor y de su complejidad geométrica, si bien su número de partes en general es menor, requiere de acabados más finos, herramientas de mejor calidad y equipos especializados.

Sin embargo, en ciertos campos como la aeronáutica, las ventajas de su diseño superan cómodamente los problemas asociados a este. En la literatura consultada, particularmente (Danieli, 1976)(Yamamoto, 1981) se destacan entre otras:

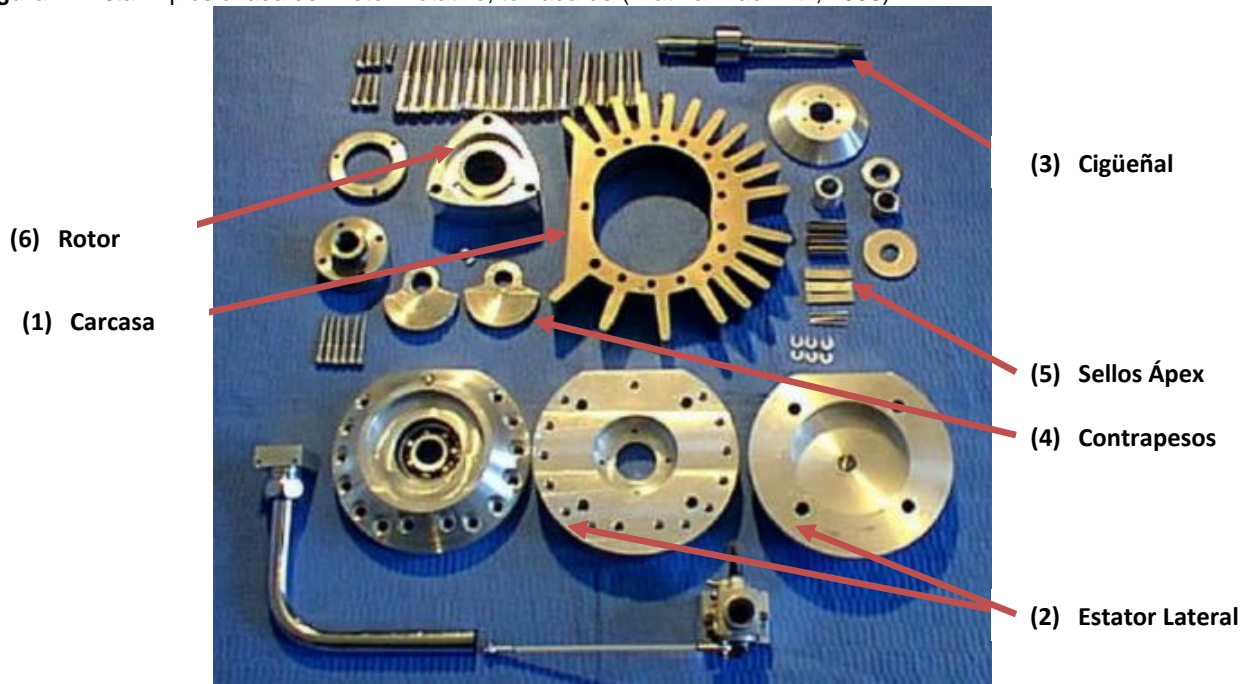
- Mejor relación potencia-peso y mejor relación potencia-volumen
- Mejor arranque en frío.
- Menores niveles de vibración y ruido.
- Mayor confiabilidad y menor número de partes móviles.

## **2.2. PARTES CONSTITUTIVAS DE UN MOTOR ROTATIVO TIPO WANKEL**

El motor rotativo tiene menos piezas móviles que un motor alternativo de 4 tiempos, que se traduce en una mayor confiabilidad de los componentes mecánicos y en un diseño más compacto. Como se observa en la figura 4, el número de partes es bastante reducido, aunque de una complejidad bastante mayor.

Para efectos de estudio se considera solo el conjunto Rotor-Estator solamente pero para ubicar al lector resulta útil identificar las tapas laterales (4), el cigüeñal (3) y sus respectivos contrapesos (5), los sellos de los vértices (6) y el volante de inercia.

Figura 4. Vista Explosionada del Motor Rotativo, tomada de (Kratz & Wackwitz, 1998)



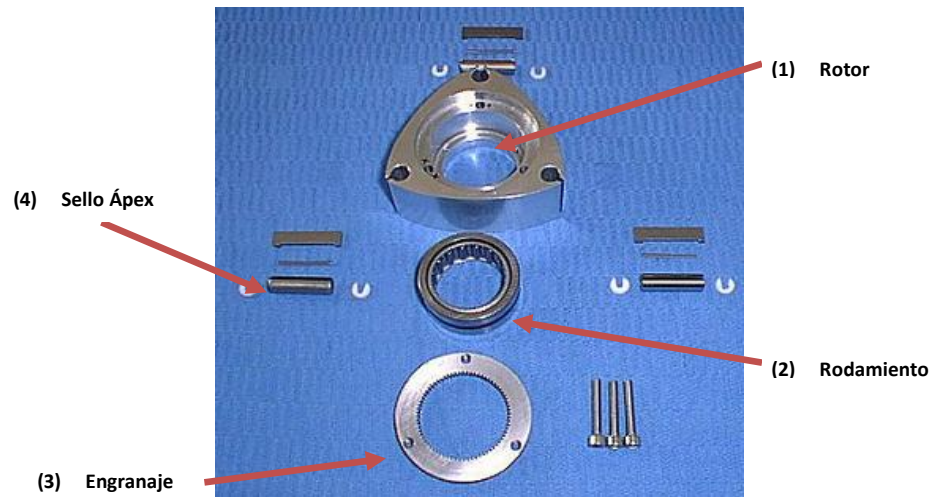
Se incluye una breve descripción de las partes que constituyen el motor y algunas imágenes que sirvan de referencia al lector.

### 2.2.1. Conjunto Rotor

El rotor (3) cumple la misma función que el conjunto de embolo-biela en el motor reciprocante, transmite la presión de los gases en combustión directamente al cigüeñal. También se encarga de la apertura de las lumbreras de admisión y escape. La figura 5 muestra un despiece del conjunto rotor, donde identificamos claramente los sellos de los vértices (1), el engranaje interno (2) el rotor (3) y el rodamiento del rotor (4).

Cada uno de sus lados laterales en conjunto con el estator y los sellos de los vértices (1) produce una cámara aislada que realiza una combustión completa en una vuelta del cigüeñal; Tiene solidario un engranaje interno (2) que se encarga de sincronizar su movimiento de modo que las puntas del rotor se encuentren en todo momento en contacto con las paredes del estator, y un rodamiento para permitir el movimiento relativo con respecto al cigüeñal.

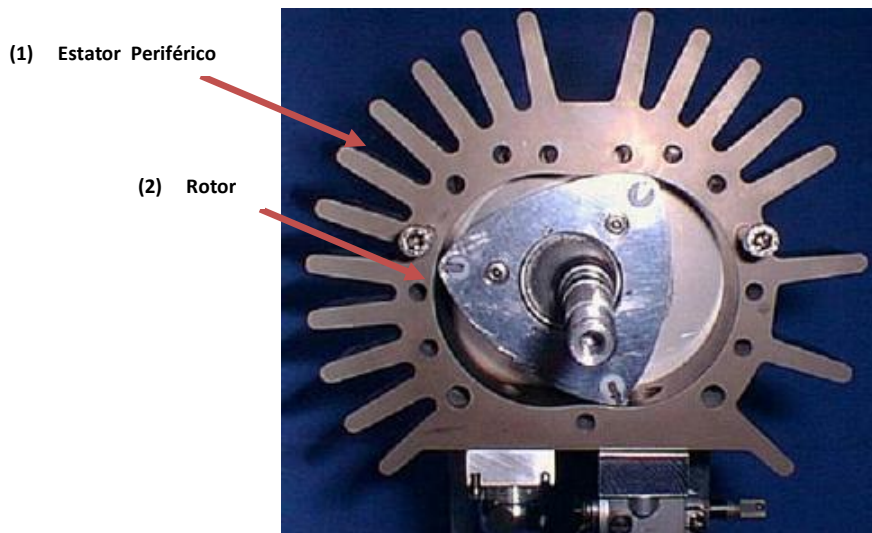
**Figura 5.** Elementos del Conjunto Rotor, imagen tomada de (Kratz & Wackwitz, 1998)



### 2.2.2. Estator Periférico

En la Figura 6 observamos la carcasa o estator periférico (2), cumple la función del bloque del motor y la culata de los motores reciprocantes. Además se encarga de guiar el movimiento del rotor para que sus puntas describan la trayectoria de la epitrocoide y se encarga de evacuar el calor excedente de la combustión, bien sea con un sistema de canales para la refrigeración por líquido como en los motores a gran escala, o mediante la utilización de un sistema de refrigeración por aletas como el que se muestra solidario a la carcasa.

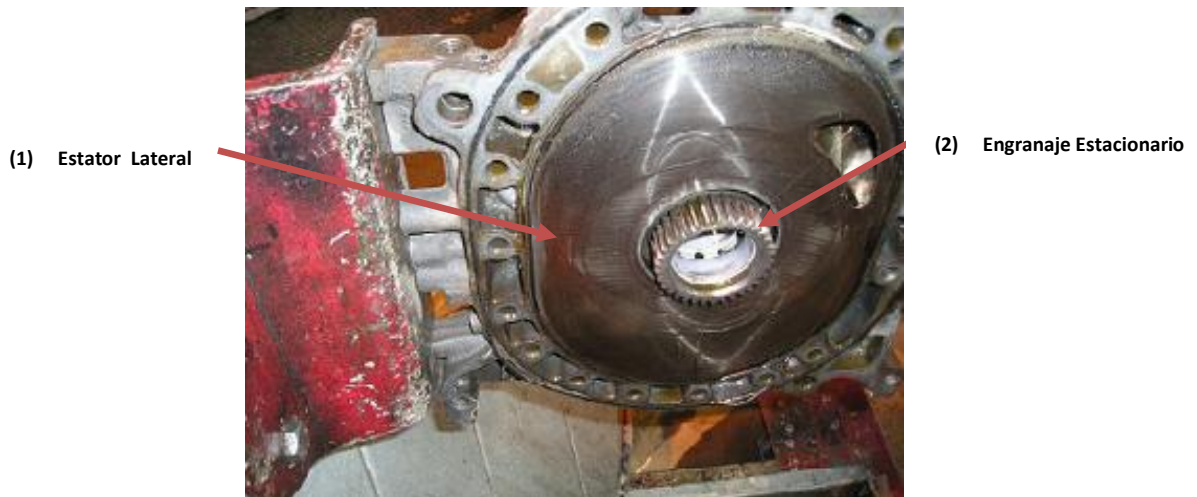
**Figura 6.** Conjunto Estator-Rotor, imagen tomada de (Kratz & Wackwitz, 1998)



### 2.2.3. Estator Lateral

La tapa, cubierta o estator lateral (1), cumple la misma función que la carcasa y elimina buena parte del calor residual, pero además permite aislar las cámaras en los motores multi-rotor y albergan el engranaje estacionario (2) que es el que se encarga de sincronizar el movimiento del rotor, como se muestra en la Figura 7.

Figura 7. Estator Lateral con engranaje estacionario, imagen tomada de (Kratz & Wackwitz, 1998)



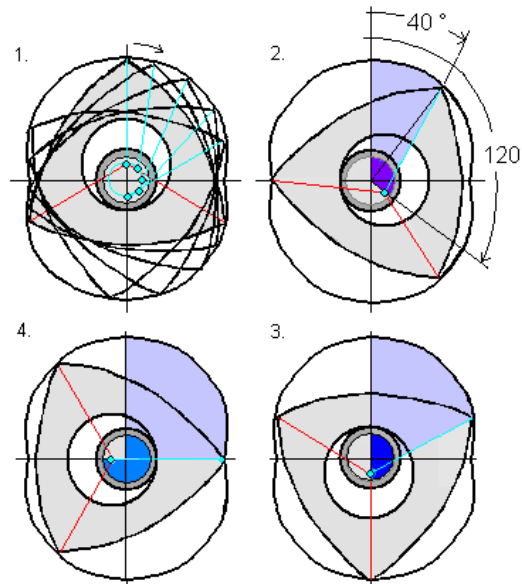
### 2.3. GEOMETRÍA DEL MOTOR WANKEL<sup>2</sup>

En el motor Wankel, un rotor de tres puntas se encuentra montado en un eje excéntrico que se encuentra centrado con el eje geométrico de una carcasa que alberga el rotor. En su forma más conocida el rotor y la carcasa forman simultáneamente tres cámaras aisladas.

Una representación sencilla de la geometría del rotor se puede apreciar en la figura 8. A medida que gira el cigüeñal, un engranaje interno solidario al rotor con una relación cinemática de 2:3 asegura que el rotor gire a 1/3 de la velocidad del cigüeñal como se observa en la figura.

<sup>2</sup> Modelo tomado de (Yamamoto, 1981)

**Figura 8.** Geometría característica del conjunto Estator-Rotor, Imagen tomada de (Kino, 2006)



Esto quiere decir que se requieren de tres vueltas completas del cigüeñal para lograr completar las cuatro etapas de un motor recíprocante equivalente en una de las cámaras. Como hay tres cámaras por rotor, tenemos un proceso de aspiración y una carrera de potencia por vuelta del cigüeñal, luego el comportamiento del motor en realidad sería más parecido al de un motor recíprocante de dos cilindros y cuatro tiempos.

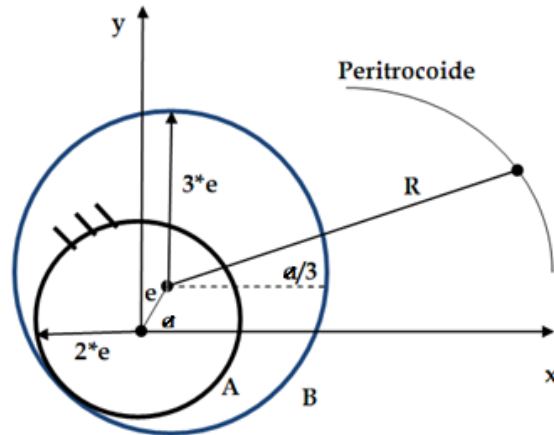
El motor Wankel carece de los trenes de válvulas de los motores de cuatro tiempos y en su lugar utiliza un sistema de lumbreras de admisión y escape, que al tener todas las cámaras simultáneamente aisladas, no presenta los mismos problemas de fugas de los motores de dos tiempos. Las lumbreras de admisión normalmente se ubican en la cara lateral del rotor, la placa de puertos o placa lateral, y por lo general tenemos dos puertos de entrada por rotor. El puerto de escape se ubica en la carcasa de forma periférica.

### **2.3.1. Forma General de la Carcasa**

La epitrocoide es un caso especial de las curvas trocoidales, que es la curva descrita por un punto ubicado en un círculo, a medida que esta gira sin deslizarse por la periferia

de otro círculo como se observa en la Figura 9. Cuando el punto que traza la curva se encuentra por fuera del círculo base, esta curva se conoce como una epitrocoide.

**Figura 9.** Generación de la peritrocoide, imagen tomada de (Rivera & Meneses, 2008)<sup>3</sup>



La relación entre los radios de estos círculos es análoga a la relación de transmisión, luego una carcasa de dos lóbulos requiere un rotor de tres lóbulos, que para funcionar adecuadamente requiere mantener una relación de velocidades de 3:2 entre la corona solidaria al rotor y el engranaje estacionario.

Si bien técnicamente el funcionamiento del motor es posible con un número ilimitado de cámaras, todos los esfuerzos se concentran actualmente en los motores de dos lóbulos, porque poseen proceso de fabricación más sencillo. Expresada en función paramétrica de “x” y “y” tenemos que la curva interior de la carcasa del motor tendría la siguiente forma:

**Ecuación 1.** Ecuación básica de la peritrocoide, (Eq. 2.4.) (Yamamoto, 1981)

$$x = e \cos (\theta) + R \cos \left( \frac{\theta}{3} \right)$$

$$y = e \sin (\theta) + R \sin \left( \frac{\theta}{3} \right)$$

Definimos “e” como la excentricidad del rotor, “R” es el radio generador y “θ” es el ángulo de giro del cigüeñal, como se observa en la Figura 9.

<sup>3</sup> Imagen generada empleando MapleSoft [DEMO]

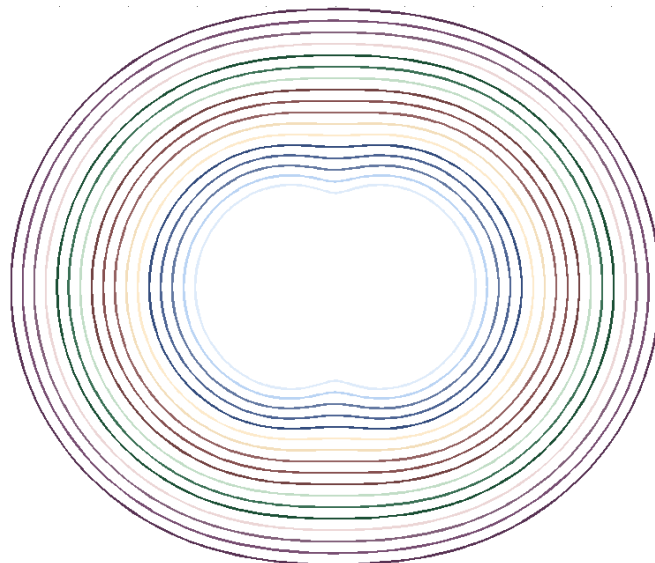
La relación de compresión en los motores alternativos se encuentra limitada solamente por el criterio del diseñador y solo es necesario modificar las dimensiones del cigüeñal para este fin. Para el caso del motor Wankel esta relación se encuentra limitada por los parámetros característicos del motor “ $e$ ” y “ $R$ ”, y la relación entre estos dos parámetros .

Como el tamaño y la trayectoria del rotor definen la superficie de trabajo, el volumen de trabajo se rige por esta misma consideración, luego los volúmenes mínimos y máximos de trabajo, y por consiguiente la relación de compresión, se encuentran definidos una vez se han seleccionado los parámetros característicos.

Es preciso proyectar esta curva teórica a una distancia “ $a$ ” hacia afuera para dar espacio al sello y cambiar su línea de contacto con la superficie de la carcasa a medida que el rotor gira.

La ecuación paramétrica de la carcasa corregida por la oscilación del sello se muestra en la ecuación 2 y podemos observar la familia de curvas que arroja para diferentes valores de “ $K$ ” en la Figura 10.

**Figura 10.** Forma general de la carcasa para diferentes valores de “ $K$ ”<sup>4</sup>



<sup>4</sup> Imagen generada empleando MATLAB R2008a

**Ecuación 2.** Ecuación modificada de la peritrocoide, (Eq. 2.4.) (Yamamoto, 1981)

$$x = e \cos(\theta) + R \cos\left(\frac{\theta}{3}\right) + a \cos\left(\frac{\theta}{3} + \phi\right)$$

$$y = e \sin(\theta) + R \sin\left(\frac{\theta}{3}\right) + a \sin\left(\frac{\theta}{3} + \phi\right)$$

### 2.3.2. Forma General del Rotor

El rotor se define como la involuta<sup>5</sup> de la trocoide que define a la carcasa, de modo que sus 3 puntas estén siempre en contacto con las paredes del motor. Considerando las ecuaciones del rotor y definiendo “ $v$ ” como el parámetro independiente para el perfil del rotor y “ $\theta$ ” como el parámetro independiente en la carcasa, se tiene que la ecuación básica de la involuta es la siguiente.

**Ecuación 3.** Ecuación básica de la involuta de la peritrocoide, (Eq. 2.9.) (Yamamoto, 1981)

$$x(v) = R \cdot \cos(2v) - \frac{3e^2}{R} \sin(6v) \cos(2v) + 2 \cdot e \cdot (\cos(2v) \cos(3v)) \sqrt{1 - \left(\frac{3e}{R} \sin(3v)\right)^2}$$

$$y(v) = R \cdot \sin(2v) - \frac{3e^2}{R} \sin(6v) \sin(2v) + 2 \cdot e \cdot (\sin(2v) \cos(3v)) \sqrt{1 - \left(\frac{3e}{R} \sin(3v)\right)^2}$$

De manera similar a la carcasa, el rotor será proyectado una distancia “ $a$ ” hacia afuera del rotor, siendo “ $a$ ” la diferencia entre el espacio reservado para el sello “ $a$ ” y el ajuste mínimo entre el rotor y la carcasa “ $S_p$ ”, esta debe tener en cuenta las limitaciones en la manufactura, la dilatación térmica de los elementos del motor, y las tolerancias de manufactura de los rodamientos. Esta proyección se muestra en la Ecuación 4.

**Ecuación 4.** Ecuación modificada de la involuta de la peritrocoide, (Eq. 2.10.) (Yamamoto, 1981)

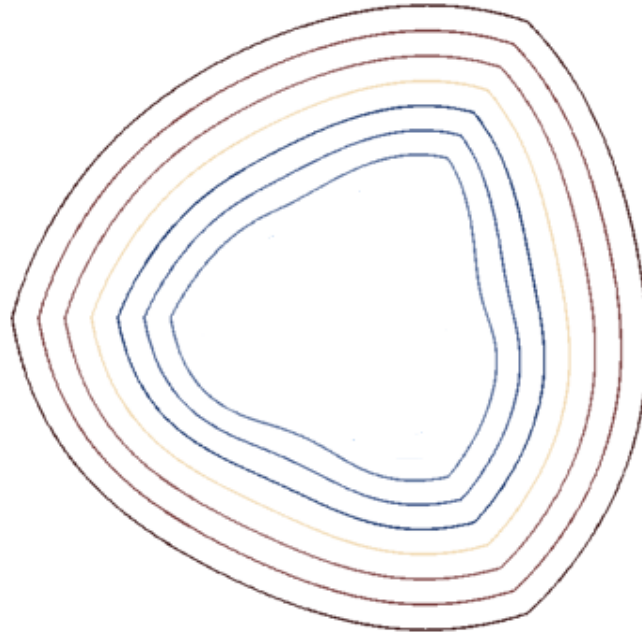
$$x'(v) = x(v) + \frac{3ea'}{2R} (\cos 5v - \cos v) + a' \cos(2v) \sqrt{1 - \left(\frac{3e}{R} \sin(3v)\right)^2}$$

$$y'(v) = y(v) + \frac{3ea'}{2R} (\sin 5v - \sin v) + a' \sin(2v) \sqrt{1 - \left(\frac{3e}{R} \sin(3v)\right)^2}$$

<sup>5</sup> Dada una curva  $C$  definimos la involuta  $C_I$  como la localización de los centros de curvatura de  $C$ . Es la operación inversa a una evoluta.

De igual forma, para diferentes valores de “ $K$ ” se obtiene una familia de curvas como se muestra en la Figura 11.

**Figura 11.** Forma general del rotor para diferentes valores de  $K^6$



### 2.3.3. Ángulo de Oscilación del Sello

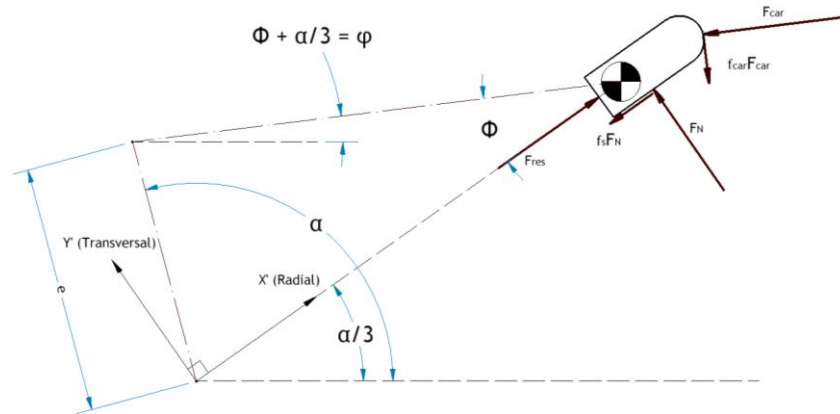
Los sellos ubicados en los vértices del rotor se encuentran perpendiculares a la superficie de este en todo momento, el ángulo de oscilación del sello es el ángulo formado por el radio generador y la normal a la superficie de la carcasa.

Se define “ $\phi$ ” como el ángulo de oscilación del sello y “ $\varphi$ ” como el ángulo de oscilación con respecto al rotor como se muestra en la figura 12. Podemos apreciar la variación del ángulo de oscilación con respecto al ángulo de cigüeñal para diferentes valores de “ $K$ ” en la Figura 13.

---

<sup>6</sup> Imagen generada empleando MATLAB R2008a

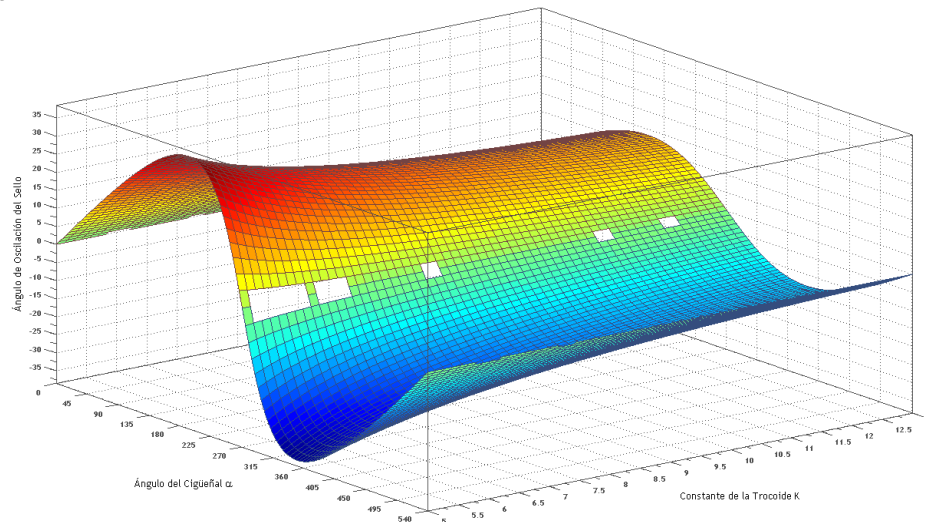
**Figura 12.** Ángulos de oscilación y diagrama de cuerpo libre en el sello



**Ecuación 5.** Ángulo de Oscilación del Sello en función de la geometría, (Eq. 2.12.) (Yamamoto, 1981)

$$\phi(K, \theta) = \tan^{-1} \frac{3 \cdot \sin\left(\frac{2\theta}{3}\right)}{K + 3 \cdot \cos\left(\frac{2\theta}{3}\right)}$$

**Figura 13.** Ángulo de Oscilación en función de “θ” y “K”<sup>7</sup>



### 2.3.4. Relación de Compresión

La relación de compresión viene dada por la relación de las áreas mínima y máxima en la cámara de trabajo, que se puede obtener a partir de las ecuaciones 2 y 4 de las secciones 2.3.1 y 2.3.2 respectivamente mediante una sumatoria de áreas como se muestra en la Figura 14.

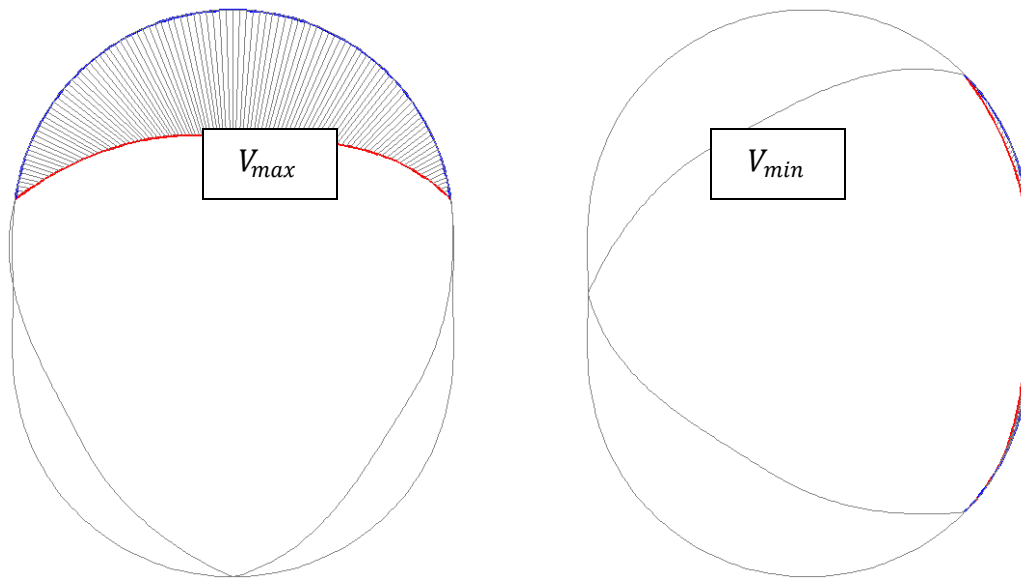
<sup>7</sup> Imagen generada empleando MATLAB R2008a

Para reducir el tiempo empleado en el cálculo de la relación de compresión podemos emplear una aproximación a la relación teórica, sin tener en cuenta el juego entre el rotor y la carcasa.

**Ecuación 6.** Aproximación de la relación de compresión teórica

$$\varepsilon_{theory} \approx \frac{3\sqrt{3}}{2} \cdot \frac{R+a}{e} \approx \frac{3\sqrt{3}}{2} K$$

**Figura 14.** Volúmenes Máximo y Mínimo de la cámara de combustión<sup>8</sup>



Además para la simulación del ciclo es preciso tener el volumen teórico de la cámara de combustión para cualquier ángulo de cigüeñal, para lo que empleamos una aproximación sencilla del volumen desplazado en la cámara con dos radios generadores “ $R_1$ ” y “ $R_2$ ”.

**Ecuación 7.** Volumen de la cámara de trabajo o ‘cilindrada’, (Eq. 2.30.) (Yamamoto, 1981)

$$R_1 = R + a ; R_2 = R + a'$$

$$V_H = V_{max} - V_{min} = \sqrt{3}eb(2R_1 + R_2)$$

El volumen mínimo de la cámara, es decir el volumen en el punto muerto superior (TDC<sup>9</sup>) se puede estimar como:

<sup>8</sup> Imagen generada empleando MATLAB R2008a

**Ecuación 8.** Volumen en el TDC o punto muerto superior.

$$V_{TDC} = \frac{V_H}{\varepsilon - 1}$$

Donde,  $\varepsilon$  corresponde a la relación de compresión. El volumen instantáneo de la cámara de combustión se puede aproximar como:

**Ecuación 9.** Volumen instantáneo de la cámara de combustión, función armónica.

$$V(\theta) = V_{TDC} + \frac{\sqrt{3}}{2} eb(2R_1 + R_2) \left[ 1 - \sin\left(\frac{2}{3}\theta + \frac{\pi}{2}\right) \right] + V_{ad}$$

Donde  $V_{ad}$  es el volumen del maquinado que se le efectúa al rotor para mejorar la combustión. Teniendo en cuenta esto podemos obtener la relación de compresión real del motor a partir de los volúmenes máximos y mínimos en la posición TDC y BDC<sup>10</sup> respectivamente:

**Ecuación 10.** Relación entre la relación de compresión real con la relación de compresión teórica.

$$V_{BDC} = \frac{\varepsilon}{\varepsilon - 1} V_H$$
$$\varepsilon_{real} = \frac{V_{BDC} + V_{ad}}{V_{TDC} + V_{ad}}$$

### 2.3.5. Engranajes de Fase

El sistema del eje excéntrico se ve afectado por las dimensiones básicas del motor, de modo que una vez establecida la excentricidad del motor, esta inevitablemente determina el diámetro mínimo del eje de salida y los diámetros primitivos de los engranajes que se encargan de controlar el movimiento del rotor y asegurarse de que las puntas de los sellos se encuentren en todo momento en contacto con las paredes de la carcasa.

La relación entre la excentricidad y los diámetros primitivos de los engranajes se muestra en la ecuación 11 a continuación.

---

<sup>9</sup> *Top Dead Center*, por sus siglas en inglés. También conocido como PMS.

<sup>10</sup> *Bottom Dead Center*, por sus siglas en inglés. También conocido como PMI.

**Ecuación 11.** Restricción a los diámetros primitivos de los engranajes, (Sec. 3.3.) (Yamamoto, 1981)

$$D_{p_e} = 2e$$

$$D_{p_i} = 6e$$

Donde  $D_p$  corresponde al valor del diámetro primitivo para los engranajes interno ( $i$ ) y externo ( $e$ ) respectivamente.

El diámetro primitivo es relación directa del número de dientes del engranaje y el módulo métrico de los dientes, de modo que utilizando el menor número de dientes admisible con módulos estandarizados podemos obtener que valores puede tomar la excentricidad de modo que la manufactura sea plausible, al menos desde el punto de vista de la utilización de un modulo comercialmente disponible. En la Tabla 1 se muestran los valores que puede tomar la excentricidad en función del número de dientes del engranaje estacionario y el modulo métrico.

**Tabla 1.** Valores de la excentricidad propuestos.

$Z_e$ $m$	16	18	20
0.6	2.4	2.7	3
0.7	2.8	3.15	3.5
0.8	3.2	3.6	4
0.9	3.6	4.05	4.5

Excentricidad  $e$  (mm.)

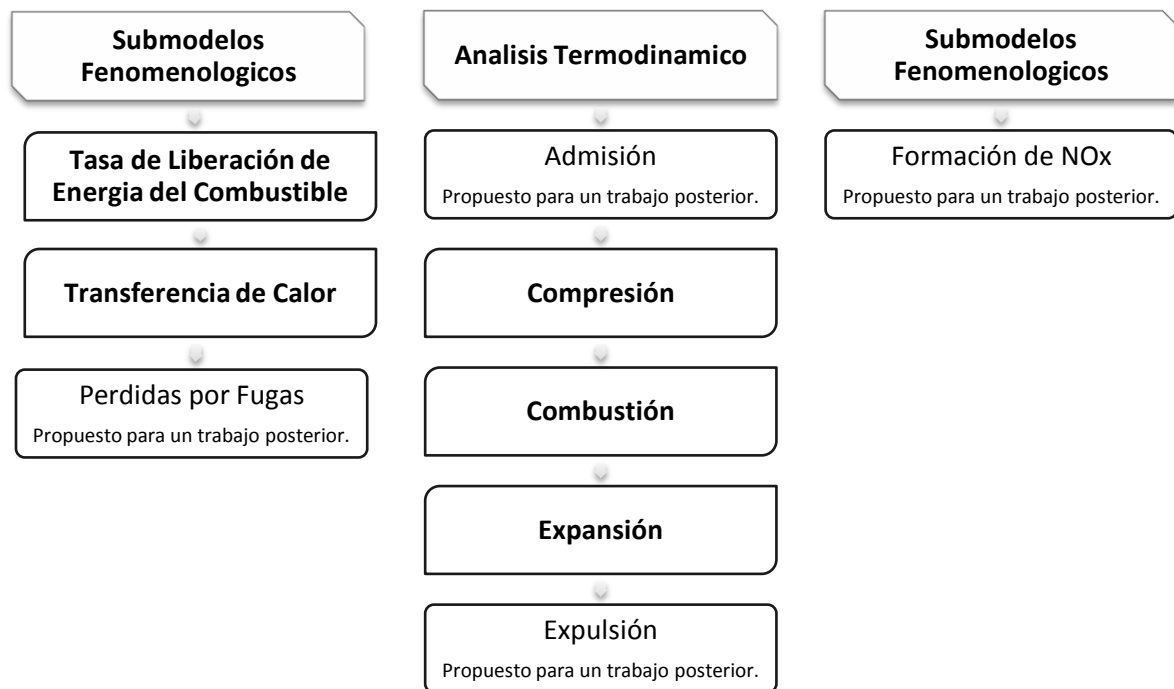
Como el engranaje interno es solidario al rotor para la evaluación del conjunto rotor es preciso efectuar una verificación de esfuerzos de acuerdo a la metodología AGMA<sup>11</sup> como se muestra en el apéndice D.

<sup>11</sup> *American Gear Manufacturers Association*, por sus siglas en inglés. Es la autoridad pertinente para la elaboración de estándares de diseño y análisis de engranajes.

## 2.4. EFICIENCIA TÉRMICA – MODELO TERMODINAMICO<sup>12</sup>

A continuación en la Figura 15 se presenta una representación esquemática de los fenómenos que se dan a lugar al interior del motor rotativo y resaltados entre ellos los procesos que son objeto de estudio del presente trabajo. En primera instancia se observa que para el análisis termodinámico se omiten las etapas de admisión y escape, lo que resulta en el uso de un modelo de llenado y expulsión ideal para el trabajo en cuestión.

Figura 15. Procesos que interactúan en el motor



### 2.4.1. Consideraciones y Simplificaciones

Para poder resolver la forma diferencial de la primera ley de la Termodinámica fueron necesarias hacer ciertas simplificaciones:

- La cámara de combustión puede considerarse como un sistema abierto, que contiene diferentes zonas, sujetas a procesos cuasiestáticos que obedecen las leyes de la termodinámica.

<sup>12</sup> Modelo tomado en su mayor parte de (Ferguson, 1986)

- Salvo por el período de combustión, los contenidos de la cámara son asumidos como una mezcla homogénea de gases ideales no reactivos.
- La entalpía promedio, energía interna y volumen específico de cada zona serán calculados a cada instante  $t$  a partir de su temperatura, presión y composición promedio.
- Durante las etapas de admisión y compresión, el contenido de la cámara es caracterizado como una mezcla de aire atmosférico, combustible y una fracción de gases residuales. Durante la etapa de expansión y escape, se asume que la cámara solo contiene productos de la combustión en equilibrio químico.
- Se asume que no hay ondas de presión en el interior de la cámara de combustión de modo que la presión de los gases es constante al interior de la cámara.

Para realizar la simulación se maneja entonces como parámetro independiente el tiempo, representado en ángulo de giro del cigüeñal con una velocidad de rotación constante en un modelo estacionario.

El funcionamiento del motor se ha dividido en 4 etapas que dependen de los parámetros de entrada del modelo, estas etapas son:

- Compresión, Cierre de la lumbrera de admisión a inicio de la combustión.
- Compresión-Combustión, Inicio de la combustión a PMS (TDC).
- Expansión-Combustión, PMS (TDC) a fin de la combustión.
- Expansión, Fin de la combustión a apertura de la lumbrera de escape.

Se emplean las rutinas para el cálculo de la combustión y para la simulación de la operación del motor de encendido por chispa<sup>13</sup> planteados por (Ferguson, 1986), y posteriormente modificados por (Buttsworth, 2002) para aprovechar la utilización del álgebra matricial incorporada en MATLAB. Cada uno de estos modelos da como resultado un conjunto de ecuaciones diferenciales y algebraicas que deben ser

---

<sup>13</sup> Motor de Combustión Interna de Ignición por Chispa – MCI ICH

resueltas de manera simultánea y que serán explicadas más adelante en el texto. Véase pág. 39.

### 2.4.2. Modelo de Compresión-Expansión

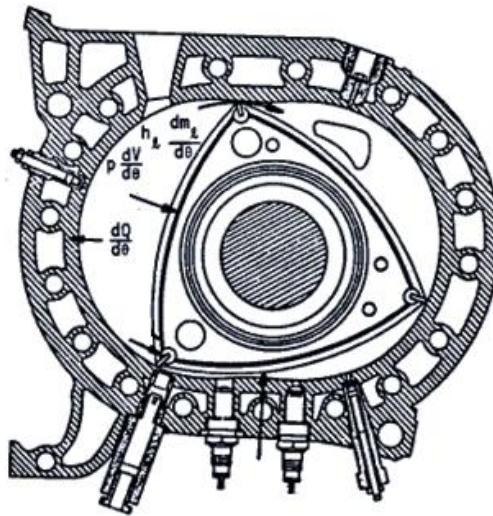
Empleamos un modelo adimensional, que supone no interesa la distribución espacial de las variables dentro de la cámara de combustión. Aplicamos la ley de la conservación de la energía a un sistema abierto como el que se muestra en la Figura 16.

**Ecuación 12.** Conservación de la energía aplicada a un sistema abierto

$$\frac{dU}{d\theta} = -p \frac{dV}{d\theta} - \frac{dQ}{d\theta} - h_{leak} \frac{dm_{leak}}{d\theta}$$

Donde “ $p$ ”, “ $V$ ” y “ $U$ ” corresponden a la presión, el volumen de la cámara y la energía interna del sistema, respectivamente; “ $\theta$ ” es el ángulo de giro del cigüeñal, “ $\frac{dQ}{d\theta}$ ” representa las pérdidas por transferencia de calor del sistema, “ $\frac{dm_{leak}}{d\theta}$ ” representa las pérdidas de masa por concepto de fugas, mientras que “ $h_{leak}$ ” corresponde a la entalpía de la masa que sale del sistema por fugas. Para el cálculo de las propiedades termodinámicas del fluido de trabajo en función de la temperatura empleamos dos subrutinas llamadas FARG y ECP.

**Figura 16.** Primera Ley aplicada al volumen de control, imagen tomada de (Borman & Nishiwaki, 1987)



La subrutina FARG – por sus siglas en inglés, *Fuel Air Residual Gas* - se encarga del cálculo de las propiedades de la mezcla aire-combustible y productos de la combustión en función de la temperatura y la presión al interior de la cámara.

La rutina debe obtener la composición molar de la mezcla aire-combustible y computar ese valor con los gases residuales producto de la combustión.

Para el cálculo de las propiedades termodinámicas empleamos las ecuaciones obtenidas por (Gordon & McBride, 1971) a partir de información experimental, en función de 7 parámetros para cada una de las sustancias que componen la mezcla.

A continuación la Ecuación 13 muestra la forma de los parámetros ajustados de (Gordon & McBride, 1971) para obtener el calor específico, la entalpía y la entropía.

**Ecuación 13.** Aproximación a las propiedades termodinámicas en función de T

$$\frac{C_p}{R} = a_1 + a_2T + a_3T^2 + a_4T^3 + a_5T^4$$

$$\frac{h}{RT} = a_1 + \frac{a_2}{2}T + \frac{a_3}{3}T^2 + \frac{a_4}{4}T^3 + \frac{a_5}{5}T^4 + a_6\frac{1}{T}$$

$$\frac{s}{R} = a_1 \ln T + a_2T + \frac{a_3}{2}T^2 + \frac{a_4}{3}T^3 + \frac{a_5}{4}T^4 + a_7$$

Donde, “ $a_1 \dots a_7$ ” corresponden a parámetros ajustados en base a mediciones de (JANAF, 1971), “ $T$ ” corresponde a la temperatura de la especie química en cuestión y “ $R$ ” es la constante específica del gas. Por su parte, para el cálculo de las propiedades de los combustibles empleamos las curvas propuestas por (Heywood, 1988) mostradas en la Ecuación 14.

**Ecuación 14.** Aproximación a las propiedades del combustible en función de T

$$\frac{C_p}{R} = a_1 + a_2T + a_3T^2 + a_4T^3 + a_5\frac{1}{T^2}$$

$$\frac{h}{RT} = a_1 + \frac{a_2}{2}T + \frac{a_3}{3}T^2 + \frac{a_4}{4}T^3 - a_5\frac{1}{T^2} + a_6\frac{1}{T}$$

$$\frac{s}{R} = a_1 \ln T + a_2T + \frac{a_3}{2}T^2 + \frac{a_4}{3}T^3 + a_5\frac{1}{T^2} + a_7$$

Se determinan las propiedades de cada una de las 10 especies identificadas y se proceden a aplicar las reglas de mezcla.

La subrutina ECP – por sus siglas en inglés, *Equilibrium Combustion Products* – se emplea para determinar las propiedades de los productos de la combustión. Para esto empleamos el método propuesto por (Olikara & Borman, 1975) para resolver la composición, calores específicos en equilibrio y otras derivadas parciales que resultan útiles en el análisis termodinámico.

El método (Olikara & Borman, 1975) emplea una expresión ajustada a datos de la tabla JANAF (JANAF, 1971) para obtener las constantes de equilibrio químico de la forma:

**Ecuación 15.** Ecuación de Olikara & Borman para la constante de equilibrio químico

$$\log K_p = A \ln \left( \frac{T}{1000} \right) + \frac{B}{T} + C + DT + ET^2$$

Donde, “ $K_p$ ” es el valor de la constante de equilibrio de la reacción, “ $T$ ” corresponde a la temperatura de la mezcla sin quemar, mientras que “ $A \dots E$ ” corresponden a parámetros ajustados a la curva.

El método ordena las ecuaciones de forma tal que puedan ser resueltas por el método de Newton-Raphson y de esta manera obtener las composiciones de los productos de la combustión.

### 2.4.3. Ley de Liberación de Calor

En este modelo fue empleada una relación empírica para expresar la fracción de calor adicionado en función del ángulo del cigüeñal por simplicidad. En este orden de ideas existen dos formas que predominan en este tipo de simulaciones como son la función armónica y la función de Wiebe. Para efectos prácticos empleamos la armónica.

**Ecuación 16.** Expresión Armónica para la velocidad de aporte de calor

$$\chi = \frac{1}{2} \left\{ 1 - \cos \left[ \frac{\pi(\theta - \theta_s)}{\theta_b} \right] \right\}$$

Donde,  $\chi$  corresponde a la fracción de masa quemada que se utiliza como parámetro de control en las etapas de Compresión-Combustión y Expansión-Combustión que emplean un modelo multizona, como se observa en el trabajo de (Ramos, 1989); “ $\theta$ ” corresponde al ángulo de giro del cigüeñal, mientras que “ $\theta_s$ ” es el ángulo de ignición de la mezcla y “ $\theta_b$ ” la duración total de la combustión en grados de giro del cigüeñal.

En función de este parámetro se puede expresar la energía del sistema y el volumen específico para el modelo multi-zona como:

**Ecuación 17.** Expresión para la energía y el volumen específicos en un modelo multi-zona

$$u = \chi u_b + (1 - \chi)u_u \quad v = \chi v_b + (1 - \chi)v_u$$

Los subíndices “ $u$ ” y “ $b$ ” hacen referencia directa a las zona de mezcla fresca y quemada, respectivamente. Tanto la energía  $u_i$  como el volumen específico  $v_i$  para ambas zonas se encuentran en función de la presión  $p$  y la temperatura  $T_i$ , de modo que podamos obtener las derivadas logarítmicas parciales requeridas en la Ecuación 12.

**Ecuación 18.**

$$v_i = v_i(T_i, p)$$

$$\frac{dv_i}{d\theta} = \frac{\partial v_i}{\partial T_i} \frac{dT_i}{d\theta} + \frac{\partial v_i}{\partial p} \frac{dp}{d\theta} = \frac{v_i}{T_i} \frac{\partial \ln v_i}{\partial \ln T_i} \frac{dT_i}{d\theta} + \frac{v_i}{p} \frac{\partial \ln v_i}{\partial \ln p} \frac{dp}{d\theta}$$

De igual forma, la relación entre  $u_i$ ,  $T_i$  y  $p$  se puede determinar mediante la expresión:

**Ecuación 19.**

$$\frac{du_i}{d\theta} = \frac{\partial u_i}{\partial T_i} \frac{dT_i}{d\theta} + \frac{\partial u_i}{\partial p} \frac{dp}{d\theta}$$

Si se sustituyen las derivadas parciales por los valores obtenidos mediante la rutina ECP obtenemos:

**Ecuación 20.**

$$\frac{du_i}{d\theta} = \left( c_{p_i} - \frac{pv_i}{T_i} \frac{\partial \ln v_i}{\partial \ln T_i} \right) \frac{dT_i}{d\theta} + v_i \left( \frac{\partial \ln v_i}{\partial \ln T_i} + \frac{\partial \ln v_i}{\partial \ln p} \right) \frac{dp}{d\theta}$$

La Ecuación 12 puede reescribirse como:

**Ecuación 21.** Conservación de la energía en función de la energía específica

$$m \frac{du}{d\theta} + u \frac{dm}{d\theta} = -p \frac{dV}{d\theta} - \frac{dQ}{d\theta} - h_{leak} \frac{dm_{leak}}{d\theta}$$

Derivando la Ecuación 17 tenemos que:

**Ecuación 22.**

$$m \frac{du}{d\theta} = m \left[ \chi \frac{du_b}{d\theta} + (1 - \chi) \frac{du_u}{d\theta} + (u_b - u_u) \frac{d\chi}{d\theta} \right]$$

Sustituyendo la Ecuación 20 en la Ecuación 22 se tiene:

**Ecuación 23.**

$$m \frac{du}{d\theta} = m \chi \left( c_{p_b} - \frac{pv_b}{T_b} \frac{\partial \ln v_b}{\partial \ln T_b} \right) \frac{dT_b}{d\theta} + m(1 - \chi) \left( c_{p_u} - \frac{pv_u}{T_u} \frac{\partial \ln v_u}{\partial \ln T_u} \right) \frac{dT_u}{d\theta} + m \left[ \chi v_b \left( \frac{\partial \ln v_b}{\partial \ln T_b} + \frac{\partial \ln v_b}{\partial \ln p} \right) + (1 - \chi) v_u \left( \frac{\partial \ln v_u}{\partial \ln T_u} + \frac{\partial \ln v_u}{\partial \ln p} \right) \right] \frac{dp}{d\theta} + (u_b - u_u) \frac{d\chi}{d\theta}$$

Mientras que el término “ $u \frac{dm}{d\theta}$ ” corresponde al escape de mezcla por las fugas en la cámara, donde se introduce un parámetro conocido como constante de Blowby “ $C$ ”. Este valor no es conocido para un motor rotativo típico luego se efectúan los cálculos con un valor típico para motores recíprocos y se propone una calibración de parámetros para un trabajo posterior.

**Ecuación 24.** Constante de Blowby

$$u \frac{dm}{d\theta} = -\frac{Cm}{\omega}$$

A partir de esta expresión tenemos que la masa para cualquier ángulo de cigüeñal es:

**Ecuación 25.** Masa instantánea para cualquier ángulo de cigüeñal

$$m = m_1 \exp \left[ -\frac{C(\theta - \theta_1)}{\omega} \right]$$

Las pérdidas por transferencia de calor para las dos zonas pueden expresarse como:

**Ecuación 26.** Perdidas por transferencia de calor para cada zona

$$Q_i = h_i A_i (T_i - T_w)$$

Donde  $h_i$  es el coeficiente de transferencia de calor por convección interna para cada una de las zonas calculado a partir de la correlación de Woschni,  $A_i$  es el área de transferencia de calor que se encuentra en función de la fracción de masa quemada y  $T_w$  es la temperatura de la cara interna de la carcasa, asumida con un valor constante de 180°C.

La rutina determina a partir del conjunto de ecuaciones diferenciales lineales ordinarias<sup>14</sup> de primer orden que son resueltas con un integrador numérico para obtener el valor de las variables mostradas a lo largo del intervalo de interés.

Por conveniencia se definen los parámetros “ $A_{ODE} \dots E_{ODE}$ ” para simplificar la escritura del sistema de ecuaciones.

**Ecuación 27.**

$$A_{ODE} = \frac{1}{m} \left( \frac{dV}{d\theta} + \frac{VC}{\omega} \right)$$

$$B_{ODE} = \frac{1}{m\omega} \left( \frac{v_b}{C_{p_b}} \frac{\partial \ln v_b}{\partial T_b} \frac{Q_b}{T_b} + \frac{v_u}{C_{p_u}} \frac{\partial \ln v_u}{\partial T_u} \frac{Q_u}{T_u} \right)$$

$$C_{ODE} = -(v_b - v_u) \frac{d\chi}{d\theta} - v_b \frac{\partial \ln v_b}{\partial T_b} \frac{h_u - h_b}{C_{p_b} T_b} \left[ \frac{d\chi}{d\theta} - \frac{(\chi - \chi^2)C}{\omega} \right]$$

$$D_{ODE} = \chi \left[ \frac{v_b^2}{C_{p_b} T_b} \left( \frac{\partial \ln v_b}{\partial T_b} \right)^2 + \frac{v_b}{p} \frac{\partial \ln v_b}{\partial T_b} \right]$$

$$E_{ODE} = (1 - \chi) \left[ \frac{v_u^2}{C_{p_u} T_u} \left( \frac{\partial \ln v_u}{\partial T_u} \right)^2 + \frac{v_u}{p} \frac{\partial \ln v_u}{\partial T_u} \right]$$

Por último observamos el sistema de ecuaciones diferenciales ordinarias de primer orden, en función de los parámetros anteriormente descritos.

<sup>14</sup> Ordinary Differential Equation - ODE

**Ecuación 28.** Conjunto de ecuaciones diferenciales ordinarias a integrar.

$$\begin{aligned}\frac{dP}{d\theta} &= \frac{A_{ODE} + B_{ODE} + C_{ODE}}{D_{ODE} + E_{ODE}} \\ \frac{dT_b}{d\theta} &= \frac{-Q_b}{m\omega\chi} \frac{\partial \ln v_b}{\partial T_b} + \frac{v_b}{C_{p_b}} \frac{\partial \ln v_b}{\partial T_b} \frac{dP}{d\theta} + \frac{(h_u - h_b)}{C_{p_b}} \left[ \frac{1}{\chi} \frac{d\chi}{d\theta} - \frac{(1-\chi)C}{\omega} \right] \\ \frac{dT_u}{d\theta} &= \frac{-Q_u}{m\omega C_{p_u}(1-\chi)} + \frac{v_u}{C_{p_u}} \frac{\partial \ln v_u}{\partial T_u} \frac{dP}{d\theta} \\ \frac{dW}{d\theta} &= P \frac{dV}{d\theta} \\ \frac{dQ_l}{d\theta} &= \frac{Q_u + Q_b}{\omega} \\ \frac{dH_l}{d\theta} &= \frac{Cm}{\omega} [(1-\chi^2)h_u + \chi^2h_b]\end{aligned}$$

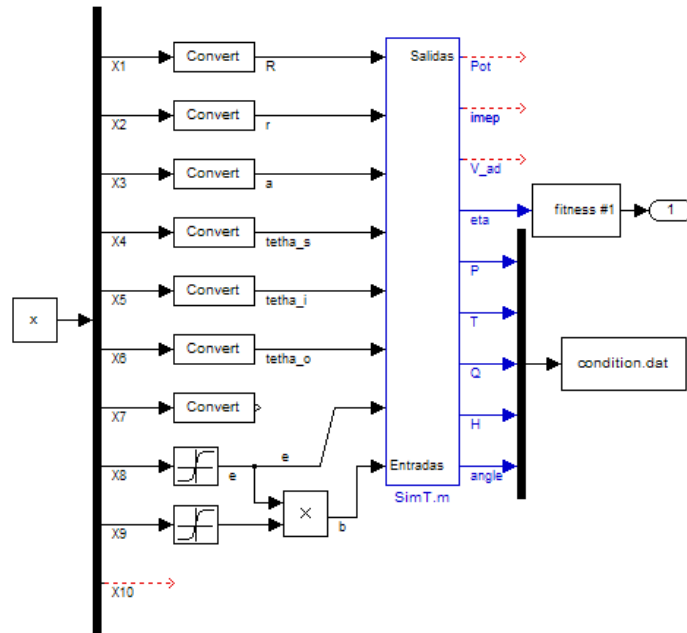
Como se observa, todas las ecuaciones cumplen con la forma  $y' = f(t, y)$ , donde la representación del tiempo se hace de manera indirecta mediante el ángulo de giro del cigüeñal. Los seis parámetros en cuestión corresponden a la variación de la presión, la temperatura en la zona de mezcla quemada, la temperatura en la zona de mezcla fresca, el trabajo mecánico efectuado, las pérdidas de calor por transferencia al ambiente y la pérdida de energía por concepto de fugas.

#### 2.4.4. Resultados del modelo de análisis termodinámico

La simulación arroja la variación de los seis parámetros de la ecuación 28 a lo largo de la duración del ciclo, a partir de 8 parámetros de entrada como se muestra en la figura 17 y que serán presentados en esta sección a partir de los cuales podemos obtener la información para cuantificar el rendimiento del motor, como son la potencia y la eficiencia térmica.

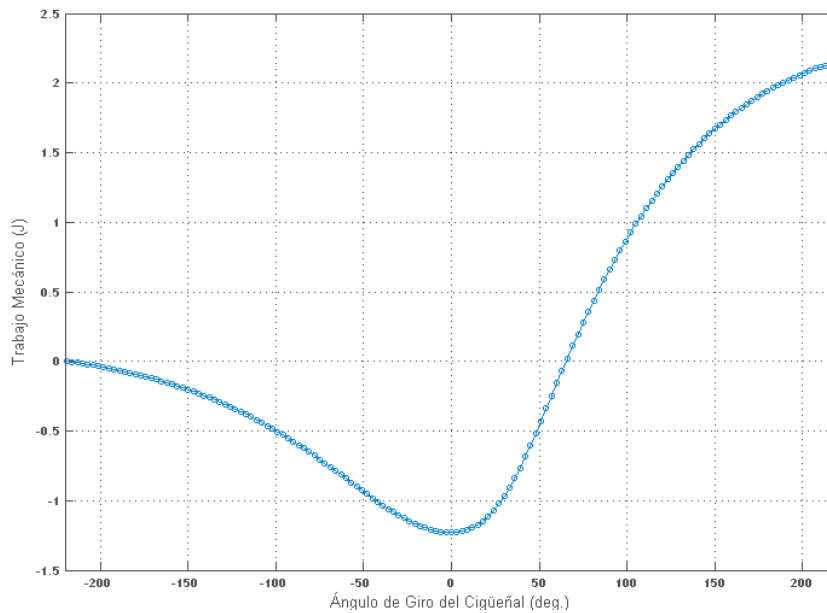
El primero (la potencia) es empleado en las restricciones del modelo para asegurarse que la solución planteada se acerque al objetivo planteado de 1.75 HP y el segundo (la eficiencia térmica) corresponde al primero de las dos funciones objetivos planteadas inicialmente y que busca ser maximizada con la variación de parámetros.

**Figura 17.** Relación de entradas y salidas para el análisis termodinámico



En la Figura 18 se puede observar la variación del trabajo mecánico total a medida que avanza el ciclo. Se observa el trabajo consumido en la etapa de compresión y después de la ignición el trabajo mecánico generado. Este valor al final del ciclo corresponde al trabajo total realizado durante el ciclo  $W_c$ .

**Figura 18.** Trabajo vs. Ángulo del Cigüeñal



La potencia generada se evalúa a partir de la Ecuación 29.

**Ecuación 29.** Potencia generada

$$\dot{W} = W_c \cdot RPM \cdot K$$

Donde “ $W_c$ ” es el trabajo realizado durante el ciclo,  $RPM$  el número de revoluciones por minuto del cigüeñal y “ $K$ ” un factor de conversión de unidades, distinto a la constante de la epitrocoide.

La eficiencia térmica se obtiene como una relación del trabajo obtenido por ciclo y la energía química requerida para llevar a cabo dicho ciclo, tal que:

**Ecuación 30.** Eficiencia térmica del ciclo

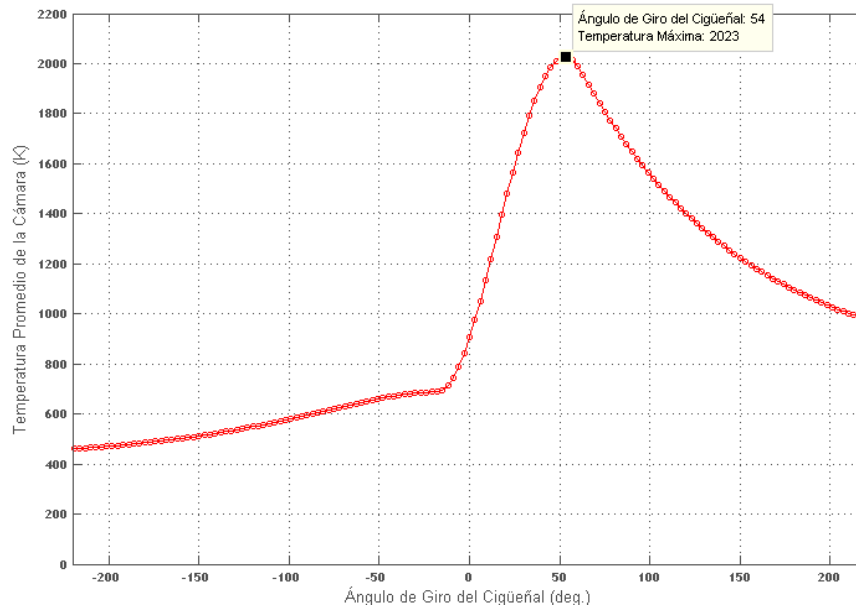
$$\eta_t = \frac{W_c}{m_f Q_{HHV}}$$

Donde, “ $\eta_t$ ” corresponde a la eficiencia térmica del ciclo, “ $m_f$ ” es la masa de combustible que entra por ciclo y “ $Q_{HHV}$ ” el poder calórico del combustible, para nuestro caso etanol.

La simulación emplea una serie de parámetros de naturaleza empírica que deben ser debidamente calibrados con información experimental cuya obtención se dificulta en un proyecto de esta naturaleza. Sin embargo, para asegurarnos de que los resultados son aceptables, es preciso comparar al menos las tendencias con los datos obtenidos por otros estudios para validar el modelo general. La calibración de parámetros del modelo queda propuesta para un trabajo posterior.

En la figura 19 se puede observar la progresión de las temperaturas de los gases al interior de la cámara a medida que se desarrolla el ciclo, este resulta ser uno de los datos fundamentales para el modelado de la transferencia de calor. Se observan temperaturas de alrededor de los 2000 K, que se encuentra dentro del rango para este tipo de simulaciones.

**Figura 19.** Temperatura de los gases vs. Ángulo del cigüeñal

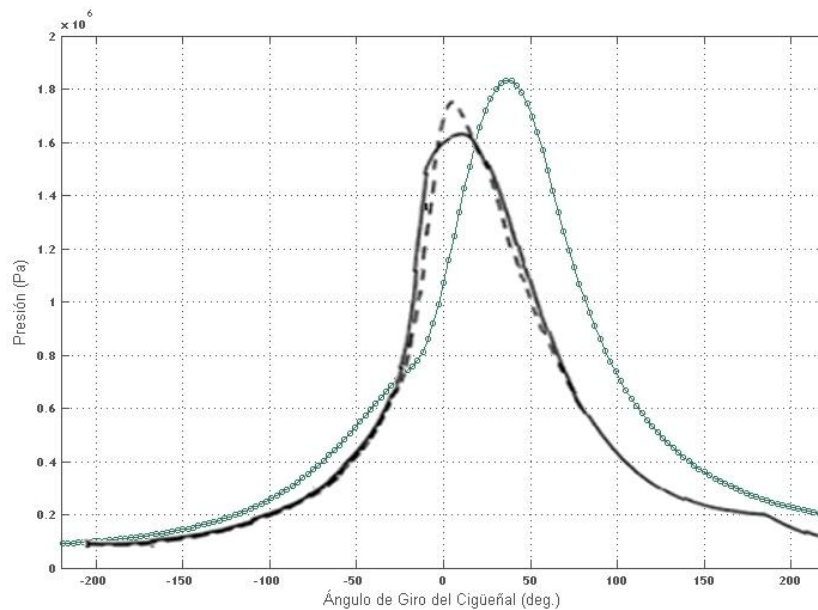


Tenemos presiones al interior de la cámara de combustión que van de los 20 hasta los 40 bares, muy acorde a las observaciones de diferentes autores para motores de prestaciones similares. Este valor está relacionado de manera muy directa con los parámetros que habíamos mencionado previamente luego solo sirve como referencia.

Una estimación del error entre el modelo matemático y los datos observados es citada por (Buttsworth, 2002) como de menos del 1% con respecto a trabajos previos del mismo tipo.

La figura 20 muestra la presión de la cámara versus el ángulo de giro del cigüeñal obtenidos de mediante la simulación del ciclo la comparación con datos experimentales que son parte del trabajo de (Danieli, 1976) resaltados en negro. Cualquier comparación cuantitativa requiere necesariamente de la calibración de parámetros para que sean verdaderamente representativos, pero por lo menos nos sirve para comparar tendencias.

**Figura 20.** Resultados de la rutina versus Datos medidos por (Danieli, 1976) - Presión vs. Ángulo del cigüeñal



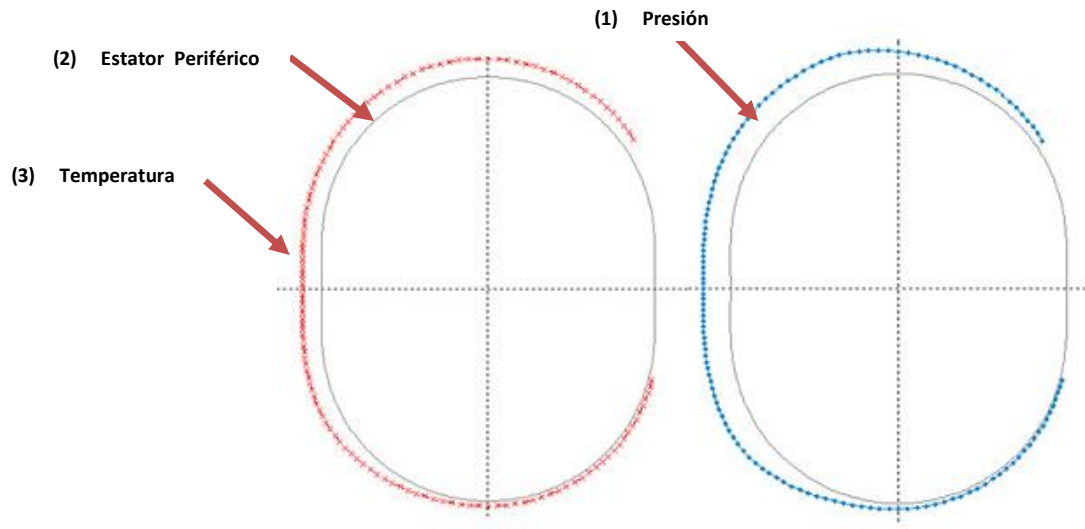
Los valores arrojados por la simulación están directamente relacionados con un punto específico del rotor, de modo que estos valores son modificados de forma tal que la carcasa se encuentre sometida a los valores de presión y temperatura máximos de los gases con los que entraría en contacto durante el funcionamiento normal del motor.

Podemos observar la distribución espacial de las condiciones de frontera de manera más fácil en la figura 21. Para más detalles del funcionamiento del ciclo véase el apéndice 3.

Estos parámetros arreglados – la presión  $P$ , la temperatura  $T$ , el flujo de calor por unidad de área  $Q$ , el coeficiente de transferencia de calor por convección  $H$  y la posición angular - son almacenados en un archivo de texto tipo ASCII en la carpeta +ANSYS con el nombre de condition.dat para ser leídos posteriormente.

Los resultados de la simulación arrojan valores de eficiencia térmica que van desde el 16% para algunos casos hasta cerca del 27% para las mejores soluciones.

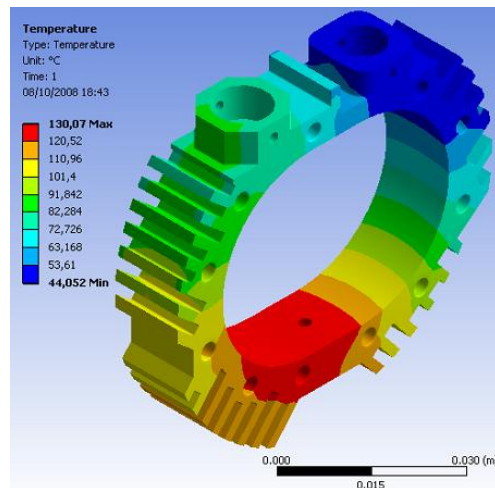
**Figura 21.** Distribución de presiones y temperaturas en la carcasa



## 2.5. DISEÑO DE LA CARCASA Y EL SISTEMA DE REFRIGERACIÓN.

El sistema de refrigeración por aletas radiales propuesto por (Rivera & Meneses, 2008), donde se puede identificar claramente dos zonas que utilizan paso entre aletas diferente para compensar la distribución de las temperaturas en la carcasa. El sistema propuesto por ellos se encuentra basado en buena medida en el sistema de refrigeración empleado en el motor OS 49 PI como se muestra en la Figura 22.

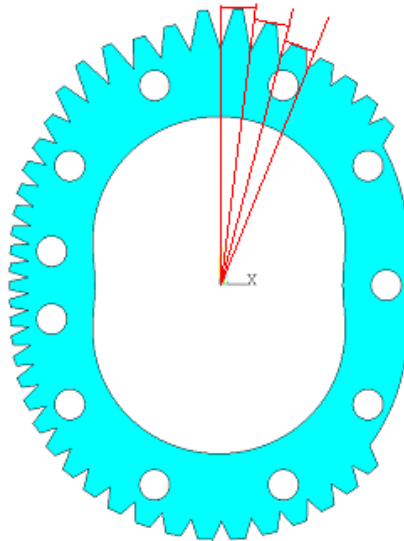
**Figura 22.** Motor 49PI tipo II de la O.S [osengines.com]<sup>15</sup>



<sup>15</sup> Imagen generada por (Rivera & Meneses, 2008) empleando ANSYS Workbench

El autor propone una modificación al sistema, manteniendo una separación angular entre aletas constante y emplear una altura variable para cada aleta, como se muestra en la Figura 23 se observa que las aletas se encuentran igualmente distribuidas a lo largo de la periferia del rotor, pero su longitud varia de una aleta a la otra.

**Figura 23.** Sistema de refrigeración propuesto



El sistema de refrigeración debe ser diseñado de manera que la transferencia de calor permita que la pared interior de la carcasa se encuentre a una temperatura de  $180^{\circ}\text{C}$ <sup>16</sup> esta disposición nos facilita el diseño de las aletas.

Esta propuesta nos permite emplear un modelo continuo, primero para el espesor mínimo de la sección transversal, y luego para la altura óptima de las aletas, para cada posición angular en función de la presión y temperatura; Para el espesor mínimo se consideraron las formulas para recipientes cilíndricos sometidos a presión interna, resultando que los esfuerzos transversales y longitudinales para la presión que manejamos no son críticos, luego es preciso otro criterio para su diseño, como el espacio requeridos para los sellos en las tapas laterales. Para las aletas utilizamos los modelos de aleta longitudinal de perfil trapecial propuestos en (Mills, 1999). Ver Apéndice 4.

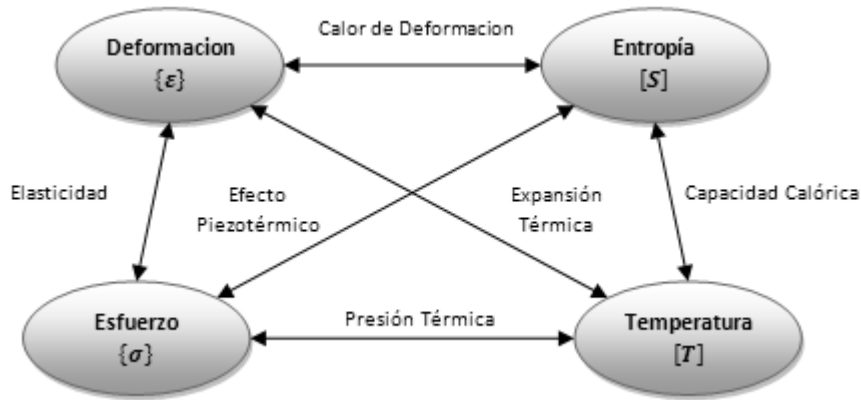
---

<sup>16</sup> Esta condición de emplea para asegurarse que el sistema de refrigeración cumpla con las consideraciones efectuadas en el análisis termodinámico.

### 2.5.1. Análisis Termoelástico

Las ecuaciones básicas para explicar el comportamiento termoelástico pueden ilustrarse mediante un ejemplo aplicado a un cristal único tomado de (Nye, 1957). Si observamos la Figura 24 tenemos las relaciones entre las 4 variables independientes en cuestión.

Figura 24. Elementos que interactúan en el análisis termoelástico



El planteamiento de (Nye, 1957) sugiere que tomemos como variables independientes los nueve componentes del estado de esfuerzos  $\sigma_{ij}$  y el valor de la temperatura  $T$ . Lo que quiere decir que los componentes de la deformación  $\xi_{ij}$  y la entropía  $S$  son determinados cuando estas variables son especificadas.

**Ecuación 31.** Derivada de la deformación para el comportamiento termoelástico

$$d\xi_{ij} = \left( \frac{\partial \xi_{ij}}{\partial \sigma_{kl}} \right)_T d\sigma_{kl} + \left( \frac{\partial \xi_{ij}}{\partial T} \right)_\sigma dT$$

Donde el primer termino tiene en cuenta los efectos el comportamiento elástico propio del material, mientras que el segundo corresponde al la expansión térmica. De manera análoga para la entropía tenemos que:

**Ecuación 32.** Derivada de la entropía para el comportamiento termoelástico

$$dS = \left( \frac{\partial S}{\partial \sigma_{kl}} \right)_T d\sigma_{kl} + \left( \frac{\partial S}{\partial T} \right)_\sigma dT$$

El primer término corresponde al efecto piezocalorico y el segundo al calor específico. Para ambas ecuaciones consideramos valida la siguiente notación:

**Ecuación 33.**

$$\left(\frac{\partial \xi_{ij}}{\partial \sigma_{kl}}\right)_T d\sigma_{kl} = \sum_{k=1}^3 \sum_{l=1}^3 \left(\frac{\partial \xi_{ij}}{\partial \sigma_{kl}}\right)_T d\sigma_{kl} = \left(\frac{\partial \xi_{ij}}{\partial \sigma_{11}}\right)_T d\sigma_{11} + \left(\frac{\partial \xi_{ij}}{\partial \sigma_{12}}\right)_T d\sigma_{12} + \dots$$

Como  $i$  y  $j$  pueden tomar valores de 1 a 3, en realidad tendríamos un conjunto de hasta 9 ecuaciones lineales, cada una con hasta 10 términos en el lado derecho.

A partir de la siguiente relación termodinámica:

**Ecuación 34**

$$\left(\frac{\partial \xi_{ij}}{\partial T}\right)_\sigma = \left(\frac{\partial S}{\partial \sigma_{kl}}\right)_T$$

Y sustituyendo las derivadas parciales por su equivalente físico podemos reescribir las ecuaciones Ecuación 31 y Ecuación 32 de manera compacta como:

**Ecuación 35.** Representación tensorial de las ecuaciones acopladas

$$\{\xi\} = [D]^{-1}\{\sigma\} + \{\alpha\}\Delta T \quad \bar{S} = \{\alpha\}^T\{\sigma\} + \frac{\rho C_p}{T_0}\Delta T$$

El parámetro  $\bar{S}$  corresponde a la densidad de entropía,  $\rho$  a la densidad y  $C_p$  al calor específico del material de la carcasa,  $T_0$  a la temperatura de referencia y  $\Delta T$  a la variación de la temperatura local entre dos nodos.

Donde el vector de deformación, el tensor de esfuerzos y el vector de coeficientes de expansión térmica son iguales a:

**Ecuación 36.** Tensor de deformación y tensor de esfuerzos

$$\begin{aligned} \{\xi\} &= [\xi_x \quad \xi_y \quad \xi_z \quad \xi_{xy} \quad \xi_{yz} \quad \xi_{xz}]^T \\ \{\sigma\} &= [\sigma_x \quad \sigma_y \quad \sigma_z \quad \sigma_{xy} \quad \sigma_{yz} \quad \sigma_{xz}]^T \\ \{\alpha\} &= [\alpha_x \quad \alpha_y \quad \alpha_z \quad 0 \quad 0 \quad 0]^T \end{aligned}$$

La matriz  $[D]^{-1}$  se conoce como la matriz de elasticidad definida como la inversa de la matriz de rigidez. El planteamiento general para un material anisotrópico se muestra en la Ecuación 37.

**Ecuación 37.** Matriz de elasticidad

$$[D]^{-1} = \begin{bmatrix} 1/E_x & -v_{xy}/E_x & -v_{xz}/E_x & 0 & 0 & 0 \\ -v_{yx}/E_y & 1/E_y & -v_{yz}/E_y & 0 & 0 & 0 \\ -v_{zx}/E_z & -v_{zy}/E_z & 1/E_z & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/G_{xy} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/G_{yz} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/G_{xz} \end{bmatrix}$$

Donde  $E_i$  corresponde al modulo de Young en la dirección  $i$ ,  $v_{ij}$  al número de Poisson y  $G_{ij}$  el modulo de resistencia a corte en el plano  $ij$ .

Si expresamos la entropía en función del calor  $Q$  empleando la segunda ley de la termodinámica para un proceso reversible, tenemos que:

**Ecuación 38.**

$$Q = T_0 S$$

Sustituyendo este término en las ecuaciones acopladas, tenemos que:

**Ecuación 39.**

$$\{\sigma\} = [D]\{\xi\} - \{\beta\}\Delta T \quad Q = T_0\{\beta\}^T\{\xi\} + \rho C_V \Delta T$$

Donde para simplificar un poco la ecuación se definieron los siguientes términos, el vector de coeficientes termoelásticas  $\{\beta\}$  y el calor específico a volumen o deformación constante  $C_V$ :

**Ecuación 40.**

$$\{\beta\} = [D]\{\alpha\} \quad C_V = C_p - \frac{T_0}{\rho}\{\alpha\}^T\{\beta\}$$

Por último sustituyendo el término  $Q$  en la ecuación de flujo de calor tenemos que:

**Ecuación 41.**

$$\frac{\partial Q}{\partial t} = T_0\{\beta\}^T \frac{\partial\{\xi\}}{\partial t} + \rho C_V \frac{\partial \Delta T}{\partial t} - [K]\nabla^2 T$$

Quedaría pendiente definir dos relaciones restantes, la relación entre  $\xi_{ij}$  y  $S$ , y la relación entre  $\sigma_{ij}$  y  $T$ . El primer efecto está relacionado con el calor generado por una

deformación mecánica, mientras que el segundo tiene que ver con el esfuerzo generado por el cambio de temperatura. La relación de estos efectos puede obtenerse a partir de la energía libre de Helmholtz “ $\Psi$ ”.

**Ecuación 42.** Energía libre de Helmholtz

$$\Psi = U - TS \qquad d\Psi = \sigma_{ij} d\xi_{ij} - SdT$$

A partir de esta ecuación y con la condición de que sea una diferencial perfecta podemos obtener la última relación termodinámica que nos interesa.

**Ecuación 43.**

$$-\frac{\partial^2 \Psi}{\partial \xi_{ij} \partial T} = \left( \frac{\partial S}{\partial \xi_{ij}} \right)_T = - \left( \frac{\partial \sigma_{ij}}{\partial T} \right)_\xi = f_{ij}$$

Una vez se define la Ecuación 41 sólo resta aplicar el principio variacional a las ecuaciones para obtener una expresión matricial de elementos finitos. Para revisar este tema en detalle se recomienda al lector consultar la guía de referencia y la documentación incluida en el paquete de ANSYS (ANSYS, Inc., 2005).

### 2.5.2. Selección del elemento PLANE223

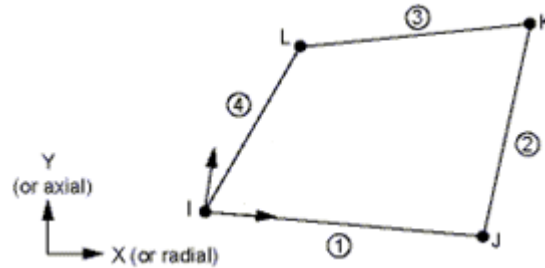
Un análisis de campos acoplados – *Multi-Field Analysis* – es una combinación de análisis de diferentes disciplinas o campos físicos de la ingeniería, que interactúan para resolver un problema específico, también conocido a menudo como análisis multifísico.

Para un análisis termoelástico, donde interactúan esfuerzos térmicos y mecánicos, tenemos un acople en una sola dirección donde la deformación térmica influye directamente en el análisis estructural, pero la deformación mecánica no afecta la distribución de temperaturas, luego el análisis no requiere de iteración entre los dos campos físicos. Hay dos maneras de efectuar un análisis multi-físico, el *acople directo* y la *transferencia de cargas*.

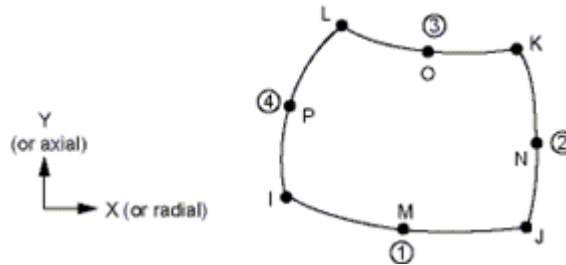
El *método directo* por lo general involucra un solo análisis con un elemento de tipo combinado que contenga los grados de libertad necesarios, y el elemento contiene las

ecuaciones requeridas para el acople. Los elementos más comunes para este tipo de análisis son los elementos PLANE13 y PLANE223, con 4 y 6 nodos respectivamente como se muestra en las Figura 25 y Figura 26, donde se observa la numeración local de los nodos y el sistema de coordenadas global.

**Figura 25.** Elemento PLANE13, imagen tomada de (ANSYS, Inc., 2005)



**Figura 26.** Elemento PLANE223 imagen tomada de (ANSYS, Inc., 2005)



Haciendo una comparación cuantitativa solamente en cuanto a rendimiento en tiempo de procesamiento tenemos que el elemento PLANE223 arroja resultados un 40% más rápido que su contraparte de 4 nodos.

El elemento PLANE223 soporta tanto la expansión térmica como los efectos piezocalóricos y emplean un método de acople matricial fuerte, mientras que el elemento PLANE13 solo soporta los efectos de la expansión térmica en forma de un vector de carga, lo que resulta en un acople débil, cosa que podría explicar el rendimiento inferior a pesar de poseer un número de nodos menor.

Los métodos de *transferencia de cargas* involucran dos o más análisis, cada uno de un campo diferente y se acoplan los campos al aplicar los resultados de un análisis como cargas en el otro. Existen diferentes formas de lograr esto. Una de las maneras más sencillas de lograr la transferencia es transferir directamente las cargas de un análisis al siguiente. Para esto se crean 2 archivos donde se definen por separado los dos

análisis, con tipos de elementos, tipo de malla y criterios de convergencia diferentes, luego se procede a solucionar el primer análisis, en nuestro caso la transferencia de calor, leer los resultados y transferir las cargas mediante el comando LDREAD<sup>17</sup> y posteriormente resolver el segundo análisis, el estructural.

La desventaja que tiene la transferencia de cargas es que es preciso generar el modelo una vez para cada uno de los análisis, luego se pierde tiempo en el proceso de generar la geometría. Para análisis que no exhiben un alto grado de interacción no-lineal, el método de transferencia de cargas resulta más eficiente y flexible porque se pueden efectuar los análisis de manera independiente, pero en general es preciso definir mas detalles y especificar manualmente las cargas a ser transferidas. Esto naturalmente favorece la selección del acople directo con el elemento PLANE223 para resolver el análisis termoelástico.

### 2.5.3. Perfil de Temperaturas y Estado de Esfuerzos

Para el diseño de la carcasa se decidió emplear aleaciones de aluminio por sus propiedades térmicas y por tener una densidad significativamente menor a la del acero. Las propiedades térmicas y mecánicas de las aleaciones contempladas se encuentran descritas a continuación en la Tabla 2.

**Tabla 2.** Propiedades Térmicas y Mecánicas de las aleaciones contempladas para la carcasa.

Propiedad		Material Propuesto			
		1060-O	2017-O	2218-T61	2618-T61
<b>DENS</b>	Densidad $\left[\frac{kg}{m^3}\right]$	2698	2810	2795	2810
<b>NUXY</b>	Número de Poisson	0.33	0.33	0.33	0.33
<b>EX</b>	Modulo de Elasticidad $[GPa]$	70-80	70-80	70-80	70-80
<b>ALPHX</b>	Expansión Térmica $\left[\frac{\mu m}{m^{\circ}C}\right]$	23.58	22.32	22.32	22.32
<b>KXX</b>	Conductividad Térmica $\left[\frac{W}{m^{\circ}C}\right]$	234	147	154	147
<b>S<sub>y</sub></b>	Esfuerzo de Cedencia $[MPa]$	76	370	305	370

<sup>17</sup> LDREAD es el comando de ANSYS para leer un archivo externo

Como se observa en la Tabla 2, se consideraron 4 diferentes aleaciones que se listan a continuación:

- *1060-O*: Es un aluminio de alta pureza, propiedades mecánicas bastante modestas pero posee la mejor conductividad térmica de las opciones contempladas.
- *2017-O*: Se emplea en aplicaciones que requieran de alta dureza y resistencia a temperaturas elevadas. Comúnmente empleado en estructuras, marcos, chasises.
- *2218-T6*: Se emplea en cabezas de pistones para motores diesel y a gasolina, posee excelentes propiedades mecánicas.
- *2618-T6*: Al igual que el 2218, este se emplea en cabezas de pistones y partes rotativas en motores de combustión interna en la industria aeroespacial.

Para determinar la temperatura en la base de las aletas empleamos un análisis de transferencia de calor bidimensional en la sección transversal de la carcasa, tomando como condiciones de frontera una temperatura superficial interior de 180° Celsius y una velocidad de flujo de calor dada por la simulación del ciclo termodinámico para la posición angular en cuestión.

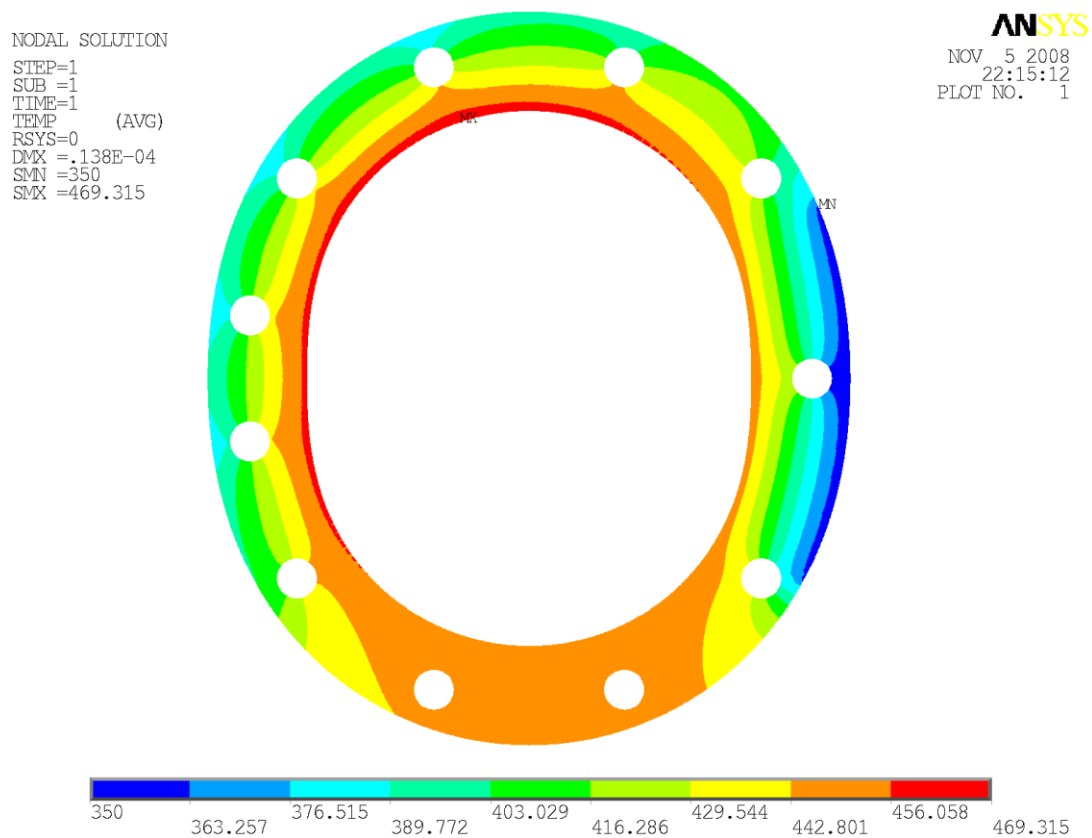
El modelo en cuestión se genera de manera paramétrica a partir de una rutina programada para ANSYS en el lenguaje APDL, que se encarga de generar la sección transversal, el enmallado, la solución, el posprocesamiento y almacena los resultados de la temperatura superficial versus posición angular en un formato tipo ASCII para que pueda ser leído posteriormente por MATLAB en la rutina desarrollada por el autor.

Como la carcasa se encuentra sometida a cargas térmicas y mecánicas se opta por realizar un análisis termo-elástico que nos permite de paso verificar el estado de esfuerzos de la carcasa, para lo cual se opta por emplear un elemento Plane223, que tienen desplazamientos y temperaturas como grados de libertad, promediando unos

4200 nodos para la pieza, tendríamos alrededor de 12600 grados de libertad en esta prueba.

En la Figura 27 observamos la distribución de temperaturas en la carcasa, de la cual obtenemos la temperatura en la base de las aletas para el diseño, de manera análoga la Figura 28 nos permite observar los efectos de esta distribución de temperaturas desigual en la carcasa en los esfuerzos asociados a la deformación térmica y su efecto combinado con la presión de los gases en la cámara de combustión, de modo que podemos comparar el esfuerzo resultante para determinar el factor de seguridad de la carcasa, en caso de obtenerse un factor de seguridad inferior a 1.2, aplicaremos un penalizador al diseño para favorecer a los diseños que cumplan con el criterio de esfuerzos.

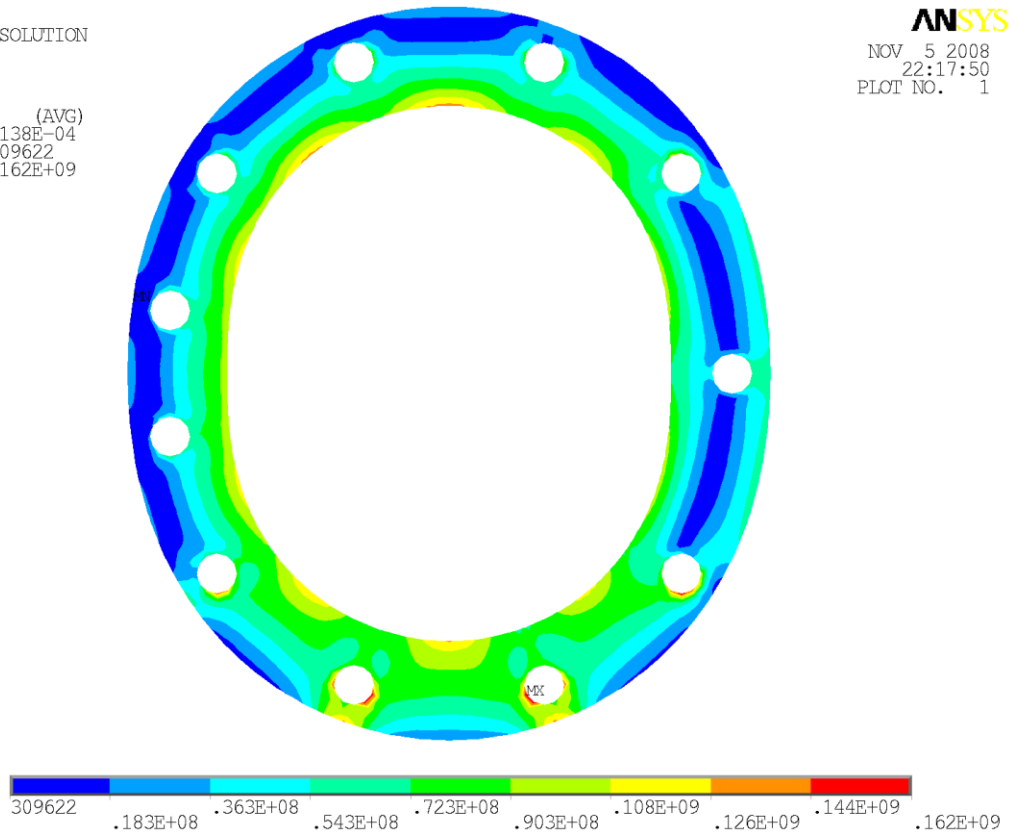
**Figura 27.** Distribución de temperaturas en la carcasa, datos en grados Kelvin.



**Figura 28.** Esfuerzo equivalente en la carcasa, datos en Pascales

NODAL SOLUTION  
STEP=1  
SUB =1  
TIME=1  
SEQV (AVG)  
DMX =.138E-04  
SMN =309622  
SMX =.162E+09

**ANSYS**  
NOV 5 2008  
22:17:50  
PLOT NO. 1



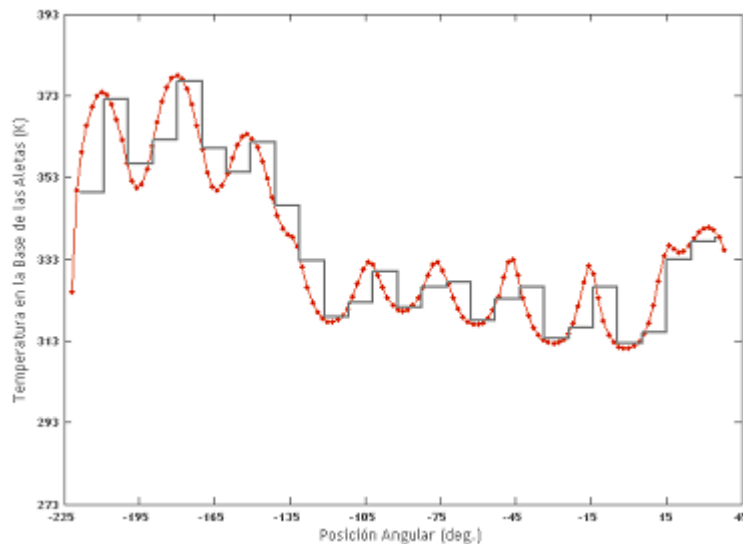
#### 2.5.4. Diseño de las Aletas Radiales

Las aletas son las encargadas de evacuar el calor del sistema al ambiente, y de su diseño adecuado dependen dos cosas. Primero, que la superficie interna de la carcasa se mantenga a una temperatura no superior a los 180° Celsius, pero tan cercana como sea posible a esta. Y segundo, de ella depende en buena medida el peso resultante de la carcasa.

De los resultados del análisis por elementos finitos obtenemos un vector de temperaturas en la superficie externa de la carcasa para el posterior diseño del sistema de refrigeración. Se introduce un parámetro adicional " $n_a$ ", que controlaría el número de aletas radiales del sistema de refrigeración.

Como el modelo de las aletas no es continuo es preciso tomar valores promedio de la temperatura en la base y el flujo de calor para cada aleta, como se puede observar en la Figura 29.

Figura 29. Temperatura en la base de las aletas



En la Figura 29<sup>18</sup> se puede observar en rojo la variación de la temperatura externa a lo largo de la carcasa. Dependiendo del número de aletas, se divide la carcasa en partes

<sup>18</sup> Imagen generada empleando MATLAB R2008a

iguales y se procede a calcular el valor promedio de la temperatura en esa sección. Esa temperatura promedio será la temperatura en la base de las aletas para cada una de las secciones, y se representa mediante un diagrama de saltos en línea negra. El diseño de las aletas se puede observar con mayor detalle en el apéndice 4.

La Ecuación 44 muestra la relación entre el flujo de calor que pasa por la aleta y sus características geométricas. En nuestro caso tenemos como restricción la temperatura en la base de las aletas y el flujo de calor por unidad de área, mientras que resaltado en rojo se encuentra el parámetro geométrico que define la longitud de la aleta.

**Ecuación 44.** Eq. Característica para las aletas trapezoidales

$$\dot{Q} = -kS_b n(T_b - T_a) \frac{K_1(2n\sqrt{x_e})I_1(2n\sqrt{L}) + I_1(2n\sqrt{x_e})K_1(2n\sqrt{L})}{K_1(2n\sqrt{x_e})I_0(2n\sqrt{L}) + I_1(2n\sqrt{x_e})K_0(2n\sqrt{L})}$$

Donde “ $I_0$ ” es la función de Bessel modificada de primera especie y orden cero, y “ $K_0$ ” la función de Bessel modificada de segunda especie y orden cero, “ $k$ ” la conductividad térmica de la carcasa, “ $S_b$ ” el área transversal de la base de la aleta, “ $T_\infty$ ” la temperatura ambiental y “ $n$ ” relaciona los parámetros geométricos con la transferencia de calor, como se muestra en la Ecuación 45.

**Ecuación 45.**

$$n^2 = \frac{h_c}{k} \frac{2L}{bt} \sqrt{1 + \left(\frac{t}{2L}\right)^2}$$

Donde “ $t$ ” es el ancho de la base de la aleta, “ $b$ ” es el ancho de la carcasa y “ $h_c$ ” corresponde al coeficiente de transferencia por convección externa, estimado en 160.

Como no es posible obtener una expresión sencilla para la ecuación característica para el diseño se optó por mantener constante el valor de “ $x_e$ ” e igual a 3 milímetros y emplear una aproximación iterativa al valor más adecuado de “ $L$ ”. Esto se hace estimando una desviación o un error “ $e$ ” entre ambos lados de la ecuación para diferentes valores de “ $L$ ” y procurar minimizar dicha desviación.

Con el propósito expreso de facilitar su manipulación se definen los parámetros “ $C_1 \dots C_5$ ” de modo que sea posible reescribir la Ecuación 44 como se muestra:

**Ecuación 46.** Eq. Característica de las aletas trapezoidales modificada

$$C_1 = \frac{C_2 K_1(X^*) + C_3 I_1(X^*)}{C_4 K_1(X^*) + C_5 I_1(X^*)} + e$$

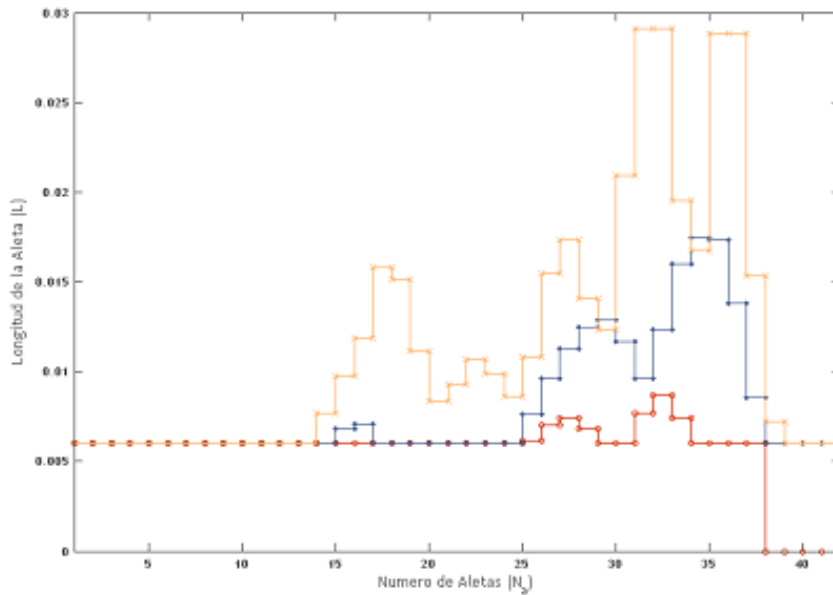
$$X^* = 2n\sqrt{x_e} \quad C_1 = \frac{\dot{Q}}{-kS_b n(T_b - T_a)} \quad C_2 = I_1(2n\sqrt{L})$$

$$C_3 = K_1(2n\sqrt{L}) \quad C_4 = I_0(2n\sqrt{L}) \quad C_5 = K_0(2n\sqrt{L})$$

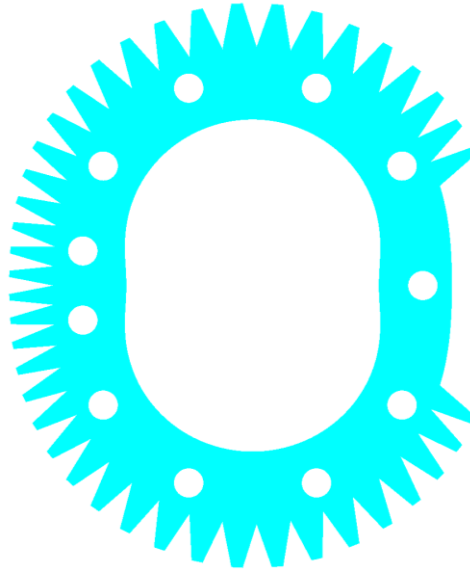
Se estima el error para diferentes longitudes y se determina la longitud más adecuada para cada una de las aletas en intervalos de un milímetro.

En la Figura 30 se muestran los resultados obtenidos para tres ejemplos distintos con diferente número de aletas, cada uno diferenciado por colores. Se observa que en todos los casos, las aletas son más largas hacia el lado izquierdo de la grafica, lo que correspondería a la zona caliente del motor. Al igual que las condiciones de frontera, resulta mucho más práctico mostrar la distribución de las aletas sobre la geometría de la carcasa, como se muestra.

**Figura 30.** Longitud de las aletas para diferentes escenarios propuestos

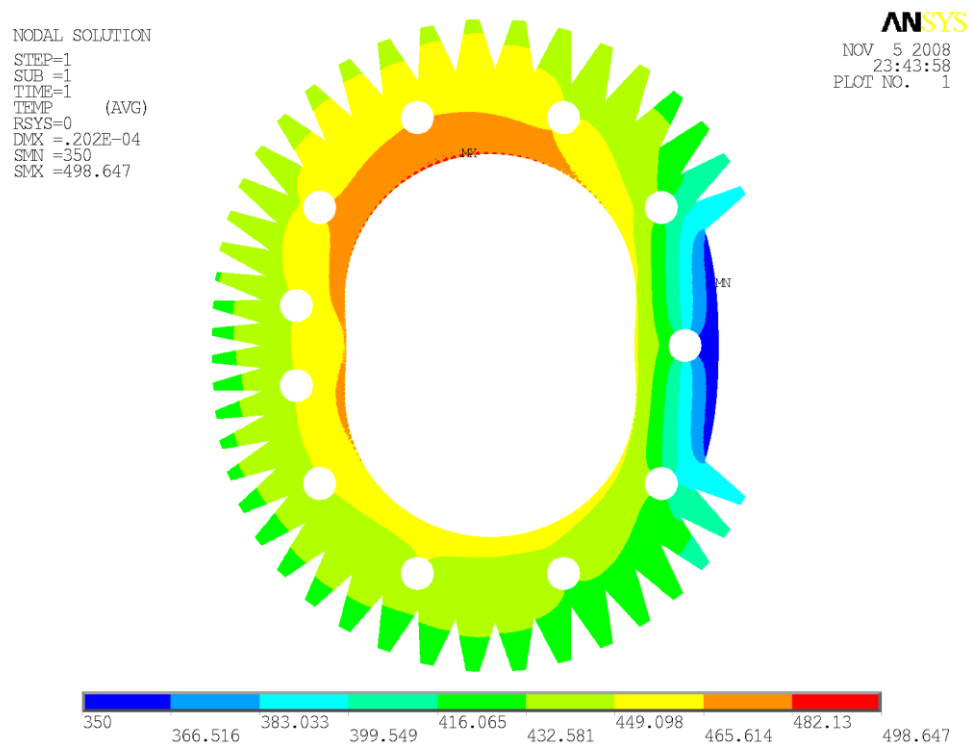


**Figura 31.** Distribución de las aletas a lo largo de la carcasa



Luego procedemos a la validación del diseño de las aletas con ANSYS – este paso es omitido en el algoritmo principal para reducir el coste computacional – y obtener un estado final de esfuerzos y temperaturas como se muestra en la figura 32.

**Figura 32.** Validación del diseño, perfil de temperaturas.



En la figura 32 podemos observar la distribución de temperaturas en la carcasa para un individuo seleccionado de manera aleatoria. Como era de esperarse la distribución de la temperatura es desigual y se observa que las temperaturas son más altas en la zona donde se lleva a cabo la combustión. Además se observa que la variación de la temperatura es poca, debido a la alta conductividad térmica del aluminio, cuando la tendencia en estos motores es la utilización de recubrimientos cerámicos.

## 2.6. DEFINICIÓN DE LA FUNCIÓN PESO

El peso o la masa para ser más precisos del motor es el segundo criterio de control para evaluar la calidad del candidato solución, para simplificar los cálculos solo tomaremos en consideración los elementos que componen el conjunto estator-rotor.

Los elementos tomados en consideración para la estimación de la masa son:

- Rotor.
- Estator Periférico o Carcasa.
- Engranaje Interno.
- Rodamiento Interno.

La Ecuación 47 determina la masa del conjunto rotor-estator a partir de los elementos individuales, y compone la segunda función objetivo.

**Ecuación 47.** Función Masa del motor

$$f_2(\vec{x}) = M = M_c + M_r + M_e + M_{rd}$$

Donde “ $M$ ” es la masa del conjunto rotor-estator, “ $M_c$ ” corresponde a la masa de la carcasa, “ $M_r$ ” a la del rotor, “ $M_e$ ” a la del engranaje y “ $M_{rd}$ ” a la del rodamiento respectivamente.

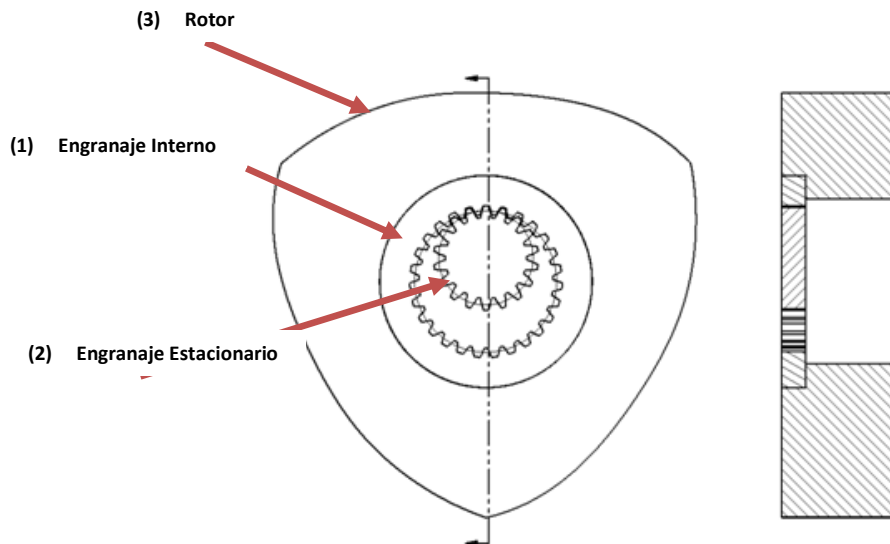
Los datos de la carcasa los obtenemos directamente del diseño de la carcasa, tomando el área de la sección transversal “ $A_c$ ”, el espesor de la carcasa “ $b$ ” y la densidad del material “ $\rho_c$ ”, como se muestra en la ecuación 48.

**Ecuación 48.** Masa de la carcasa

$$M_c = V_c \rho_c = A_c b \rho_c$$

La Figura 33 muestra un predimensionamiento del rotor en función de sus dimensiones más elementales y podemos observar que la masa del rotor corresponde a la proyección del área bajo la curva que compone al rotor (1) menos el espacio ocupado por el engranaje interno (2) y por el rodamiento, luego es preciso en primera instancia la selección o al menos una aproximación geométrica de estos componentes.

**Figura 33.** Forma simplificada del rotor. Dimensiones del Rodamiento del Rotor



### 2.6.1. Dimensiones del Rodamiento del Rotor

El trabajo realizado por (Rivera & Meneses, 2008) sugiere la utilización de rodamientos de rodillos de agujas y parten de una recomendación de diseño que consiste en que el diámetro del cigüeñal en la sección de acople debe ser igual o cercana a 5 veces la excentricidad  $e$ .

A su vez (Yamamoto, 1981) plantea una restricción geométrica al alojamiento del rodamiento del rotor, tal que debe cumplirse la siguiente condición:

**Ecuación 49.** Restricción al alojamiento del rodamiento

$$d_{rd} \geq d_e + 2e$$

Donde, “ $d_{rd}$ ” es el diámetro del alojamiento o de la pista externa del rodamiento y “ $d_e$ ” el diámetro del cigüeñal en esa sección en cuestión.

Lo que se plantea es generar una base de datos con rodamientos de agujas dentro del rango que varía la excentricidad y de la cual se pueda extraer información relevante como diámetro de la pista externa, diámetro del eje, ancho, velocidad de rotación límite y el peso.

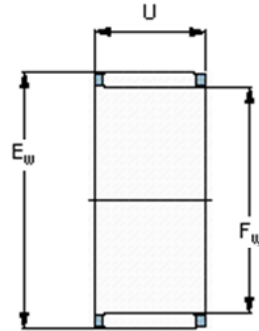
Los rodamientos tomados en consideración son mostrados en la Tabla 3.

**Tabla 3.** Rodamientos contemplados para el rotor, tabla tomada de SKF

Dimensiones (mm.)			Cargas (kN)			Vel (RPM)	Masa (kg.)	Referencia
$F_w$	$E_w$	$U$	$C$	$C_0$	$P_u$			
10	14	10	5.61	6.1	0.695	28000	0.0025	K 10x14x10 TN
10	14	13	5.83	6.3	0.72	28000	0.0046	K 10x14x13 TN
10	16	12	7.65	7.2	0.85	28000	0.0055	K 10x16x12 TN
12	17	13	9.13	10.4	1.22	26000	0.0049	K 12x17x13 TN
12	18	12	9.52	10	1.18	26000	0.006	K 12x18x12 TN
14	20	12	9.9	10.6	1.25	24000	0.0085	K 14x20x12
15	21	15	13.8	16.3	2	24000	0.011	K 15x21x15
15	21	21	18.7	24.5	3	24000	0.017	K 15x21x21
16	24	20	20.5	23.6	2.9	22000	0.022	K 16x24x20
18	28	16	18.7	18.6	2.28	19000	0.024	K 18x28x16
20	28	20	22.9	28.5	3.45	18000	0.027	K 20x28x20
20	28	25	29.2	39	4.9	18000	0.032	K 20x28x25
20	30	30	34.1	41.5	5.2	17000	0.049	K 20x30x30
23	35	16	24.2	23.2	2.9	15000	0.029	K 23x35x16 TN
25	35	30	44.6	62	7.8	15000	0.065	K 25x35x30

De manera que con el valor de la excentricidad procedemos a efectuar una búsqueda en tabla para determinar el menor de los rodamientos que cumple con las restricciones de diseño y obtener tanto la dimensión del alojamiento, como la masa del rodamiento, como se observa en la figura 34.

**Figura 34.** Dimensiones del rodamiento de agujas



### 2.6.2. Dimensiones del engranaje interno

El engranaje interno se diseña de acuerdo a las especificaciones de la norma AGMA<sup>19</sup> la verificación con esfuerzos por flexión como se muestra en el apéndice D.

**Ecuación 50.**

$$F = \frac{W_t P K_v K_o K_m K_s K_b}{\sigma_{adm} J}$$

Se aproxima el peso del rodamiento interno a un disco de diámetro exterior “ $d_e$ ” igual a 9 veces la excentricidad del rotor, el diámetro interno que corresponde a la corona y a el espesor obtenido con la verificación de esfuerzos. Para una explicación detallada de los parámetros, consultar el apéndice D.

Se modela que el engranaje se encuentra fabricado en un acero SAE AISI 8340, luego la masa del engranaje “ $M_e$ ” puede escribirse como:

**Ecuación 51.**

$$M_e = \frac{\pi}{4} \left( (9e)^2 - \left( 6e - \frac{7}{6}m \right)^2 \right) F \cdot \rho_{8340}$$

Donde, “ $\rho_{8340}$ ” es la densidad del acero SAE AISI 8340, “ $F$ ” es el ancho del rodamiento obtenido a partir de la Ecuación 50 y “ $m$ ” es el módulo del engranaje.

<sup>19</sup> American Gear Manufacturers Association, por sus siglas en inglés. Es la autoridad pertinente para la elaboración de estándares de diseño y análisis de engranajes

Si el ancho del rodamiento resulta mayor al ancho de la carcasa será preciso añadir una extensión para dar alojamiento completo al rodamiento, luego la masa del rotor podrá ser calculada mediante la Ecuación 52.

**Ecuación 52.**

$$M_r = \left\{ A_r b - \frac{\pi}{4} [(9e)^2 e + (7e)^2 (b - e)] + \frac{\pi}{4} (U - b - e) [(9e)^2 - (7e)^2] \right\} \rho_{rotor}$$

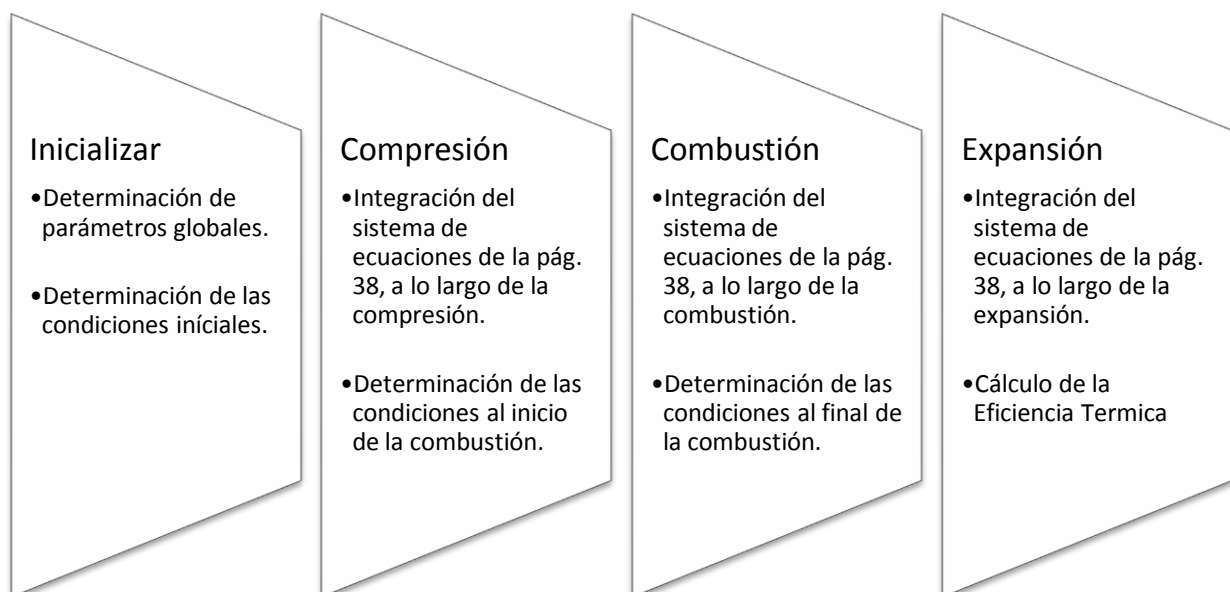
Donde, “ $\rho_{rotor}$ ” corresponde a la densidad del material seleccionado para el rotor, que en este caso es el mismo que el de la carcasa, y los demás parámetros que han sido utilizados a lo largo de la sección anterior.

## 2.7. REPRESENTACIÓN DE LAS FUNCIONES OBJETIVO

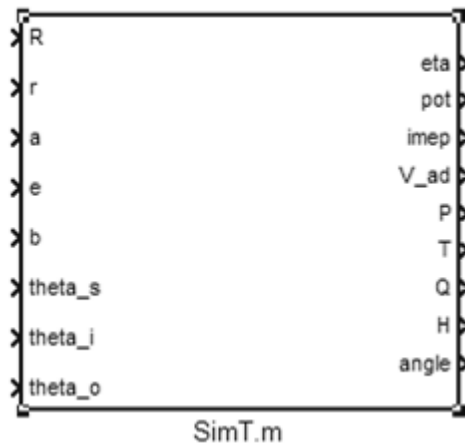
La primera función (eficiencia térmica), resulta particularmente difícil de representar de manera esquemática porque no sigue la estructura de un algoritmo convencional. Para ver el funcionamiento en detalle véase el apéndice 3.

Se puede desmenuzar a grandes rasgos en las etapas mostradas en la Figura 35, que complementa el diagrama de entradas y salidas de la Figura 36, donde se observa la eficiencia térmica como una de las salidas de la función *SimT.m*.

**Figura 35.** Función Objetivo 1 – Eficiencia Térmica

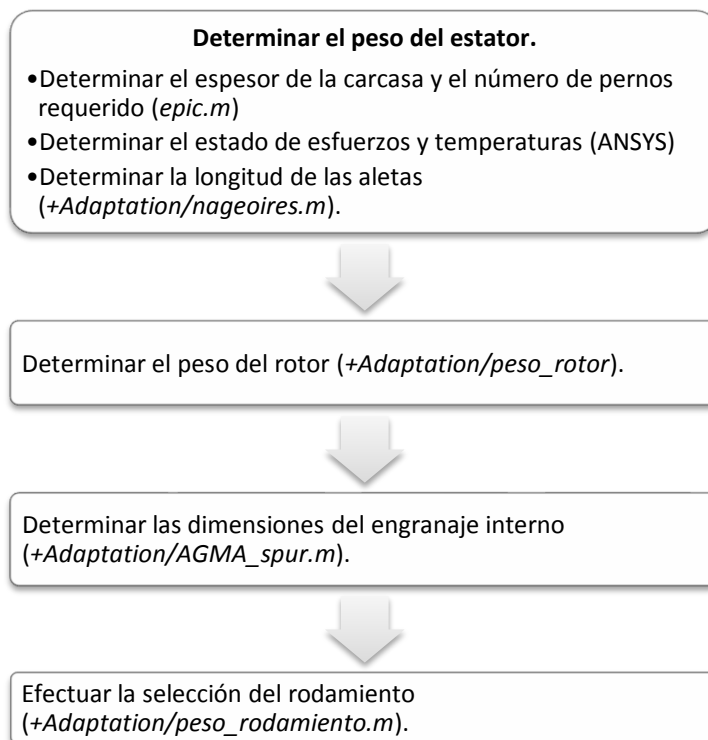


**Figura 36.** Diagrama I/O para la Función Objetivo 1



La segunda función objetivo (la masa del conjunto rotor-estator) es completamente secuencial y se obtiene a partir de los parámetros geométricos vistos anteriormente, los resultados del análisis en ANSYS y dos parámetros propios como son el número de aletas y el material de la carcasa. En la Figura 37 se muestra una representación esquemática del proceso de evaluación de la función objetivo 2.

**Figura 37.** Función Objetivo 2 – Masa del Conjunto Rotor-Estator



### 3. ALGORITMO DE OPTIMIZACIÓN

El objetivo del diseño óptimo es encontrar la combinación de parámetros que permitan minimizar una función objetivo, sujetos a una serie de restricciones de comportamiento. Cuya importancia fue reconocida inicialmente por la industria aeroespacial donde los diseños de las aeronaves son controlados por el peso.

Las técnicas analíticas y determinísticas son las herramientas que se emplean comúnmente para el análisis de sistemas en problemas de ingeniería. Sin embargo, la aplicación de dichas técnicas a problemas reales se dificulta en primera instancia, por el altísimo costo computacional de las mismas y por la presencia de ruido en el espacio de búsqueda.

En general podemos identificar tres grupos de técnicas de búsqueda que son empleadas en el diseño óptimo multidisciplinario que normalmente son problemas no lineales como son:

- *Gradient-based Methods*, o métodos de búsqueda basados en el gradiente.
- *Population-based Methods*, o métodos de búsqueda basados en una población.
- *Otros*, Búsqueda aleatoria, métodos termodinámicos, etc.

Los métodos basados en una población, como son los algoritmos genéticos, la optimización por partículas y la optimización memética, son apropiados para resolver problemas multiobjetivo para espacios de búsqueda no derivables, discontinuos y con mucho ruido.

El objeto de este proyecto no es hacer un estudio extensivo de la programación genética sino explorar la aplicabilidad de los métodos disponibles a problemas de ingeniería, y el autor propone que una comparación objetiva de los diferentes métodos disponibles para un mismo problema como un trabajo de grado complementario que daría una visión más amplia de las técnicas de optimización disponibles.

### 3.1. INTRODUCCIÓN A LOS ALGORITMOS GENÉTICOS<sup>20</sup>

Los Algoritmos Genéticos (GAs) son métodos de búsqueda adaptativa que busca de alguna manera simular los mecanismos observados en la naturaleza que determinan la supervivencia de las especies, en particular los principios de la selección natural y la genética. Los GAs son referidos a menudo como métodos de optimización global porque pueden explorar de manera efectiva espacios de solución muy amplios sin atascarse en mínimos locales.

#### 3.1.1. Principio de Funcionamiento

Los principios básicos de los GAs derivan de los principios de la selección natural observados inicialmente por Darwin. Las ideas de Darwin pueden resumirse en tres principios fundamentales:

- *Existe una población de individuos con diferentes habilidades y propiedades. Además existe un límite para el tamaño de la población.*

Hemos observado como las propiedades y habilidades de un individuo se encuentran caracterizadas por su 'fenotipo' y codificadas en su 'genotipo'. Debido a que los recursos de un sistema son limitados, el número de individuos de la población en cuestión tiene un tope máximo.

- *La naturaleza crea nuevos individuos con características similares a los individuos existentes.*

Los individuos de una población no permanecen iguales, sino que cambian con el tiempo. Los vástagos son similares a sus padres y heredan algunas de sus cualidades.

- *Los individuos prominentes son seleccionados más a menudo para transferir sus genes a generaciones futuras.*

---

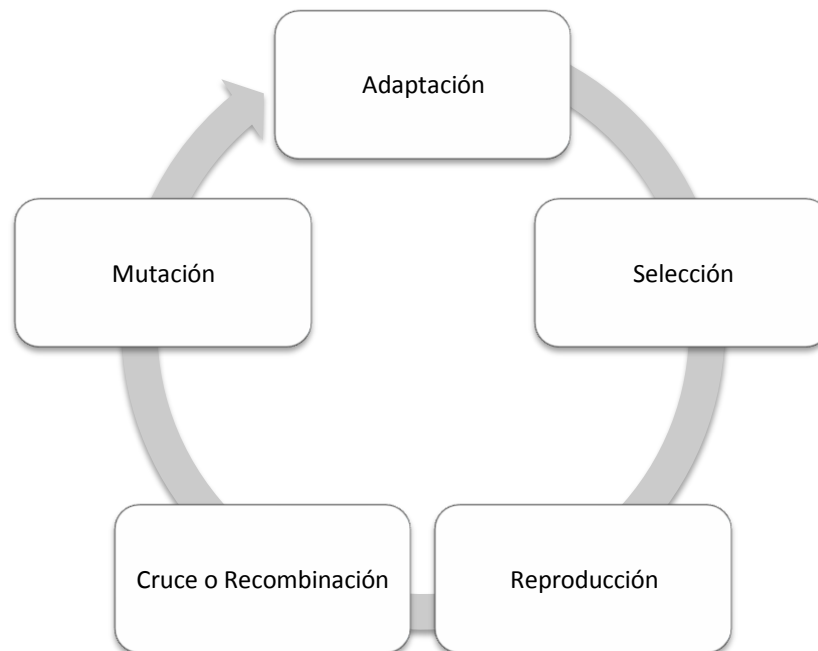
<sup>20</sup> Documentación obtenida (Pedregal, 1963) (Alvarez, 2000) (Gen & Runwei, 2000) (Rothlauf, 2006) (Sastry, 2007)

Por último, los individuos que poseen un valor de aptitud mayor, tienen una mayor probabilidad de reproducción que los demás individuos, por lo tanto los individuos más débiles serán removidos paulatinamente a medida que progresan las generaciones. Como la población es purgada se espera que la aptitud media de la población aumente con el tiempo.

### 3.1.2. Funcionalidad del Algoritmo Genético

Los pasos para evolucionar la solución una vez establecida la función aptitud, el tamaño de la población y la codificación de los parámetros pueden resumirse en el esquema de funcionamiento del mecanismo de búsqueda genética, como se muestra en la Figura 38.

Figura 38. Mecanismo de Búsqueda Genética



#### · *Inicialización:*

La población inicial de candidatos es por lo general seleccionada de manera aleatoria del espacio de búsqueda. Sin embargo, la utilización de información adicional como criterio de selección de la población inicial puede incorporarse fácilmente.

- *Evaluación:*

Una vez se ha inicializado la población o se ha generado una nueva población es preciso evaluar la función objetivo de cada uno de los candidatos.

- *Selección:*

El criterio de selección favorece la reproducción de los candidatos con una mejor función de aptitud y por tanto la supervivencia del más apto, como mecanismo de selección. La idea principal de la selección es preferir las mejores soluciones sobre las peores, con este fin se han propuesto diferentes procedimientos de selección como métodos proporcionales a la aptitud como la selección por ruleta, y métodos ordinales como la selección por torneo y la selección por truncado.

- *Recombinación:*

El operador de cruzamiento o recombinación combina partes de dos o más candidatos para crear un candidato nuevo, y posiblemente mejor, es decir un vástago. Existen diversos mecanismos para este fin pero la idea principal es que el vástago no sea idéntico a ninguno de sus dos padres sino que combine las características de ellos y origine una característica propia.

- *Mutación:*

El operador de mutación produce variaciones aleatorias en una solución posible. Al igual que la recombinación existen diversos mecanismos de mutación, pero en general se busca uno o más cambios en las características de un individuo. Esto agrega cierto grado de aleatoriedad a la búsqueda y permite explorar porciones del espacio solución en la vecindad del candidato.

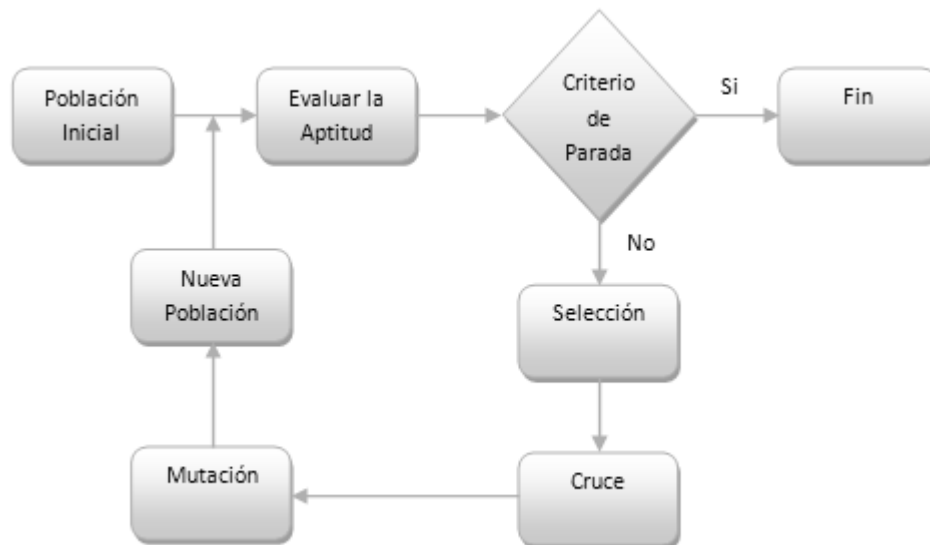
- *Reemplazo:*

La generación nueva creada a partir de los procesos de selección, recombinación y mutación, reemplaza a la población original de padres. Existen diversas técnicas como el elitismo y el reemplazo estacionario comúnmente utilizadas en los GAs.

- *N-Iteración:*

El último paso consiste en repetir los pasos 2 al 6 hasta que el criterio de parada sea alcanzado, como se observa en la figura 39.

**Figura 39.** Representación esquemática de un AG típico, tomada de (Sastry, 2007)



La aptitud asigna un valor relativo a cada candidato solución en la población. Esta medida puede ser la misma función objetivo, puede ser la función objetivo afectada por una función de penalización o puede ser una función subjetiva donde el diseñador selecciona la mejor opción en lugar de la peor, y en general la aptitud o el nivel de adaptación de una solución puede ser cuantificada por uno o por varios criterios, como es el caso de la optimización multi-objetivo.

### 3.2. ALGORITMOS DE OPTIMIZACIÓN MULTI-OBJETIVO (MOO)

En los problemas reales a menudo contienen múltiples funciones objetivo que compiten entre sí, luego existen una serie de técnicas que pueden manejar múltiples funciones objetivo, como por ejemplo minimizar el costo de fabricación y a la vez maximizar la confiabilidad del producto, y que permiten formular el problema como una optimización multiobjetivo.

La expresión canónica para un problema de optimización es:

$$\max Z = f(x)$$

$$g_i(x) \geq 0, \quad i = 1, 2, \dots, m$$

Donde, se define  $Z$  como función del vector  $x$ , sujeto al conjunto de restricciones  $g_i(x)$ .

Sin perder mucha generalidad se puede escribir un problema de optimización multiobjetivo como:

$$\max\{z_1 = f_1(x), z_2 = f_2(x), \dots, z_q = f_q(x)\}$$

$$g_i(x) \geq 0, \quad i = 1, 2, \dots, m$$

Donde, se define “ $z_1 \dots z_q$ ” como funciones del vector  $x$ , y que se encuentran sujetas al conjunto de restricciones  $g_i(x)$ .

En ingeniería mecánica a nivel de proyectos de pregrado el enfoque empleado en trabajos previos relacionados con la utilización de algoritmos genéticos aplicados al diseño mecánico (Velez Langs, 1999) (Arenas Jimenez & Serrano Arango, 2000) (Rodriguez Cañon, 2004) ellos emplean lo que se conoce como ponderación de objetivos para facilitar la obtención de una respuesta o un juego de parámetros óptimos. El funcionamiento de la ponderación de objetivos se observa a continuación:

$$z(x) = \chi_1 f_1(x) + \chi_2 f_2(x) + \dots + \chi_q f_q(x)$$

$$\sum_{i=1}^{i=q} \chi_i = 1$$

La solución sin embargo, dependerá de la importancia relativa que se le dé a las funciones objetivo, comprometiendo la calidad del diseño con un criterio muy subjetivo.

Para compensar esto se hacen varias corridas con diferentes factores de ponderación para generar un ‘conjunto’ de soluciones que sirva de referencia al diseñador, haciendo evidente una de las características ejemplares de los problemas multi-objetivo, no existe una solución única que sea optima con respecto a todos los criterio empleados, debido a que éstos a menudo se encuentran en conflicto. Una solución puede ser la mejor para uno de los criterios y la peor para los demás, por tanto, existe un conjunto de soluciones que sencillamente no pueden ser comparadas entre sí, este conjunto es conocido como frente de Pareto o conjunto de soluciones no-dominadas.

Un ejemplo típico puede ayudar a aclarar el concepto de frente de Pareto, se presenta a continuación como se muestra en la Ecuación 53.

**Ecuación 53.** Problema Multi-Objetivo

$$\max \left\{ f_1(x) = 1 - e^{-4x_1} \sin^6(6\pi x_1), f_2(x) = g(x) \left[ 1 - \left( \frac{f_1(x)}{g(x)} \right)^2 \right] \right\}$$

$$g(x) = 1 + 9 \left( \sum_{i=1}^6 \frac{x_i}{4} \right)^{0.25}$$

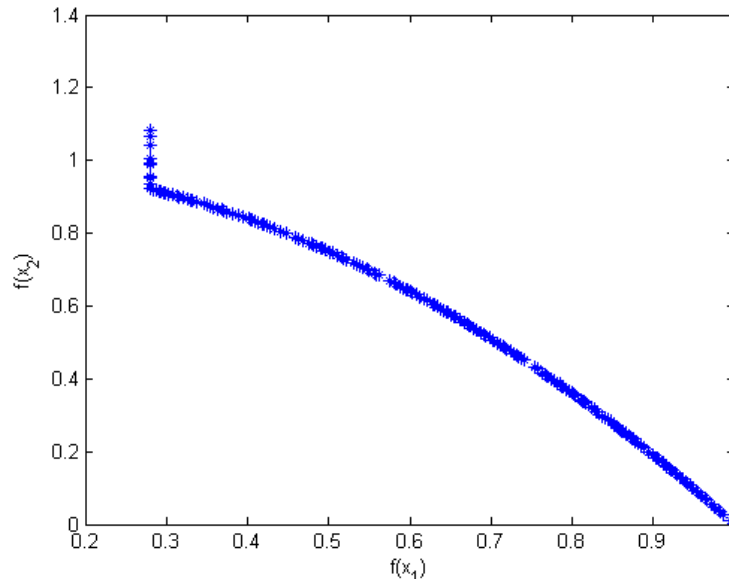
$$0 \leq x_i \leq 1, i = 1, \dots, 6.$$

Donde,  $f_1(x)$  y  $f_2(x)$  son dos funciones que se emplean comúnmente para la calibración de estos métodos de búsqueda y de la cual existe una solución exacta para efectos de comparación.

Para este problema con 6 variables se obtiene una aproximación al frente óptimo como muestra en la figura 40. Cada uno de los puntos observados corresponde a cada uno de los individuos de la población final graficados de forma  $f_1(x)$  versus  $f_2(x)$ .

Obsérvese la forma cóncava del frente de Pareto, que es característico cuando se evalúan funciones objetivo que compiten entre sí.

**Figura 40.** Frente de Pareto para el ejemplo 1



El proceso de toma de decisiones puede obtenerse de cualquiera de las dos formas, bien sea una solución óptima sujeta a cierto criterio, o la identificación de todas las soluciones no-dominadas o su aproximación para otorgar una visión global del problema. Esta última forma es más deseable en el caso de que no se posean conocimientos previos o alguna preferencia específica en cuanto a los objetivos, pero resulta en un mayor costo computacional.

Con base en el informe presentado por (Kundle, 2005) que hace una comparación en términos de desempeño, exactitud, sensibilidad a la presencia de ruido en el espacio solución y diversidad del frente óptimo, se presentan los algoritmos de optimización multi-objetivo (MOOGA) considerados más adecuados para la solución del problema en cuestión. Con este criterio reducimos la lista de alternativas de solución a tres opciones:

- NSGA-II (2000) – Non-dominated Sorting Genetic Algorithm
- SPEA2+ (2004) – Strength Pareto Evolutionary Algorithm
- Micro-GA2 (2003) – Micro-Genetic Algorithm

El Micro-GA2 es una alternativa promisorio ya que esta específicamente diseñado para trabajar en poblaciones pequeñas para reducir el número de evaluaciones, pero también resulta ser el más complejo de los tres por incluir parámetros auto-adaptativos y los detalles específicos de su funcionamiento de momento no están documentados.

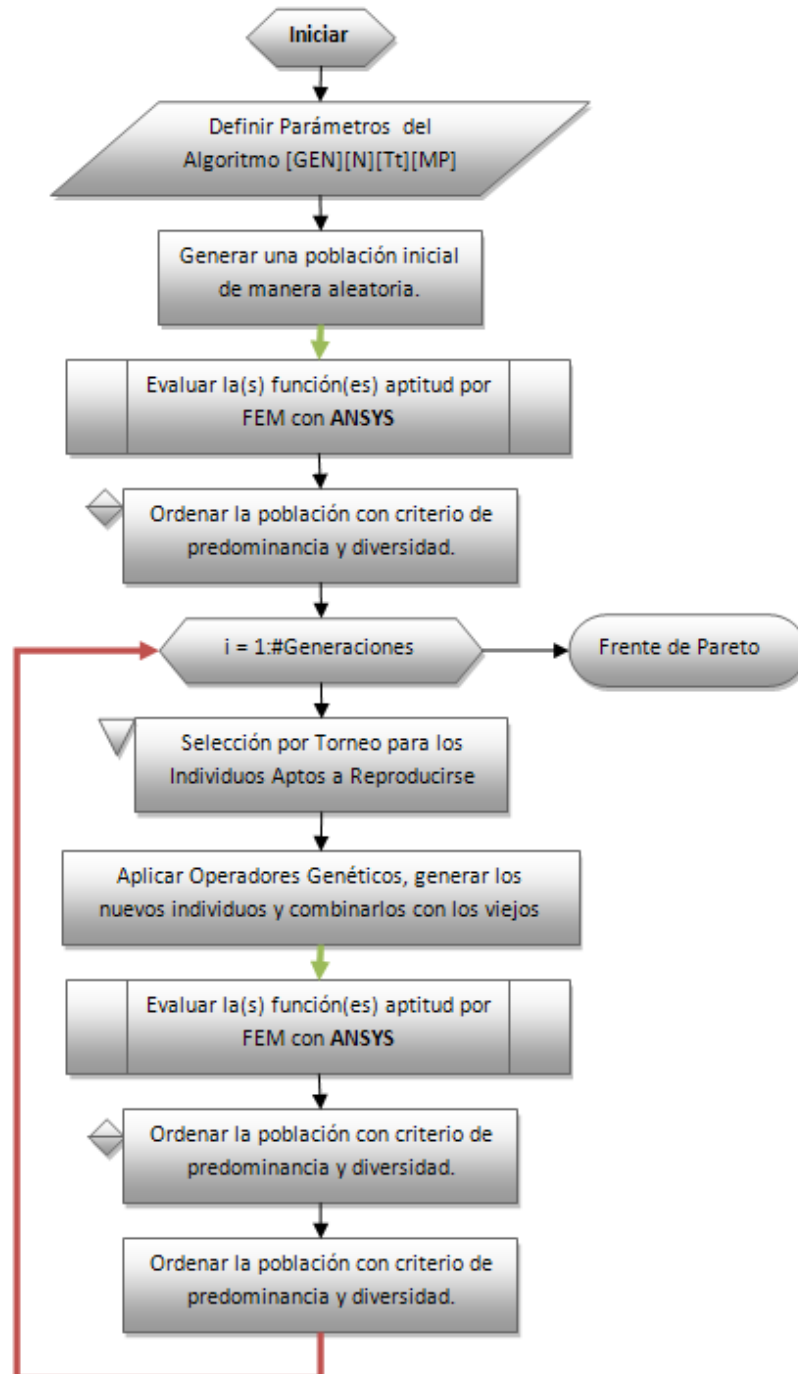
En cuanto al SPEA2+ se observa que en algunos problemas se obtienen muy buenos resultados en algunos casos en comparación del NSGA-II, y en otros no tanto, por lo que no quedan muy claras las ventajas comparativas de su utilización.

Para efectos del presente trabajo de grado se opta por emplear el algoritmo NSGA-II propuesto inicialmente por (Deb, Pratap, Agarwal, & Meyarivan, 2002) al considerarse como un método muy robusto, competitivo, actualmente es uno de los métodos de optimización más populares que hay y por consiguiente existe suficiente documentación disponible para una fácil implementación del mismo.

### 3.3. NON-DOMINATED SORTING GENETIC ALGORITHM – NSGA-II

La representación esquemática del algoritmo NSGA-II se presenta en la figura 41.

Figura 41. Esquema de Funcionamiento Propuesto para el Algoritmo NSGA-II



El funcionamiento del algoritmo difiere del planteamiento genérico, en la medida que la selección de los individuos no se efectúa con base al valor de la función aptitud, sino en función del rango asignado a cada individuo de acuerdo al criterio de predominancia y diversidad.

En la figura 41 la línea verde destaca el intercambio de datos entre el algoritmo y nuestro modelador FEM como un proceso externo y con una línea roja el proceso iterativo a lo largo de  $N$  generaciones como parte de la evolución de la respuesta.

La selección de los parámetros que determinan el desempeño del GEA<sup>21</sup>, afectara en buena medida la aproximación de la solución final al valor óptimo real, la diversidad de las soluciones obtenidas y el tiempo de convergencia del algoritmo. Los parámetros que influyen directamente en el desempeño del algoritmo son:

- [GEN]: El numero de generaciones.
- [N]: El tamaño de la población que permanecerá constante a lo largo del proceso.
- [Tt]: El tamaño de la sub-población que entra a la selección por torneo.
- [MP]: *Mating Pool Size* ó la cantidad de Individuos aptos para reproducción.
- Los respectivos operadores genéticos que veremos más adelante.

### 3.3.1. Estructura De Datos

Está claro que para el funcionamiento de un algoritmo genético típico en un espacio de búsqueda  $\Phi_g$  requiere, necesariamente, de la utilización de un sistema de representación, empleando el concepto de genotipos y fenotipos de manera análoga a la empleada en biología. La función objetivo se dividirá entonces en dos partes, la primera convertirá el espacio genotípico  $\Phi_g$  en un espacio fenotípico  $\Phi_f$ , y la segunda pasará del espacio fenotípico al espacio real.

$$f_g(x^g): \Phi_g \rightarrow \Phi_f$$

$$f_f(x^f): \Phi_f \rightarrow \mathbb{R}$$

---

<sup>21</sup> Genetic and Evolutionary Algorithms, GEA

Donde,  $f = f_f (f_g(x^g))$ , y la transformación  $f_g$  viene dada por el tipo de representación empleada. La función  $f_f$  representa la función objetivo y asigna un valor de aptitud  $f_f(x^f)$  a cada individuo  $x^f \in \Phi_f$ . Los operadores genéticos son aplicados a los individuos que se encuentran en  $\Phi_g$ .

Para codificar los parámetros se emplea una combinación entre la representación binaria para lo que son valores discretos, como pueden ser diámetros de taladrados, rodamientos, calibres o partes estandarizadas, y la representación real para los parámetros continuos como son longitudes, ángulos, etc.

En general, podemos expresar dichos parámetros – llamados genes – en una cadena compuesta llamada cromosoma, cada una de estas cadenas representa a un individuo o candidato solución.

$$\bar{x}^p = \{x_1^p, x_2^p, x_3^p, \dots, x_L^p\} \quad i \in \{I_{\mathbb{R}} \cup I_{\mathbb{Z}}\}, p = 1, 2, 3, \dots, N$$

Donde:

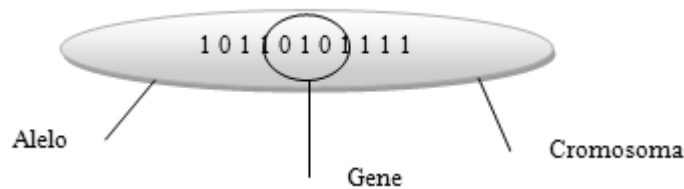
$x_i^p$ : Gen  $i$  del individuo  $p$ .

$L$ : Longitud del cromosoma o el número de parámetros.

$N$ : Tamaño de la población.

Los AG convencionales tradicionalmente emplean una representación binaria de sus parámetros continuos, limitando el número de valores posibles que puede tomar dentro de un rango y codificándolo dentro de una cadena binaria de longitud especificada. Sin embargo, con la aparición de operadores genéticos especialmente diseñados para la utilización con parámetros continuos quedaron evidenciados los problemas asociados a esta metodología.

**Figura 42.** Representación binaria de una característica, tomada de (Sastry, 2007)



Aunque la forma más natural de representar problemas continuos es mediante una representación continua con fenotipos continuos, en trabajos previos se ha demostrado que trabajar con genotipos de una cardinalidad menor a la del alfabeto permite analizar una mayor cantidad de información en comparación con cadenas enteras. Las representaciones binarias, como la mostrada en la figura 42, más comunes para valores reales son la representación binaria, gris (gray) y unaria.

La representación binaria es la representación más comúnmente utilizada para GAs selecto-retrocombinativos, es decir Algoritmos Genéticos que utilizan el operador de recombinación como el principal operador de búsqueda, utilizando este tipo de GAs el operador de mutación reduce su función a ruido de fondo.

El espacio de búsqueda  $\Phi_g$  se denota mediante la expresión  $\Phi_g = \{0,1\}^l$ , donde,  $l$  es la longitud del vector binario que representa al candidato solución de forma  $x^f = (x_1^f, x_2^f, x_3^f \dots x_l^f)$ . El tamaño del espacio de búsqueda se denota como  $|\Phi_g| = 2^l$ .

Cuando se codifican variables continuas mediante una representación binaria la precisión de la representación dependerá de la longitud de la cadena o del número de bits que representen la variable. Cuando codificamos una variable continua con un vector de longitud  $l$  podemos aspirar a una precisión máxima de  $\frac{1}{2^{l+1}}$ .

Por ejemplo, los parámetros característicos de una barra circular son su longitud, su diámetro y su material. Para este caso podríamos emplear una representación compuesta con dos parámetros reales para las dimensiones y uno binario para la selección del material.

$$\bar{x}^p = \{d_{real}^p \ L_{real}^p \ M_{bin}^p \}$$

Donde,  $d_{real}^p$  corresponde al valor del diámetro,  $L_{real}^p$  a la longitud de la barra y  $M_{bin}^p$  el material en cuestión para el individuo  $p$ . Si existen 4 posibilidades de material, podemos emplear una representación binaria a 2-bits, luego  $M_{bin}^p$  tomaría los valores 00, 01, 10 y 11.

Considérese pues una cadena binaria de longitud  $l$  en un sistema de un solo parámetro y que se encuentra siguiendo una tendencia.

$$x_n = 0111111111111111$$

El siguiente valor en la espacio de búsqueda real es 1000000000000000, sin embargo observamos que la representación binaria no está relacionada con el numero anterior, de modo que tomara al sistema varias generaciones llegar al valor siguiente y continuar el proceso de explotación.

$$\begin{aligned} \{x_1\}_g &= 0111111111111111 & \{x_1\}_f &= 16383 \\ \{x_1\}_f &= 1000000000000000 & \{x_1\}_f &= 16384 \end{aligned}$$

Esto sucede porque los espacios genotípicos y fenotípicos de búsqueda no son el mismo, luego la tendencia de uno no va a poder ser seguida con la misma facilidad que en el otro.

### 3.3.2. Operadores de Cruce ó Recombinación

Es preciso emplear dos tipos de operador de cruce, uno para la representación binaria y otro un poco más complejo para la representación real.

Para los parámetros continuos se emplea la *Recombinación Binaria Simulada* (ó SBX<sup>22</sup>, por sus siglas en inglés) que intenta emular el efecto del cruce de un punto empleado en la representación binaria.

---

<sup>22</sup> Simulated Binary Crossing, SBX

Se requiere generar un número aleatorio  $u$  cuyo valor oscila entre 0 y 1, y se procede a calcular el parámetro  $\bar{\beta}(u)$  tal que:

$$\bar{\beta}(u) = \begin{cases} (2u)^{\frac{1}{\eta_c+1}} & \text{si } u \leq 0.5 \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta_c+1}} & \text{si } u > 0.5 \end{cases}$$

Donde  $\eta_c$  se conoce como el índice de distribución de la recombinación, y los autores de esta técnica recomiendan que este valor varíe entre 1 y 2.

Considere los padres:

$$\{(x_1^1 \ x_2^1 \ x_3^1 \ \dots \ x_L^1)_{real} \ (x_{L+1}^1 \ x_{L+2}^1 \ x_{L+2}^1 \ \dots \ x_M^1)_{bin}\}$$

$$\{(x_1^2 \ x_2^2 \ x_3^2 \ \dots \ x_L^2)_{real} \ (x_{L+1}^2 \ x_{L+2}^2 \ x_{L+2}^2 \ \dots \ x_M^2)_{bin}\}$$

Tomando los padres  $x^1$  y  $x^2$  las cadenas continuas  $(x_1^1 \ x_2^1 \ x_3^1 \ \dots \ x_L^1)$  y  $(x_1^2 \ x_2^2 \ x_3^2 \ \dots \ x_L^2)$  producirán dos individuos nuevos que podemos llamar  $x^3$  y  $x^4$  de acuerdo a lo siguiente.

$$x^3 = (x_1^3 \ x_2^3 \ x_3^3 \ \dots \ x_L^3)$$

$$x^4 = (x_1^4 \ x_2^4 \ x_3^4 \ \dots \ x_L^4)$$

Teniendo a  $k = 1, 2, 3, \dots, L$ , los genes nuevos serán:

$$x_k^3 = \frac{1}{2}[(1 - \bar{\beta}_k)x_k^1 + (1 + \bar{\beta}_k)x_k^2]$$

$$x_k^4 = \frac{1}{2}[(1 + \bar{\beta}_k)x_k^1 + (1 - \bar{\beta}_k)x_k^2]$$

Es decir que por cada posición del primer hijo elegimos al padre cuyo  $k$ -ésimo será transmitido a este descendiente, y la posición respectiva del segundo hijo se completa con el valor de gene correspondiente.

La representación binaria por su parte emplea un operador de cruce de dos puntos (2PX\_V1) que se aplica a la población padre con una probabilidad dada - en los problemas de ingeniería  $p_c$  oscila entre 60 y 100% -. Este operador extrae los genomas

binarios del cromosoma de dos candidatos solución y los cortan en dos puntos escogidos de manera aleatoria, produciendo 3 segmentos por individuo que serán intercambiados entre sí para producir dos cadenas que formaran parte de dos individuos nuevos. De esta manera, los hijos creados heredan algunas de las características de sus padres.

Tomando los padres  $x^1$  y  $x^2$  del ejemplo anterior que poseen las cadenas binarias  $(x_1^1 x_2^1 x_3^1 \dots x_M^1)$  y  $(x_1^2 x_2^2 x_3^2 \dots x_M^2)$ . Se selecciona de manera aleatoria un punto de corte  $c_1$  que oscile entre  $1 \leq c_1 \leq M$ , y una longitud del cruce  $c_2$  que varíe entre  $c_1 \leq c_2 \leq M - 1$ . Donde,  $M$  es la longitud de la cadena binaria.

$$\text{Padres: } \{x_1^1 x_2^1 | x_3^1 \dots x_{M-1}^1 | x_M^1\} \{x_1^2 x_2^2 | x_3^2 \dots x_{M-1}^2 | x_M^2\}$$

$$\text{Hijos: } \{x_1^1 x_2^2 | x_3^2 \dots x_{M-1}^2 | x_M^1\} \{x_1^2 x_2^1 | x_3^1 \dots x_{M-1}^1 | x_M^2\}$$

Tomemos por ejemplo la recombinación tomando las siguientes cadenas como padres:

$$x_1 = \{(7 \quad 2 \quad 8)_{real} (110100)_{bin}\}$$

$$x_2 = \{(5 \quad 9 \quad 15)_{real} (101001)_{bin}\}$$

Con un índice de distribución de recombinación  $\eta_c$  de 1, seleccionamos un valor de  $u$  de manera aleatoria para cada uno de los parámetros reales y  $c_1$  para los parámetros binarios. Para simplificar mostraremos los resultados en la Tabla 4 a continuación.

**Tabla 4.** Ejemplo del Operador de Recombinación

Parámetro	$u_{real}$	$\bar{\beta}(u)$	$x_k^3$	$x_k^4$
1	0.8147	1.6428	4.3575	7.6428
2	0.9058	2.3038	11.5632	2.5632
3	0.1270	0.7568	14.1488	8.8512

$c_1$	$c_2$	$x_k^3$	$x_k^4$
3	5	110100	101101

Se producen las cadenas 3 y 4, que se observan a continuación, y que comparten algunas propiedades con las cadenas originales.

$$x_3 = \{(4.35 \ 11.56 \ 14.14)_{real} \ (110100)_{bin}\}$$

$$x_4 = \{(7.64 \ 2.56 \ 8.85)_{real} \ (101101)_{bin}\}$$

### 3.3.3. Operador De Mutación

Los algoritmos genéticos empleados en problemas de ingeniería se caracterizan por reducir el efecto del operador de mutación a un nivel marginal, en lo que se denominan algoritmos selectoretrocombinativos.

Al igual que para el operador de cruce, es preciso emplear dos mecanismos diferentes según el sistema de representación. Para la representación binaria el operador de mutación se aplica independientemente a cada individuo de la población padre con cierta probabilidad  $p_m$ , y efectúa un cambio en uno de los genes seleccionado de manera aleatoria. En general se reportan buenos resultados utilizando una tasa de mutación  $p_m = \frac{1}{L}$ , donde  $L$  es la longitud de la cadena binaria.

$$Padre: \{x_{L+1}^1 \ x_{L+2}^1 \ x_{L+2}^1 \ \dots \ x_{M-1}^1 \ x_M^1\}$$

$$Hijo: \{x_{L+1}^1 \ x_{L+2}^1 \ x_{L+2}^1 \ \dots \ y_{M-1}^1 \ x_M^1\}$$

Si bien la utilización de una mutación totalmente aleatoria puede no contribuir mucho a la explotación de un punto con buen desempeño, su finalidad última es la exploración del espacio de búsqueda y mantener la diversidad en la población, luego es ampliamente aceptada como mecanismo evolutivo.

Por ejemplo, para una de las cadenas seleccionadas para el ejemplo anterior  $x_1$ , y para un punto de salto en el tercer bit, tenemos que:

$$x_1 = \{(7 \ 2 \ 8)_{real} \ (110100)_{bin}\}$$

$$x_2 = \{(7 \ 2 \ 8)_{real} \ (11\mathbf{1}100)_{bin}\}$$

La representación continua emplea la mutación polinomial, que es análoga de alguna manera al SBX.

$$x_k^m = x_k^1 + (\Delta x_k) \delta_k$$

Donde  $\Delta x_k$  es el rango de  $x_k$ , y  $\delta_k$  es una variación pequeña calculada a partir de una distribución polinomial, empleando la siguiente fórmula:

$$\delta_k(r_k) = \begin{cases} (2r_k)^{\frac{1}{\eta_m+1}} - 1, & \text{si } r_k < 0.5 \\ 1 - [2(1 - r_k)]^{\frac{1}{\eta_m+1}}, & \text{si } r_k \geq 0.5 \end{cases}$$

Donde  $r_k$  es un número aleatorio entre 0 y 1, y  $\eta_m$  corresponde al índice de distribución de la mutación que toma valores entre 1 y 2, y digamos que los parámetros se encuentran en un rango de 1 a 15.

Para la misma cadena del ejemplo anterior, tenemos que:

**Tabla 5.** Ejemplo del operador de mutación

Parámetro	$r_k$	$\delta_k(r_k)$	$x_k^2$
1	0.9649	0.7350	18.0251
2	0.1576	-0.2980	-2.4691
3	0.9706	0.7575	19.3622

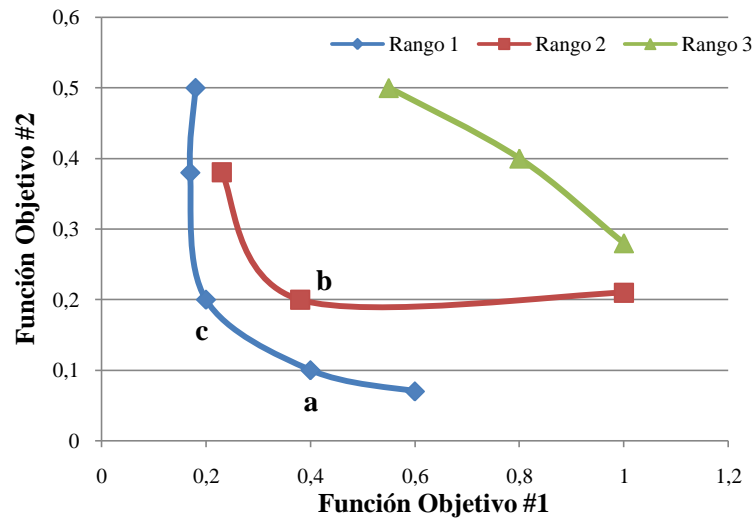
La mutación se aplica a uno de los parámetros escogido al azar.

$$x_2 = \{(7 \quad 2 \quad 19.3622)_{real} (111100)_{bin}\}$$

### 3.3.4. Mecanismo De Selección

Como alternativa al método de funciones agregadas u objetivos ponderados empleados en proyector anteriores, se planteo la utilización de un mecanismo de selección por torneo binario empleando el ranking de Pareto y la distancia de separación como criterios de selección en lugar de las funciones objetivo, para evitar que la población converja al óptimo de una función objetivo y obtener un conjunto de soluciones con una buena distribución.

Figura 43. Ordenamiento por Predominancia



La población se categoriza de acuerdo a un ranking de dominación que es asignado a los individuos de la población basados en los valores de sus múltiples objetivos, como se observa en la figura 43.

Para el caso particular de un problema con dos objetivos que compiten entre sí, es posible utilizar el concepto de predominancia. Por ejemplo, la solución “a” predomina sobre la solución “b” porque su comportamiento es mejor para ambas funciones objetivo, en general se encuentra mas cerca de la esquina inferior izquierda, sin embargo es indiferente o no-dominante para con la solución “c” debido a que ambas dominan una de las funciones objetivo.

Comenzando por el juego de soluciones que no son dominadas por ninguna otra solución en la población son agrupadas y se les es asignado un ranking de 1. Por ejemplo en la figura 43, tanto “a” como “c” se consideran soluciones no-dominadas, se les es asignado el ranking de 1 y pasan a formar parte del grupo 1. Luego se agrupan las demás soluciones que sean no-dominadas por ninguna de las soluciones restante, es decir que solo sean dominadas por candidatos solución del grupo 1, y se les asigna un ranking de 2.

El proceso de selección continúa de esa manera de modo que quedamos con diferentes subpoblaciones con diferentes rankings. Las soluciones que tenga un ranking dado serán consideradas como no-dominadas por las soluciones que pertenezcan a ese mismo grupo y se considerara como dominadas por soluciones con un ranking menor.

Por su parte la distancia de separación<sup>23</sup> provee un estimado de la densidad de soluciones que se encuentran en la vecindad de una solución. Esta distancia se obtiene ordenando la población de candidatos en orden ascendente del valor de su función objetivo, y obtenemos el valor de la distancia de separación de un individuo como el promedio de la distancia a sus dos vecinos más próximos.

La selección por torneo por lo general se efectúa entre dos individuos, conocido como torneo binario, pero el tamaño de la población puede variarse para ajustar la presión o intensidad de la selección. Este método garantiza de cierta manera la diversidad de la población, pero en algunos casos puede resultar en pérdida de material genético al encontrarse subpoblaciones de individuos muy bien adaptados que se descarten mutuamente en las etapas iniciales del proceso.

### **3.3.5. Función de Penalización**

Los algoritmos genéticos y por extensión los algoritmos genéticos multi-objetivo, son en esencia técnicas de optimización no-restringidas, por lo que resulta de especial importancia la manera como se manejan las restricciones del diseño. La forma más comúnmente empleada de trabajar con las restricciones es la utilización de penalizadores como se muestra en la Ecuación 54.

**Ecuación 54.**

$$adapatacion_i = f_i(\hat{x}) + Q_i(\hat{r})$$

---

<sup>23</sup> La distancia de separación es la medida de separación entre la solución en cuestión

Donde  $adaptacion_i$  corresponde al valor de la función aptitud del i-esimo individuo de la población,  $f_i(\hat{x})$  es el valor de la función a ser minimizada y  $Q_i$  corresponde al valor del penalizador por violar las restricciones del problema.

La violación de algunas restricciones en algunos casos es más grave que en otros y esto naturalmente debería verse reflejado en el valor de la función de penalización para permitir al sistema distinguir entre una solución no factible y una imposibilidad. Luego para cada uno de los elementos del vector de restricciones de restricciones y en función de que tan lejana se encuentra la solución del espacio de búsqueda permisible, se asignaran diferentes valores a la función objetivo, como muestra la Ecuación 55.

**Ecuación 55.** Función de Penalización

$$Q_i(\hat{r}) = Q_{i1} + Q_{i2} + \dots + Q_{ip}$$

$$Q_j = \begin{cases} r_j |f_i(\hat{x})| & \text{si } r_j \neq 0 \\ 0 & \text{si } r_j = 0 \end{cases}$$

Por ejemplo, para el requerimiento de potencia tenemos que:

$$r_1 = \begin{cases} \frac{(\dot{W}_{min} - \dot{W})}{\dot{W}_{min}} & \text{si } \dot{W} < 1.5 \text{ HP} \\ \frac{(\dot{W} - \dot{W}_{max})}{\dot{W}_{max}} & \text{si } \dot{W} > 2 \text{ HP} \\ 0 & \text{si } \dot{W} \in (1.5, 2) \end{cases}$$

Si la potencia obtenida es digamos de 1.25HP, estamos violando el conjunto de restricciones y se debe proceder a calcular el valor del penalizador, como se muestra:

$$r_1 = \frac{(\dot{W}_{min} - \dot{W})}{\dot{W}_{min}} = \frac{1.5 - 1.25}{1.5} = 0.1667$$

Luego, el valor de la función objetivo de esta solución en cuestión deberá multiplicarse por  $(1 + r_1)$ , para eliminar gradualmente a las soluciones no-factibles sin eliminarlas inmediatamente y reducir la diversidad genética de la población.

### 3.4. DOMINIO DE LA FUNCIÓN APTITUD

Después de un análisis comprensivo de la metodología de diseño propuesta y de los modelos presentados en el capítulo anterior, y considerando el alto costo computacional de la evaluación de las funciones objetivo se consideró conveniente limitar los parámetros de diseño a un total de 7 variables continuas y 3 parámetros discretos, para obtener un vector de variables de diseño ( $\hat{x}$ ) de longitud 10. Empleando estas variables con conjunto con los modelos matemáticos planteados en el capítulo 2 y con los modelos complementarios descritos en el apéndice D, podemos definir las características geométricas y físicas fundamentales del conjunto Estator-Rotor.

#### 3.4.1. Dominio de las Variables Continuas

El dominio de las variables continuas se encuentra definido en la función *valeurs.m*<sup>24</sup>, que se encuentra empaquetada en la carpeta +GA. El algoritmo en si fue programado de manera tal que se puedan añadir y quitar parámetros según se crea conveniente de manera que se pueda adaptar fácilmente a otro tipo de problemas. Los valores contemplados para las variables continuas son los siguientes:

$$\hat{x} = (x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7)_{real}$$

1. Radio Generador ( $R$ ):  $x_1 \in (20 \times 10^{-3}, 25 \times 10^{-3}) |m|$
2. Relación de Compresión Deseada  $\varepsilon_r$ :  $x_2 \in (7,11) |1/1|$
3. Separación de los Sellos ( $a$ ):  $x_3 \in (0.2 \times 10^{-3}, 0.6 \times 10^{-3}) |m|$
4. Ángulo de Encendido ( $\theta_s$ ):  $x_4 \in (-55, -15) |^\circ TDC|$
5. Cierre Lumbrera de Admisión ( $\theta_i$ ):  $x_5 \in (-210, -190) |^\circ TDC|$
6. Apertura Lumbrera de Escape ( $\theta_o$ ):  $x_6 \in (190,210) |^\circ TDC|$
7. Numero de Aletas Radiales ( $n_a$ ):  $x_7 \in (30,46) |1/1|$

---

<sup>24</sup> Ver apéndice 1

Los valores de las variables que por efectos de los operadores genéticos lleguen a encontrarse fuera de estos valores, serán necesariamente considerados como un valor no factible y por tanto será aplicado un penalizador a valor de la adaptación, de modo que el rango de la solución se vea afectado y se pueda eliminar gradualmente estos individuos de la población.

### **3.4.2. Dominio de las Variables Discretas**

Una variable discreta es aquella que no puede tomar todos los valores dentro de los límites de la variable, sino que sólo puede tomar ciertos valores dentro de la escala. Un ejemplo sencillo de una variable discreta sería el número de tornillos que puede tener una junta cualquiera, donde sencillamente no tiene sentido hablar de fracciones de tornillos, necesariamente necesitamos saltos de uno en uno, en algunos casos de cuatro en cuatro.

Para el diseño se identificaron 3 variables que fueron tratadas como variables discretas. Como vimos en la sección 2.3.5 existe una relación directa entre la excentricidad del rotor y el diámetro primitivo del engranaje interno, y de la relación del módulo del engranaje y el número de dientes. Este número de dientes del engranaje, como quedo evidenciado en el ejemplo anterior, debe ser un número entero y si deseamos emplear un módulo estandarizado – que es necesario si se considera la fabricación en un futuro – se reducen los valores que puede tomar la excentricidad de manera considerable.

El ancho de la carcasa se planteo inicialmente para que los valores posibles coincidiesen con los espesores de láminas comercialmente disponibles. Sin embargo, para la escala del motor los saltos entre los diferentes calibres de las láminas resultan muy altos, por lo que en la mayoría de los casos se obtienen resultados insatisfactorios que favorecerían a un solo espesor de lámina. La recomendación de (Yamamoto, 1981) para el ancho de la carcasa sugiere un valor de 5 veces la excentricidad del rotor, por lo que se decidió tomar un numero de valores por arriba y por debajo de este valor para esta variable, de esta forma garantizamos no alejarnos demasiado de la recomendación de diseño. Por último, la selección del material es controlada con un parámetro discreto.

$$\hat{x} = (x_8 \quad x_9 \quad x_{10})_{bin}$$

Los últimos tres parámetros se muestran a continuación:

8. Excentricidad del Rotor ( $e$ ):  $x_8 \in (2.40 \quad 2.70 \quad 3.00 \dots 4.50) |mm|$   
 9. Ancho de la Carcasa ( $b$ ):  $x_9 \in (3.50 \quad 3.71 \quad 3.92 \dots 6.50)e |1/1|$   
 10. Selección del Material:  $x_{10} \in (1060 \quad 2017 \quad 2218 \quad 2618)$

El número de posibilidades contempladas nos sugiere una codificación binaria de 4, 4 y 2 bits respectivamente, para obtener una cadena en representación binaria de longitud 10.

### 3.4.3. Conjunto de Restricciones

El conjunto de restricciones se encarga de limitar el espacio de búsqueda y en algunos casos muy específicos puede servir para mejorar el desempeño del AG, en general no hay ninguna garantía de que los sistemas restringidos vayan a facilitar el proceso evolutivo. No obstante, las restricciones son necesarias para asegurar la calidad de la respuesta y reducir al mínimo el número de soluciones no factibles.

El conjunto de restricciones se muestra a continuación:

$$(\hat{r}) = [r_1 \quad r_2 \quad r_3 \quad r_4 \quad r_5 \quad r_6 \quad r_7 \quad r_8]$$

1. Requerimiento de Potencia:  $r_1 = \begin{cases} \frac{(\dot{W}_{min} - \dot{W})}{\dot{W}_{min}} & \text{si } \dot{W} < 1.5 \text{ HP} \\ \frac{(\dot{W} - \dot{W}_{max})}{\dot{W}_{max}} & \text{si } \dot{W} > 2 \text{ HP} \\ 0 & \text{si } \dot{W} \in (1.5, 2) \end{cases}$
2. Factor de Seguridad Mínimo:  $r_2 = \begin{cases} \frac{(SF_{min} - SF)}{SF_{min}} & \text{si } SF < 1.2 \\ 0 & \text{si } SF > 1.2 \end{cases}$
3. Constante de la Epitrocoide:  $r_3 = \begin{cases} \frac{(K_{min} - K)}{K_{min}} & \text{si } K < 7 \\ \frac{(K - K_{max})}{K_{max}} & \text{si } K > 13 \\ 0 & \text{si } K \in (7, 13) \end{cases}$

4. Espesor mínimo del Rotor:  $r_4 = \begin{cases} \frac{(2e-A_m)}{2e} & \text{si } A_m < 2e \\ 0 & \text{si } A_m \geq 2e \end{cases}$
5. Bolsillo del Rotor:  $r_5 = \begin{cases} \frac{(V_{ad}-A_r b)}{A_r b} & \text{si } V_{ad} > A_r b \\ 0 & \text{si } V_{ad} \leq A_r b \end{cases}$
6. Relación de Compresión:  $r_6 = \begin{cases} \frac{(\varepsilon_{real} - \varepsilon_{theory})}{\varepsilon_{theory}} & \text{si } \varepsilon_{real} > \varepsilon_{theory} \\ 0 & \text{si } \varepsilon_{real} \leq \varepsilon_{theory} \end{cases}$
7. Diámetro mínimo del Rodamiento:  $r_7 = \begin{cases} \frac{(E_w + 2e - F_w)}{E_w + 2e} & \text{si } F_w < E_w + 2e \\ 0 & \text{si } F_w \geq E_w + 2e \end{cases}$

#### 4. ANÁLISIS DE RESULTADOS

El algoritmo fue ejecutado una cantidad determinada de veces con la función objetivo en cuestión y con diferentes ejercicios de prueba como el que se muestra en la Ecuación 53 (Ver. Pag.77), con diferentes tamaños de población y número de generaciones para evaluar su comportamiento general y hacer una estimación preliminar del tiempo de cálculo total.

La función objetivo arroja como resultado una matriz de  $N \times 2$ , donde  $N$  es el tamaño de la población, y donde esos valores corresponden al nivel de adaptación asignado para cada uno de los miembros de la población para ambas funciones objetivos. Además, obtenemos un vector que almacena los valores de las funciones objetivo sin penalizadores (Eficiencia Térmica y Peso), un identificador que nos indica si una solución es factible o no-factible y el rango o frente al cual pertenecen cada uno de los individuos de la población.

Con esta información hacemos una grafica de  $f_1(\bar{x}) + FP(\bar{r})$  versus  $f_2(\bar{x}) + FP(\bar{r})$ , como se muestra en la figura 44. Las soluciones no-factibles se muestran en color gris y las soluciones factibles se indican en negro, mientras que los frentes compuestos con las mejores soluciones son resaltados como se muestra.

En las primeras iteraciones del algoritmo no se encontraron soluciones factibles en la población entera, luego necesariamente se tuvo que aumentar el tamaño a 150 individuos para aumentar la posibilidad de obtener soluciones factibles. A su vez evidencia que el espacio de búsqueda es particularmente ruidoso, luego se modifican la probabilidad de cruce y mutación para facilitar al algoritmo la exploración del espacio de búsqueda por soluciones factibles.

Los datos mostrados son resultados para la ejecución del algoritmo genético con los siguientes parámetros:

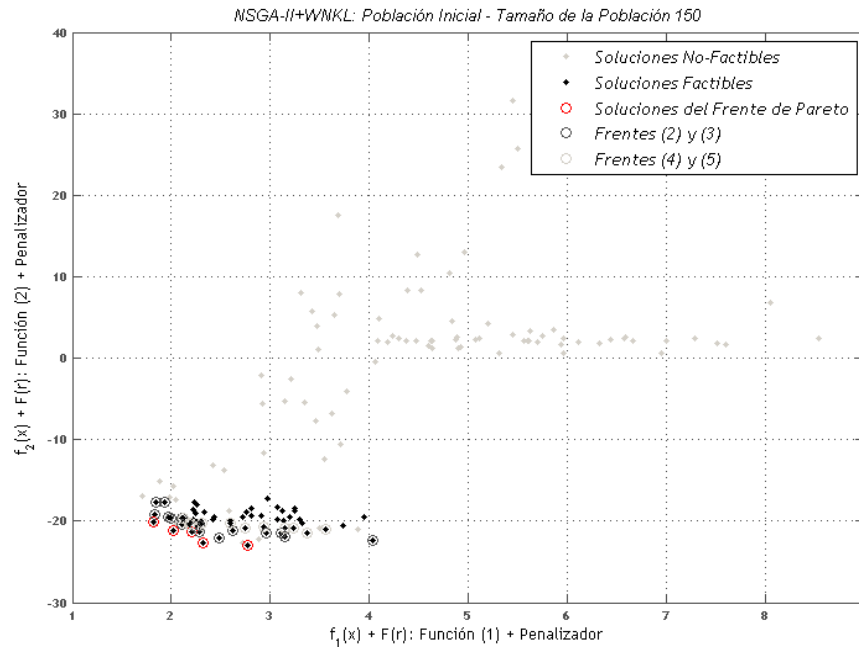
- Tamaño de la Población: 150 Individuos
- Numero de Generaciones: 30 Generaciones
- Probabilidad de Cruce: 50%
- Probabilidad de Mutación: 50%
- Tamaño del Torneo: 2 (Torneo Binario)
- Índice de distribución de la Recombinación: 2
- Índice de distribución de la Mutación: 2

Para aumentar la probabilidad de convergencia del algoritmo, se decidió repetir la población inicial hasta que las soluciones factibles fueran medianamente representativas en la población.

La figura 44 se muestra la población inicial para la corrida principal. Se observa que la población se encuentra ampliamente dominada por soluciones no-factibles que se encuentran desplazadas hacia el cuadrante superior izquierdo de la grafica, mientras que las soluciones factibles – que para este caso representan alrededor del 40% de la población – se encuentran concentradas en la esquina inferior derecha, indicando la preferencia del algoritmo hacia estas.

También se puede observar que el primer frente, que se encuentra resaltado en rojo, consta de 5 individuos y representa marginalmente el 3% de la población total.

**Figura 44.** Población Inicial – Tamaño de la Población 150 –  $f_1(x) + FP(r)$  vs.  $f_2(x) + FP(r)$

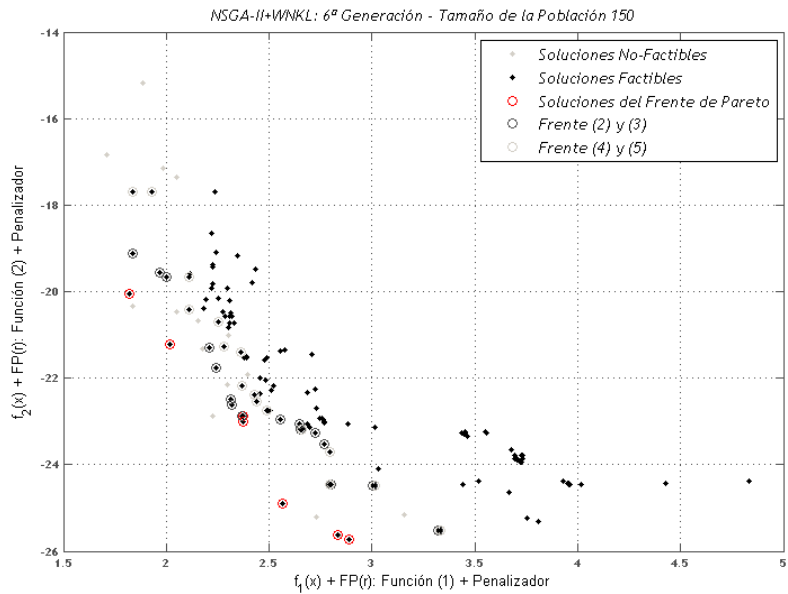


Se espera que gradualmente sean purgadas de la población las soluciones no-factibles durante la etapa de exploración del espacio de búsqueda, hasta que gradualmente solo queden uno o dos frentes, es partir de ese punto empieza lo que se conoce como proceso de explotación del espacio de búsqueda.

Se toman pantallazos cada 6 generaciones para observar el avance de la población y verificar que el comportamiento del algoritmo sea el adecuado. Emplearemos la aptitud media de la población como un parámetro sin unidades para seguir el avance de la población.

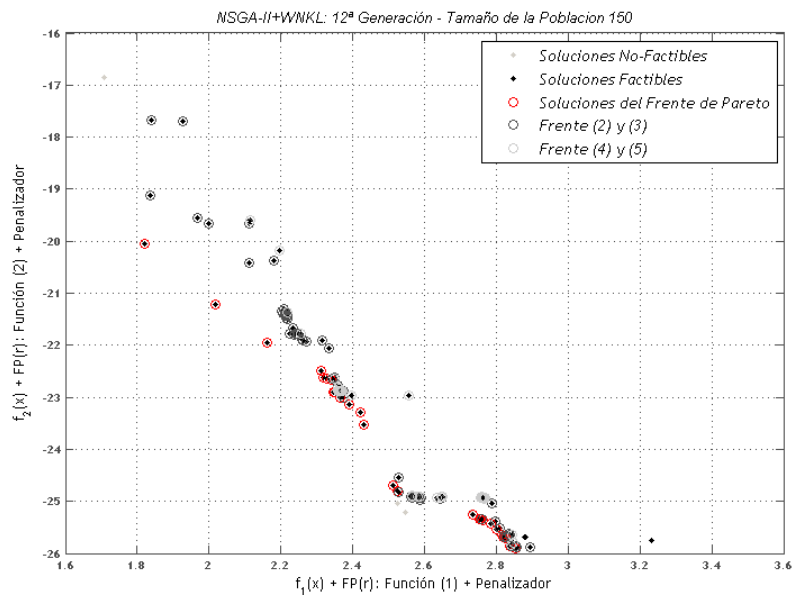
Podemos observar en la figura 45, para la 6ª generación como el número de individuos no-factibles ha disminuido considerablemente y se ha reducido a alrededor del 12%. La aptitud media de la población es de  $[2.75, -22]$ .

**Figura 45.** 6ª Generación – Tamaño de la Población 150 –  $f_1(x) + FP(r)$  vs.  $f_2(x) + FP(r)$



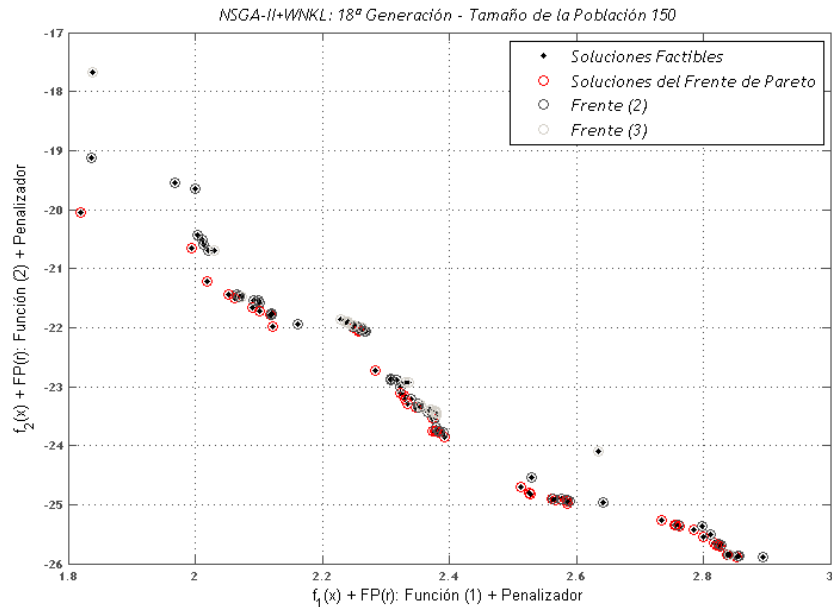
Se observa en la Figura 46 la población para la 12ª generación, como las soluciones no-factibles están a punto de ser purgadas de la población y representan difícilmente un 4% del tamaño total de la población. Las soluciones factibles se concentran en los frentes 1 al 5 y la aptitud media de la población se encuentran en  $[2.47, -23.47]$ .

**Figura 46.** 12ª Generación – Tamaño de la Población 150 –  $f_1(x) + FP(r)$  vs.  $f_2(x) + FP(r)$



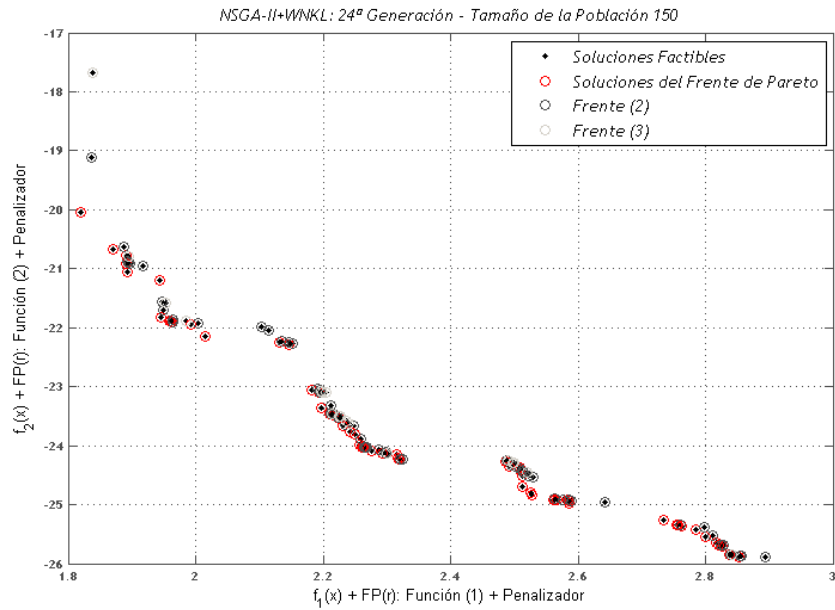
Se observa en la Figura 47, para la generación 18 la población se compone exclusivamente de soluciones factibles ordenadas en los rangos 1,2 y 3. A partir de este punto podemos considerar que la fase de exploración del algoritmo ha concluido y que ha dado inicio a la explotación del espacio de búsqueda. La aptitud media de la población se encuentra en  $[2.35, -23.00]$ .

**Figura 47.** 18ª Generación – Tamaño de la Población 150 –  $f_1(x) + FP(r)$  vs.  $f_2(x) + FP(r)$

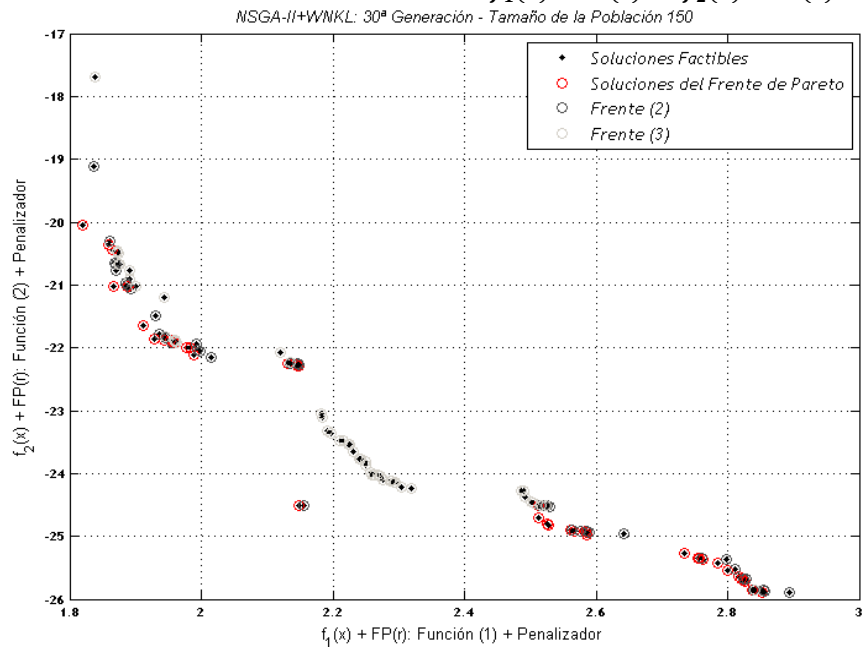


La figura 48 muestra el avance de la población para la generación 24. Se observa como la población gradualmente se va acercando a la esquina inferior izquierda mientras que mantiene sus extremos fijos. Estos extremos corresponden a los valores mínimos y máximos, según sea el caso, para cada una de las funciones objetivo, como si se tratase de una optimización sencilla. La aptitud media de la población en este punto es de  $[2.32, -23.62]$ . Se observa que la población tiende hacia minimizar la primera función objetivo, mientras que la eficiencia térmica ha observado solo una mejoría simbólica. Cosa que era de esperarse teniendo en cuenta que el peso se encuentra solamente limitado por las dimensiones del motor, mientras que la eficiencia térmica está limitada por las leyes de la termodinámica.

**Figura 48.** 24ª Generación – Tamaño de la Población 150 –  $f_1(x) + FP(r)$  vs.  $f_2(x) + FP(r)$



**Figura 49.** 30ª Generación – Tamaño de la Población 150 –  $f_1(x) + FP(r)$  vs.  $f_2(x) + FP(r)$



La Figura 49 muestra la población al final de las 30 generaciones. Se observa que aun existen soluciones en los frentes (2) y (3) lo que quiere decir que el frente de Pareto no se encuentra aún completamente desarrollado, pero que por cuestiones de coste computacional ha sido preciso detener.

Los resultados de la corrida se presentan en la tabla 6, distinguiendo el valor de la aptitud media para los individuos del frente de Pareto, para la población completa y los valores máximos y mínimos observados.

**Tabla 6.** Aptitud media de la población a lo largo de 36 generaciones

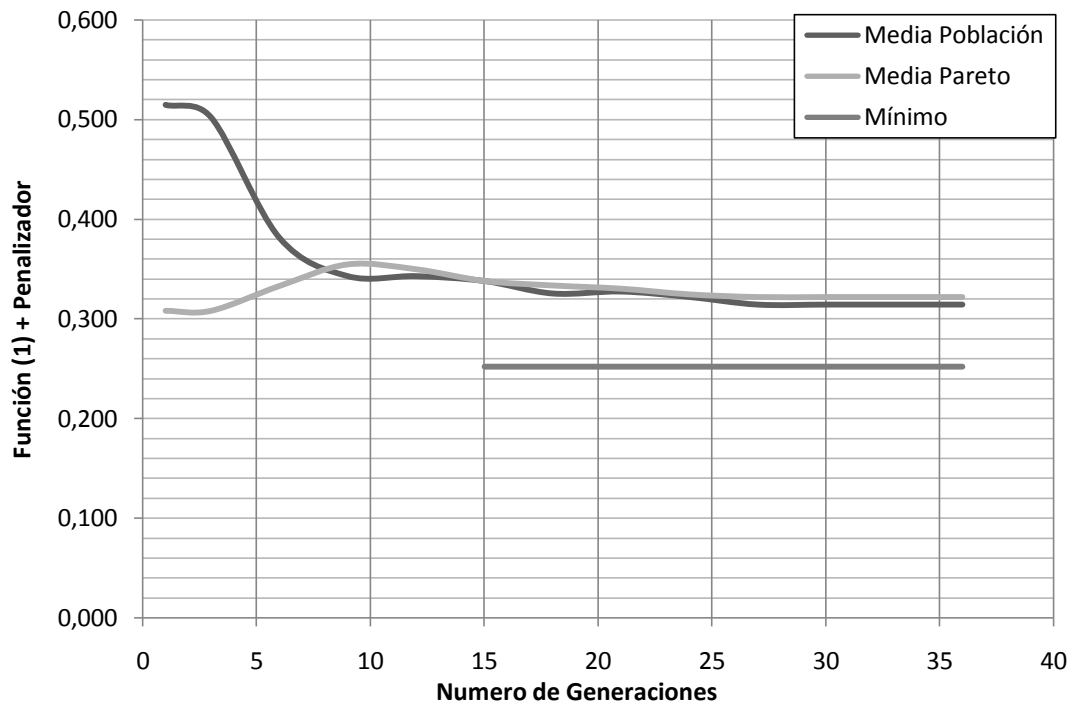
Gen	%F	%NF	%Pareto	AMPob <sup>25</sup>		AMPar <sup>26</sup>		$f_{1min}(x)$	$f_{2min}(x)$
1	41%	59%	3%	0.515	-8.942	0.308	-21.637	0.237	-23.005
3	41%	59%	3%	0.502	-7.140	0.308	-21.637	0.237	-23.005
6	87%	12%	5%	0.381	-22.30	0.333	-23.286	0.237	-25.730
9	96%	4%	15%	0.343	-23.47	0.355	-24.270	0.237	-25.896
12	96%	4%	18%	0.343	-23.47	0.350	-24.023	0.237	-25.896
15	100%	0%	30%	0.338	-23.13	0.338	-23.687	0.252	-25.896
18	100%	0%	36%	0.326	-23.00	0.334	-23.527	0.252	-25.896
21	100%	0%	49%	0.328	-23.22	0.330	-23.399	0.252	-25.896
24	100%	0%	42%	0.322	-23.62	0.325	-23.762	0.252	-25.896
27	100%	0%	31%	0.314	-23.26	0.322	-23.450	0.252	-25.896
30	100%	0%	31%	0.314	-23.26	0.322	-23.450	0.252	-25.896
33	100%	0%	31%	0.314	-23.26	0.322	-23.450	0.252	-25.896
36	100%	0%	31%	0.314	-23.26	0.322	-23.450	0.252	-25.896

En la figura 50, se observa de manera grafica la variación de la función objetivo (1) a lo largo de las 30 generaciones y como cerca de la 25ª generación la aptitud media se mantiene constante. Este es uno de los indicadores de convergencia del algoritmo, sin embargo, es deseable y el autor sugiere realizar pruebas con un número de generaciones mayor hasta que la población completa se concentre en el frente de Pareto.

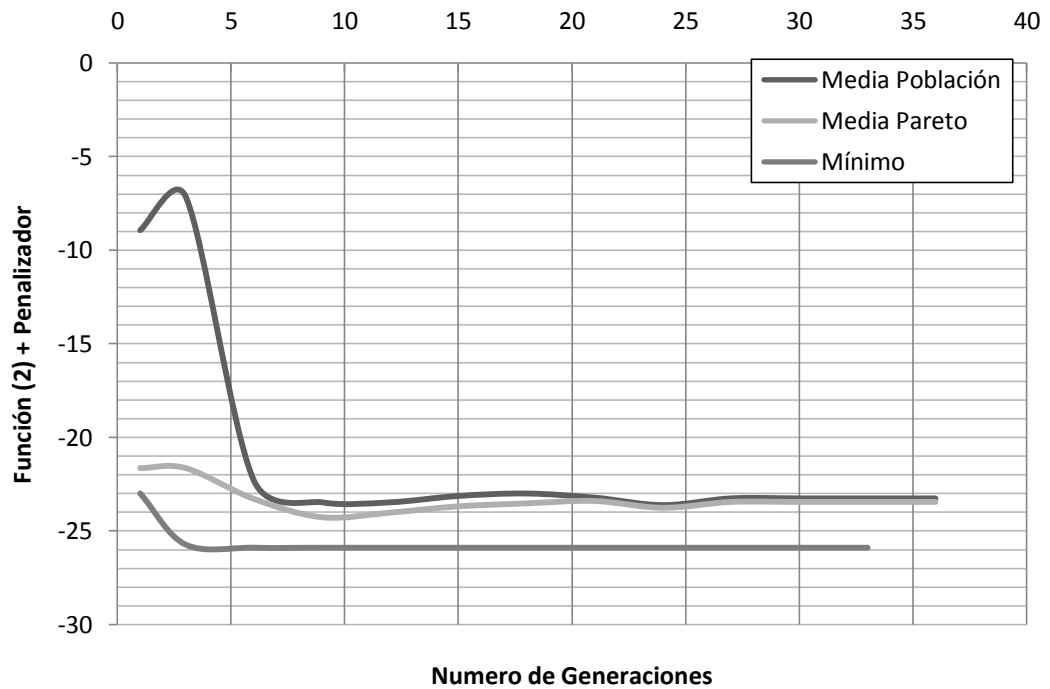
<sup>25</sup> AMPob: Aptitud Media de la Población

<sup>26</sup> AMPar: Aptitud Media del Frente de Pareto

**Figura 50.** Variación de la función (1) a lo largo de 36 generaciones



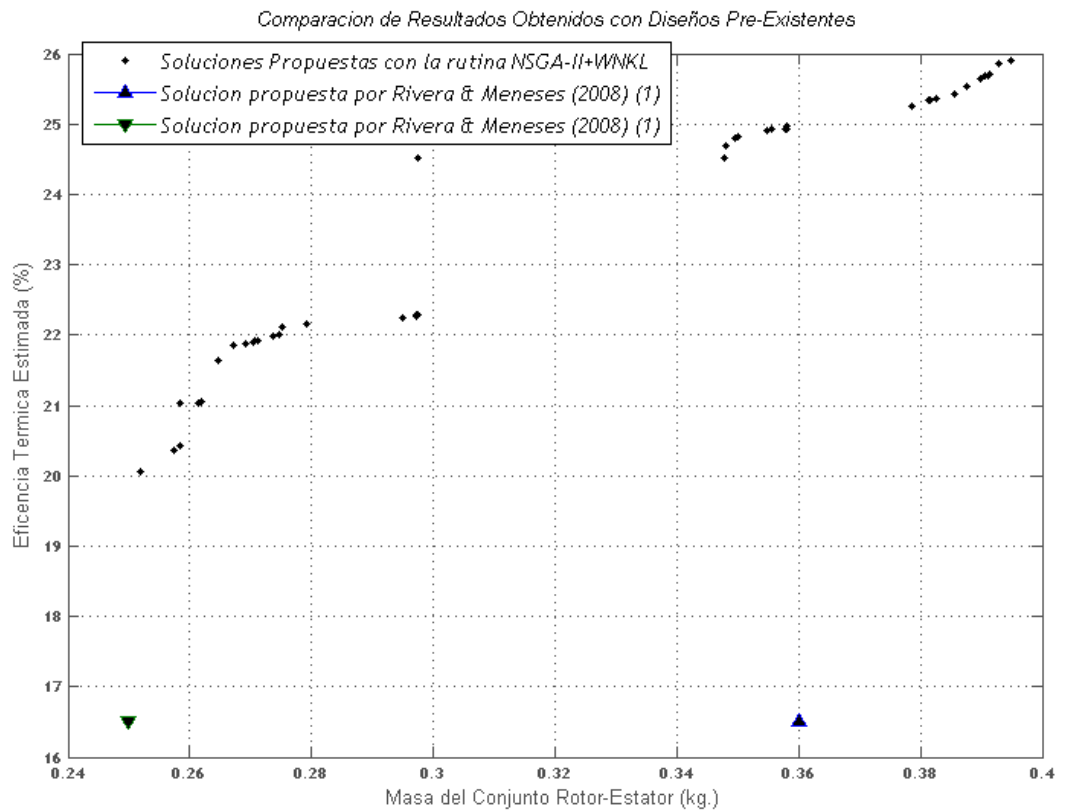
**Figura 51.** Variación de la función (2) a lo largo de 36 generaciones



La Figura 51 muestra la variación de la función objetivo (2). Aquí se observa más fácilmente como la aptitud oscila un poco entre las generaciones 20 y 25, indicando un desplazamiento repentino de la población hacia un mínimo de la función (1) y como la función (1) es más sensible a la variación de parámetros que la eficiencia térmica.

Por último, es preciso hacer la comparación de los datos obtenidos con el diseño propuesto por (Rivera & Meneses, 2008) para efectos de cuantificar la ganancia obtenida. Tenemos dos puntos de diseño dependiendo de la selección de materiales como se muestra en la Figura 52.

**Figura 52.** Comparación de Resultados



Se observa los diseños propuestos (Rivera & Meneses, 2008) resultan particularmente competitivos a nivel de peso, pero la ganancia en cuanto a la eficiencia térmica del modelo es apreciable, donde las soluciones del frente de

Pareto toman valores de entre 20.05% y 25.89%, para obtener una ganancia comparativa de entre 3.55% y 9.39%.

Para el peso del conjunto Rotor-Estator, no nos es posible determinar una ganancia global de esa maneta tan clara como en el caso de la eficiencia, aunque se observan ganancias para soluciones individuales.

El autor propone revisar los modelos y unificar criterios de diseño, en especial en cuanto a la inclusión del engranaje interno en la evaluación del peso, y la metodología de diseño del mismo, para que la comparación sea verdaderamente significativa. Por lo pronto, los resultados son útiles en la medida que las tendencias se mantengan y solo varié la ganancia final.

El frente de Pareto da como resultado 47 individuos de los cuales se seleccionan los que se consideren más significativos para realizar el análisis de sensibilidad de parámetros. Los individuos se encuentran resaltados en la Tabla 7. Para efectos de presentación son omitidos ciertos valores.

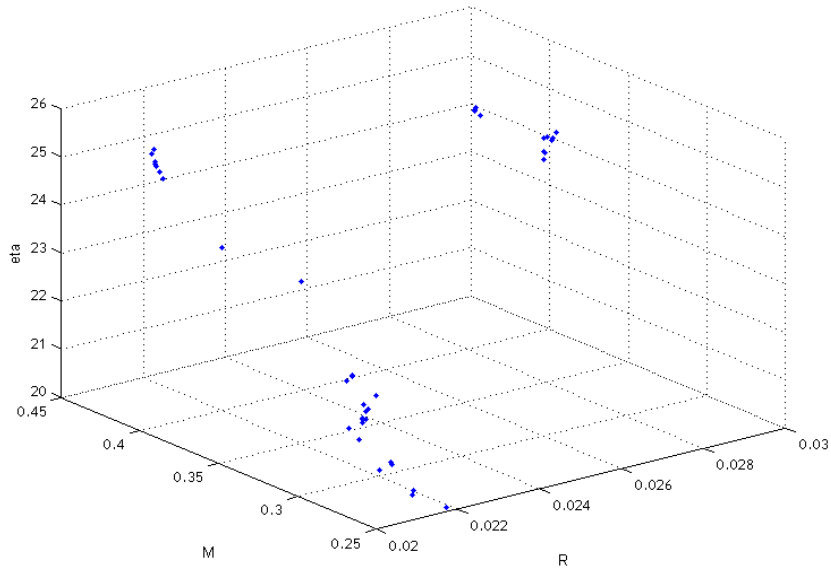
Para facilitar el análisis se opta por graficar la variación de las funciones objetivo en función de los parámetros de diseño como el mostrado en la Figura 53. Aquí se observa, como era de esperarse, una relación directa entre la masa del conjunto Rotor-Estator y el radio generador de la epitrocoide  $R$ . Mientras que para la eficiencia térmica no observamos una tendencia apreciable.

Para la constante de la epitrocoide  $K$ , observamos que los mejores resultados se encuentran para valores de entre 6.2 y 7.2, como se muestra en la Figura 53. Información que sirve para validar parcialmente la recomendación de diseño realizada por (Yamamoto, 1981).

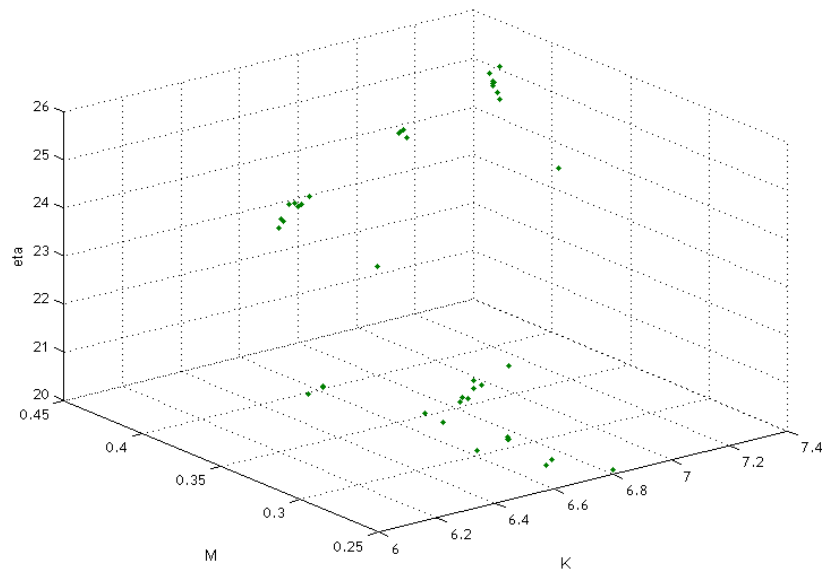
**Tabla 7.** Individuos del Frente de Pareto

	<b>R</b>	<b>r</b>	<b>a</b>	<b>na</b>	<b>e</b>	<b>b</b>	<b>b/e</b>	<b>M</b>	<b>eta</b>
1	21.8	7.41	0.3	36	3.2	15.54	4.86	0.25	20.05%
2	28.0	10.38	0.3	31	4.5	19.93	4.43	0.35	24.80%
3	27.8	10.55	0.3	30	4.5	18.00	4.00	0.35	24.70%
4	20.0	10.50	0.3	42	2.8	18.40	6.57	0.39	25.68%
5	20.0	10.40	0.3	42	2.8	19.60	7.00	0.39	25.64%
6	20.0	10.71	0.3	42	2.8	19.60	7.00	0.39	25.43%
7	27.5	10.24	0.4	44	4.05	24.01	5.93	0.38	25.35%
8	20.0	10.59	0.3	42	2.8	18.40	6.57	0.39	25.71%
9	20.0	10.54	0.3	42	2.8	18.40	6.57	0.39	25.70%
10	27.5	10.27	0.4	44	4.05	24.01	5.93	0.38	25.36%
11	20.0	10.82	0.3	41	2.8	17.20	6.14	0.39	25.86%
12	20.0	10.39	0.3	42	2.8	16.00	5.71	0.39	25.54%
13	27.4	10.22	0.4	45	4.05	24.01	5.93	0.38	25.34%
14	20.1	10.92	0.3	41	2.8	16.00	5.71	0.39	25.90%
15	27.5	10.16	0.4	44	4.05	17.07	4.21	0.38	25.26%
16	28.4	10.53	0.3	31	4.5	19.93	4.43	0.36	24.92%
17	28.3	10.57	0.3	31	4.5	29.57	6.57	0.36	24.93%
18	28.3	10.50	0.3	30	4.5	19.93	4.43	0.35	24.91%
19	28.5	10.67	0.3	30	4.5	19.93	4.43	0.36	24.98%
20	28.3	10.50	0.3	30	4.5	19.93	4.43	0.35	24.91%
21	27.9	10.46	0.3	30	4.5	19.93	4.43	0.35	24.83%
22	28.2	10.53	0.3	31	4.5	31.50	7.00	0.36	24.92%
23	21.0	9.12	0.3	40	3.5	16.25	4.64	0.30	22.25%
24	21.2	9.44	0.3	40	3.5	14.00	4.00	0.30	22.28%
25	21.1	9.14	0.3	40	3.2	12.80	4.00	0.28	22.15%
31	20.0	11	0.3	42	2.8	16.00	5.71	0.35	24.51%
32	21.2	9.43	0.3	40	3.5	14.00	4.00	0.30	22.27%
33	20.2	8.72	0.3	41	3.2	16.23	5.07	0.26	21.64%
34	21.2	7.38	0.3	38	3.2	16.91	5.29	0.26	20.44%
35	20.0	9.25	0.3	41	3.2	15.54	4.86	0.27	21.86%
36	20.7	9.37	0.3	41	3.2	17.60	5.50	0.28	22.12%
40	20.7	9.08	0.3	41	3.2	15.54	4.86	0.27	22.00%
41	20.0	11	0.3	42	3.2	17.60	5.50	0.30	24.51%
42	20.4	8.80	0.3	41	3.2	12.80	4.00	0.27	21.87%
43	20.4	6.96	0.3	37	3.2	16.91	5.29	0.26	21.02%
44	20.5	8.89	0.3	41	3.2	22.40	7.00	0.27	21.92%
45	20.8	6.96	0.3	37	3.2	15.54	4.86	0.26	21.02%
46	21.2	6.98	0.3	38	3.2	21.71	6.79	0.26	20.37%
47	20.5	8.83	0.3	40	3.2	13.49	4.21	0.27	21.89%

**Figura 53.** Grafica de R versus Función Masa versus Eficiencia Térmica

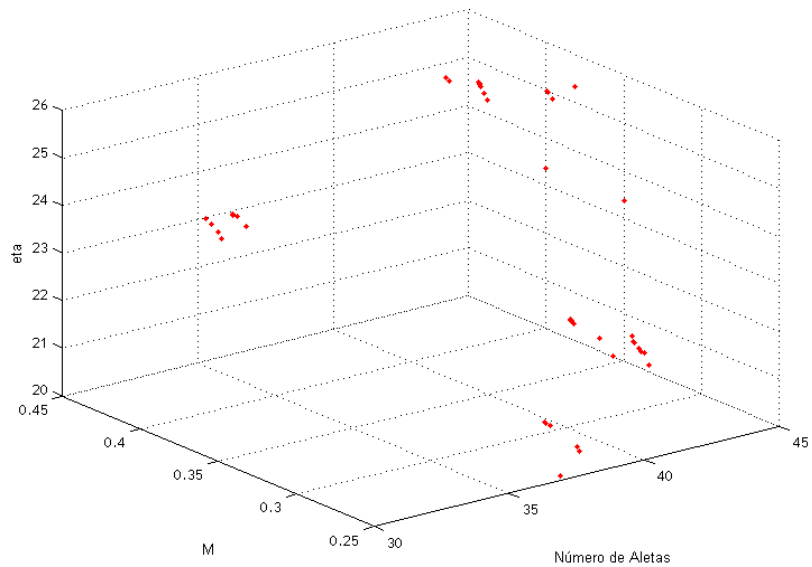


**Figura 54.** Grafica de K versus Función Masa versus Eficiencia Térmica



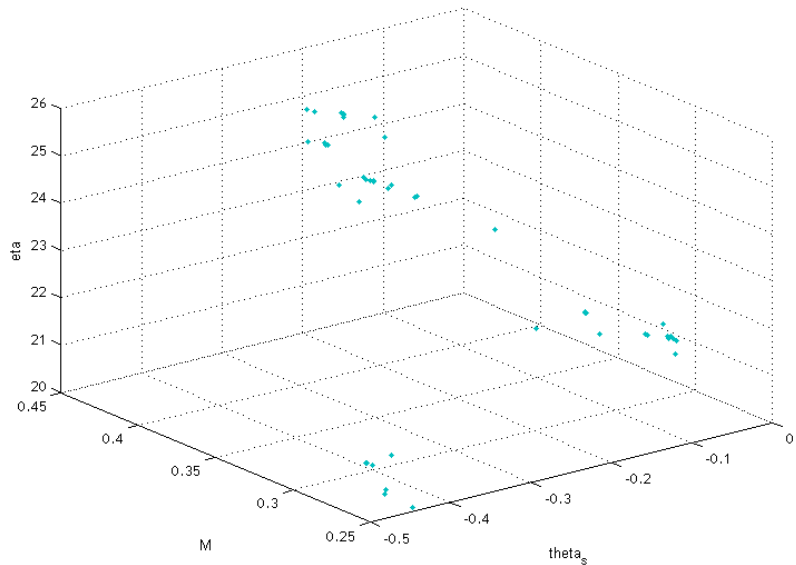
Para el número de aletas, observamos en la figura 55 una ligera proporcionalidad para ambas funciones, en función del número de aletas, donde solo unos cuantos individuos escapan a esta tendencia.

**Figura 55.** Grafica del número de aletas versus Función Masa versus Eficiencia Térmica



La Figura 56, muestra una relación directa entre el adelanto de la ignición con la eficiencia térmica.

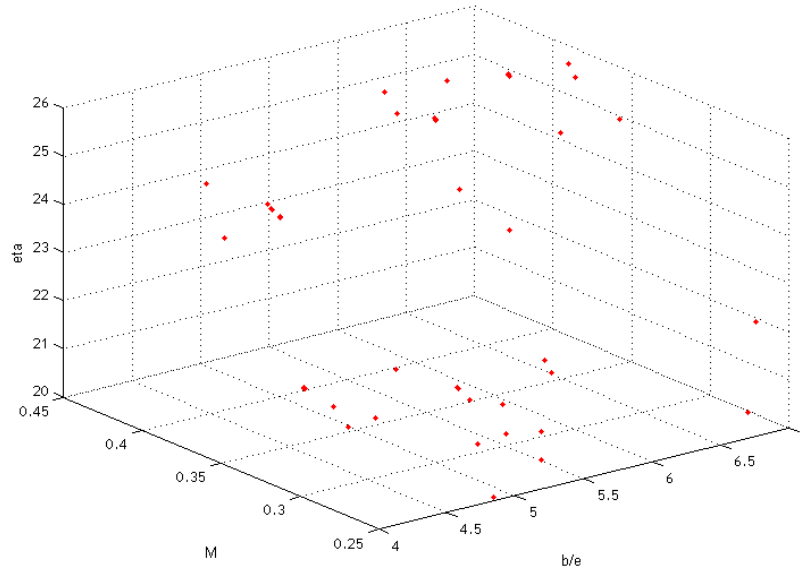
**Figura 56.** Grafica del ángulo de encendido versus Función Masa versus Eficiencia Térmica



Por último, la Figura 57 no muestra ninguna tendencia apreciable para alguna de las funciones objetivo con respecto a la relación entre el espesor de la carcasa y su excentricidad. Esto se hacía con el propósito de validar la recomendación de

diseño propuesta por (Yamamoto, 1981) de mantener una relación espesor/excentricidad igual a 5.

**Figura 57.** Grafica de la relación b/e versus Función Masa versus Eficiencia Térmica



El autor propone aumentar el tamaño de la población y el número de generaciones para tomar más puntos para la validación de las recomendaciones de diseño y el análisis de sensibilidad de parámetros.

## 5. CONCLUSIONES

Surgen una serie de observaciones acerca de la metodología de diseño y del algoritmo de optimización, en base a esto, podemos concluir que:

### **Respecto al algoritmo de optimización**

- La utilización de algoritmos genéticos multiobjetivo (MOOGA)<sup>27</sup> en problemas reales, resulta viable y sus resultados comparativos son aceptables. Sin embargo, se precisa un análisis más profundo para la obtención de un algoritmo genético orientado específicamente hacia el diseño mecánico.
- La utilización del sorteo no-dominado como alternativa para la selección de los individuos facilita el control y posterior eliminación de las soluciones no-factibles de la población, sin requerir modificaciones extensivas en la programación.
- La integración de los algoritmos genéticos con el método de análisis por elementos finitos sirve como complemento y esta acorde a metodologías de diseño. Sin embargo, el elevado costo computacional para obtener la solución de dichos análisis pone un límite práctico a la aplicabilidad de los algoritmos genéticos con el análisis por elementos finitos.

### **Respecto al modelo matemático del motor rotativo Wankel**

- La utilización de modelos cero-dimensionales simplifica en buena medida los cálculos requeridos a costa de precisión en los resultados obtenidos. Además estos modelos en su mayoría poseen abstracciones y simplificaciones que se hacen pensando en motores alternativos.

---

<sup>27</sup> Multi-Objective Optimization Genetic Algorithms - MOOGA

- La utilización de parámetros calibrados especialmente para motores reciprocantes en las simulaciones, resta credibilidad a los resultados obtenidos por estas rutinas.
- Es necesaria la inclusión del proceso de admisión y de expulsión de gases en la cámara de combustión para complementar el modelo.
- La inclusión de un análisis multifísico en el diseño de componentes mecánicos permite recuperar en buena medida el sentido físico del fenómeno que se pierde cuando se recurre a una norma estandarizada. Además, es una alternativa viable cuando no existen normas o recomendaciones de diseño aplicables.
- El análisis por elementos finitos aplicado a la carcasa no permite el aprovechamiento del procesamiento en paralelo porque los solucionadores incluidos en el paquete de análisis se encuentran limitados a trabajar con matrices simétricas. Sin embargo, la implementación en paralelo funciona y sirve como plataforma para trabajos futuros que puedan aprovechar los beneficios de esta arquitectura.
- El conjunto de soluciones obtenidas resulta muy pequeño y no permite apreciar de manera concluyente la sensibilidad del modelo a la variación de parámetros.
- Los resultados obtenidos de las simulaciones no son representativos a menos que sean validados de alguna manera por datos experimentales, luego nos hemos limitado a un análisis comparativo de tendencias con la información obtenida en la literatura presentada.

## 6. RECOMENDACIONES

La utilización de algoritmos de optimización en el diseño mecánico es un objeto de estudio sumamente complejo y cuya aplicabilidad en la industria local no ha empezado a explorarse. Es autor desea notar que este trabajo de grado no tiene como objeto principal la obtención de un diseño apto para la construcción, sino que busca presentar una metodología de diseño alternativa que integre los avances recientes en el campo de las ciencias de la computación y comparar sus resultados con los obtenidos mediante metodologías convencionales. El trabajo en ningún momento entrega un diseño que se pueda asegurar es óptimo, pero ilustra la metodología de obtención de una solución aproximada a este.

Como se menciona antes, este trabajo de grado es la continuación de una línea de proyectos de investigación que pretende servir como plataforma para trabajos futuros. Teniendo esto en cuenta, el autor desea hacer las siguientes recomendaciones para quien desee realizar investigaciones futuras en esta área.

- Estudiar la construcción de un banco de pruebas y de un sistema de adquisición de datos que permita la calibración de los parámetros empíricos utilizados en el modelo.
- Elaborar una comparación cuantitativa del rendimiento y el nivel de aproximación para diferentes algoritmos de optimización multiobjetivo, para la elaboración o modificación de un algoritmo orientado hacia el diseño mecánico.
- Adecuar un centro de cómputo orientado a la computación en paralelo es fundamental para la continuidad de esta línea de investigación.
- Modificar el algoritmo existente para que almacene todas las soluciones evaluadas, de esta manera se incrementa el tamaño de la muestra se incrementa dramáticamente para el análisis de sensibilidad.

- Modificar los parámetros del algoritmo y evaluar los resultados para un mayor número de iteraciones y mejorar la calidad de los resultados obtenidos.
- Solicitar financiación interna y externa para la elaboración de proyectos en esta línea de trabajo a nivel de pregrado, maestría e incentivar las publicaciones técnicas para la reclasificación del grupo de investigación.

## BIBLIOGRAFÍA

- Alvarez, L. F. (2000). *Design Optimization Based on Genetic Programming*. Tesis Doctoral, University of Bradford, Department of Civil and Environmental Engineering, UK.
- ANSYS, Inc. (2005). *ANSYS Elements Reference: ANSYS Release 10.0*. Canonsburg.
- ANSYS, Inc. (2005). *Distributed ANSYS Guide: ANSYS Release 10.0*. Canonsburg.
- Arenas Jimenez, A., & Serrano Arango, D. O. (2000). *La Computación Evolucionaria En La Optimización Del Diseño Mecánico: Aplicación De Los Algoritmos Genéticos En La Optimización De Una Bomba De Pistones Axiales De Desplazamiento Fijo*. Tesis (Pregrado), Universidad Industria de Santander, Escuela de Ingeniería Mecánica, Bucaramanga.
- Argonne National Laboratory Group. (2002). *MPICH2: High-performance and Widely Portable MPI*. Recuperado el Octubre de 2008, de <http://www.mcs.anl.gov/research/projects/mpich2/>
- Borman, G., & Nishiwaki, K. (1987). Internal Combustion Heat Transfer. *Progress in Energy and Combustion Science* , 13, 1-46.
- Buttsworth, D. R. (2002). *Spark Ignition Internal Combustion Engine Modelling using Matlab*. University of Southern Queensland.
- Daly, C. H. (03 de Enero de 2002). *ME 356 Machine Design Analysis*. Recuperado el 08 de Agosto de 2008, de University of Washington: <http://courses.washington.edu/mengr356/>
- Danieli, G. A. (1976). *A performance model of a Wankel engine, including the effects of burning rates, heat transfer, leakage and quenching compared with measured pressure time histories*. Tesis Doctoral, Massachusetts Institute of Technology.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A Fast Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* , 2, 182-197.
- Ferguson, C. R. (1986). *Internal Combustion Engines*. Wiley.

- Fernández, P. R. (11 de Junio de 2002). *Ingeniería Energética*. Recuperado el 06 de Agosto de 2008, de <http://personales.ya.com/universal/TermoWeb/index.html>
- Gen, M., & Runwei, C. (2000). *Genetic Algorithms and Engineering Design*. Ashikaga, Japón: John Wiley & Sons, Inc.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. USA: Addison–Wesley Publishing Company.
- Gordon, S., & McBride, B. J. (1971). *Computer Program for Calculation of Complex Chemical Equilibrium Composition*. NASA Publication.
- Heywood, J. B. (1988). *Internal Combustion Engine Fundamentals*. McGraw-Hill.
- JANAF. (1971). JANAF Thermochemical Tables. *U.S National Bureau of Standards Publications* .
- Jones, J. B., & Dugan, R. E. (1997). *Engineering Thermodynamics, First Edition*. New York: Prentice-Hall.
- Kino. (2006). Kolbenbewegung am Wankelmotor.
- Kratz, T., & Wackwitz, G. (1998). *Entwicklung, Konstruktion, Fertigung und Erprobung des Prototypen eines Wankelmotors für den Einsatz im Flugmodellbau*. Berlin: Technische Fachhochschule Berlin.
- Kundle, D. (2005). *A Summary and Comparison of MOEA Algorithms*.
- Mills, A. F. (1999). *Heat Transfer* (2nd ed. ed.). Upper Saddle River, NJ: Prentice Hall.
- Moaveni, S. (1999). *Finite Element Analysis: Theory and Application with ANSYS*. Upper Saddle River, New Jersey: Prentice Hall.
- Norman, T. J. (1983). *A Performance Model of a Spark Ignition Wankel Engine*. Tesis de Maestria, Massachusetts Institute of Technology, Massachusetts.
- Nye, J. F. (1957). *Physical Properties of Crystals: Their Representation by Tensors and Matrices*. USA: Oxford University Press.
- Olikara, C., & Borman, G. L. (1975). *Calculating Properties of Equilibrium Combustion Products with Some Applications to I.C. Engines*. SAE Paper.

- Pedregal, P. (1963). *Introduction to Optimization*. Ciudad Real, España: Springer-Berlag.
- Poli, R., Langdon, W. B., & McPhee, N. F. (2008). *A Field Guide to Genetic Programming*.
- Raju, M. S., & Willis, E. A. (1990). *Analysis of Rotary Engine Combustion Processes Based on Unsteady, Three-Dimensional Computations*. NASA Lewis Research Center. Cleveland, Ohio: NASA Technical Memorandum.
- Ramos, J. I. (1989). *Internal Combustion Engine Modelling*. Pittsburgh: Hemisphere Publishing Corporation.
- Rivera, D., & Meneses, Y. (2008). *Material Educativo Virtual y Diseño con su respectivo proceso de manufactura de un Motor Rotativo Tipo Wankel*. Tesis (Pregrado), Universidad Industrial de Santander, Escuela de Ingeniería Mecánica, Bucaramanga.
- Rodriguez Cañon, J. E. (2004). *Aproximacion al Diseño Óptimo del Chasis de una Motocicleta utilizacion Análisis de Elementos Finitos y Algoritmos Evolutivos*. Tesis (Pregrado), Universidad de los Andes, Ingeniería Industrial, Bogotá, DC.
- Rothlauf, F. (2006). *Representations for Genetic and Evolutionary Algorithms* (2ª Edición ed.). Mannheim: Springer.
- Sastry, K. N. (2007). *Genetic Algorithms and Genetic Programming for Multiscale Modelling; Applications in Material's Science and Chemistry and Advances in Scalability*. Urbana: Illinois Genetic Algorithms Laboratory.
- Seshadri, A. (2006). *Multi-Objective Optimization Using Evolutionary Algorithms (MOEA)*.
- Velez Langs, O. E. (1999). *Aplicación De Los Algoritmos Genéticos En Problemas De Optimización Combinatoria*. Universidad Industrial de Santander. Bucaramanga: Escuela de Ingeniería de Sistemas e Informática.
- Yamamoto, K. (1981). *Rotary Engine*. Tokyo, Japón: Sankaido.

## APÉNDICES

### A.1. ALGORITMO DE OPTIMIZACIÓN – (NSGA-II-WNKL.m)

El algoritmo de optimización se programo con base en el algoritmo NSGA-II presentado por (Deb, Pratap, Agarwal, & Meyarivan, 2002) tomando como referencia la implementación del mismo realizada por (Seshadri, 2006), del cual se presentara una descripción detallada.

#### · Inicializar Población (+GA/inicializar\_poblacion.m)

La población inicial se obtiene de manera aleatoria con distribución normal teniendo en cuenta el dominio de las variables y las restricciones si estas aplican. Esta información se encuentra almacenada en un archivo aparte que controla el número de variables, tanto continuas como discretas, y los valores que pueden tomar.

```
function [vmin,vmax,inter,lmin,lmax,linter,binmin,binmax] = valeurs(L)

%%% Pre-Indexar Variables
vmin    = zeros(1,7);   vmax = zeros(1,7);
lmin    = zeros(1,3);   lmax = zeros(1,3);

%%% Definir Valores Máximos y Mínimos para las Variables Continuas
vmin(1) = 20e-3;       vmax(1) = 25e-3;% Radio Generador
vmin(2) = 7;           vmax(2) = 11;   % Relación de Compresión Real
vmin(3) = 0.2e-3;     vmax(3) = 6e-4; % Separación de los Sellos
vmin(4) = -55;        vmax(4) = -15;  % Encendido de la Chispa
vmin(5) = -210;       vmax(5) = -190; % Cierre lumbrera de Admisión
vmin(6) = 190;        vmax(6) = 210;  % Apertura lumbrera de Escape
vmin(7) = 30;         vmax(7) = 46;   % Numero de Aletas

%%% Definir Valores Máximos y Mínimos para las Variables Discretas
lmax(1) = 11;          % Excentricidad del Rotor
lmax(2) = 14;          % Espesor de la Carcasa
lmax(3) = 3;           % Material de la Carcasa

inter    = vmax - vmin;
linter   = lmax - lmin;
binmin   = [];        binmax  = [];
for i = 1:size(lmax,2)
    binmin = [binmin dec2bin(lmin(i),L(i))];
    binmax = [binmax dec2bin(lmax(i),L(i))];
end
```

Donde la variable  $L$  es un vector de longitud igual al número de parámetros que van a emplear representación binaria y contiene la longitud de las cadenas de cada uno de los parámetros.

```
function [real,bin] = inicializar_poblacion(N,L)

%% Obtener el dominio de las variables de diseño
import GA.valeurs
[vmin,vmax,inter,lmin,lmax,linter] = valeurs(L);

%% Generar de manera aleatoria el parametro
real = ones(N,1)*vmin + rand(N,size(vmin,2)).*(ones(N,1)*inter);
bin = [];
parfor i = 1:size(lmax,2)
    bin = [bin,dec2bin(ceil(linter(i)*rand(N,1)),L(i))];
end
```

Las variables *real* y *bin* son almacenadas como características o propiedades de una variable llamada población para así facilitar su manipulación y transporte.

- Ordenamiento y Distancia de Separación (+GA/fast\_sorting\_algorithm.m)

El funcionamiento general del FSA – por sus siglas en ingles, *Fast Sorting Algorithm* – se muestra a continuación:

- Para cada individuo  $n$  en la población de tamaño  $N$  se hace lo siguiente:
  - Reiniciar un vector  $S_p$  que contendrá la lista de individuos en la población que sean dominado por  $n$ .
  - Reiniciar un contador  $P_n$  que llevara registro del número de individuos que predominan sobre  $n$ .
  - Para cada individuo  $q$  en la población  $N$  se repite que:
    - Si  $n$  domina a  $q$  entonces añádase  $q$  al vector  $S_p$ ;  $S_p = S_p \cup \{q\}$
    - Si  $q$  domina a  $n$  entonces incrementese el contador  $P_n$  una unidad.
  - Si  $P_n = 0$ , es decir ningún individuo de la población domina a  $n$ , entonces  $n$  pertenecerá al primer frente y se le asigna un rango de 1. El control de los individuos que pertenecen al primer frente se lleva mediante un vector  $F_1$  tal que  $F_1 = F_1 \cup \{n\}$ .

- Se reinicia el contador de frentes a 1.
- Lo siguiente se repite mientras el  $i$ -ésimo frente contenga elementos, es decir mientras  $F_i \neq [\emptyset]$ :
  - Reiniciar el vector  $Q$  que contendrá la lista de los parámetros del  $(i+1)$ -ésimo frente.
  - Para cada individuo  $n$  contenido en el frente  $F_i$  se repite que:
    - Para cada individuo  $q$  contenido en el vector  $S_p$  de  $n$ :
      - Hay un decremento en el contador de individuos que dominan a  $q$  tal que  $P_q = P_q - 1$
      - Si  $P_q = 0$ , entonces ninguno de los individuos en los frentes restantes domina a  $q$ , por ende podemos asignar a  $q$  al frente  $i + 1$ , luego  $Q = Q \cup \{q\}$  y  $q_{rank} = i + 1$ .
  - Se incrementa el contador de frentes una unidad.
  - Se almacenan los individuos contenidos en  $Q$  en  $F_i$ .

Una vez concluido el ordenamiento de la población pasamos a asignar la distancia de separación. Ya que el proceso de selección se hace dando prioridad al frente al que pertenece la solución y luego a la distancia de separación, se concluye que necesariamente la asignación de dicha distancia debe realizarse entre individuos del mismo frente. El procedimiento para calcular la separación es el siguiente:

- Para cada uno de los frentes  $F_i$  con  $n$  individuos se repite que:
  - Se reinicia asigna un valor de separación cero a todos los individuos del frente, es decir que  $F_i(d_j) = 0$ , donde  $j$  corresponde al  $j$ -ésimo individuo del frente  $i$ .
  - Para cada una de las funciones objetivo  $m$  se repite que:
    - Se ordenan los individuos del frente  $F_i$  basados en el valor del objetivo  $m$ , y se almacenan en un vector  $I$ .
    - Se asigna un valor de infinito a los límites máximos y mínimos de la función objetivo.  $I(d_1) = \infty$ ,  $I(d_n) = \infty$ .

- Para  $k = 2$  hasta  $(n - 1)$ :
  - $I(d_k) = I(d_k) + \frac{I(m_{k+1}) - I(m_{k-1})}{m_{max} - m_{min}}$
  - Donde  $I(m_k)$  corresponde al valor del m-esimo objetivo del k-esimo individuo en el vector  $I$ .

Por último aplicamos sumamos uno al rango de los individuos que violen las restricciones del diseño.

```
function x = fast_sorting_algorithm(x)

rank      = 1;
N         = size(x.real,1);
m         = size(x.adaptation,2);
x.rank    = zeros(N,1);
F(rank).f = [];
solucion  = [];

%%% Determinar el primer frente
for n = 1:N
    solucion(n).n = 0;    solucion(n).p = [];
    for q = 1:N
        comme = zeros(1,2);
        for p = 1:m
            if x.adaptation(n,p) > x.adaptation(q,p)
                comme(1) = comme(1) + 1;
            elseif x.adaptation(n,p) < x.adaptation(q,p)
                comme(2) = comme(2) + 1;
            end
        end
        end
        if comme(1) == 0
            solucion(n).p = [solucion(n).p q];
        elseif comme(2) == 0
            solucion(n).n = solucion(n).n + 1;
        end
    end
    if solucion(n).n == 0
        x.rank(n,1) = 1;    F(rank).f = [F(rank).f n];
    end
end
for i = 1:N
    if x.penalty(i) == 1
        x.rank(i) = x.rank(i) + 1;
    end
end
%%% Determinar los siguientes frentes
while ~isempty(F(rank).f)
    Q = [];
```

```

for i = 1:length(F(rank).f)

    if ~isempty(solucion(F(rank).f(i)).p)
        for j = 1: length(solucion(F(rank).f(i)).p)
            solucion(solucion(F(rank).f(i)).p(j)).n = ...
                solucion(solucion(F(rank).f(i)).p(j)).n - 1;
            if solucion(solucion(F(rank).f(i)).p(j)).n == 0
                Q = [Q solucion(F(rank).f(i)).p(j)];
                x.rank(solucion(F(rank).f(i)).p(j),1) = rank + 1;
            end
        end
    end
end
rank = rank + 1;
F(rank).f = Q;
end

%%% Determinar la Diversidad de la Solución
for i = 1:(rank - 1)
    Y = [];
    temp = F(i).f;
    for j = 1:size(temp,2)
        Y = [Y; x.adaptation(temp(j),:)]];
    end
    dist = zeros(size(Y,1));
    for k = 1:m
        [I,index] = sortrows(Y,1);
        dist(1) = inf; dist(size(Y,1)) = inf;
        if (size(Y,1) >= 3)
            for l = 2:(size(Y,1)-1)
                maxmin = I(1,k) - I(size(Y,1),k);
                dist(l) = dist(l) + (I(l+1,k) - I(l-1,k))/maxmin;
            end
        end
    end
end
for j = 1:size(temp,2)
    x.distance(temp(index(j)),1) = dist(j);
end
end

```

· **Selección por Torneo (+GA/seleccion\_torneo.m)**

```

function [X] = seleccion_torneo(x,piscine,tournoi)

pool = zeros(1,piscine);

for i = 1:piscine
    c = zeros(1,tournoi);           % Reiniciar los Vectores
    r = zeros(1,tournoi);
    d = zeros(1,tournoi);
    for j = 1:tournoi
        c(j) = ceil(2*piscine*rand(1)); % Seleccion Aleatoria
        r(j) = x.rank(c(j));
    end
end

```

```

        d(j)      = x.rank(c(j));
    end

    cmin = find(r == min(r));

    if length(cmin) == 1
        pool(i) = c(cmin);
    else
        cmax = find(d(cmin) == max(d(cmin)));
        if length(cmin) ~= 1
            cmax = cmax(1);
        end
        pool(i) = c(cmin(cmax));
    end
end

X.real = x.real(pool,:);    X.bin = x.bin(pool,:);

```

· Operadores Genéticos (+GA/genetic\_operator.m)

La subrutina se encarga de seleccionar de manera aleatoria los individuos, aplicar según sea el caso el operador genético respectivo y asegurarse de que las cadenas resultantes se encuentren dentro del dominio de la función.

```

function [offspring] =
genetic_operator(x,mu,mum,vmax,vmin,lmax,binmax,binmin,L,crossover)

real      = x.real;
bin       = x.bin;
[N,n]     = size(bin);
p         = 1;

function bq = bethaq(parent)
function delta = deltaq(parent)
function [preal,pbin] = genparent()
function [o1real,o2real,p1bin,p2bin] = SBX(vmax,vmin)
function [o1bin,o2bin] = f2PX_V1(p1bin,p2bin)
function [bin] = feasbin(bin,i)
function [oreal,obin] = PX(vmax,vmin)
function [obin] = P1(obin)

while p < 2*N
%%% SBX + 2PX_V1
    if rand(1) < crossover
        [offspring.real(p,:),offspring.real(p+1,:),p1bin,p2bin] ...
            = SBX(vmax,vmin);
        [offspring.bin(p,:),offspring.bin(p+1,:)] ...
            = f2PX_V1(p1bin,p2bin);
        for j = 1:size(lmax,2)
            [offspring.bin(p,:)] = feasbin(offspring.bin(p,:),j);
        end
    end
    p = p + 1;
end

```

```

        [offspring.bin(p+1,:)] = feasbin(offspring.bin(p+1,:),j);
    end
    p = p + 2;
    %%% PX + P1
    else
        [offspring.real(p,:),olbin] = PX(vmax,vmin);
        [offspring.bin(p,:)] = P1(olbin);
        for j = 1:size(lmax,2)
            [offspring.bin(p,:)] = feasbin(offspring.bin(p,:),j);
        end
        p = p + 1;
    end
end
end

```

Los operadores genéticos son definidos como funciones implícitas dentro del archivo *genetic\_operator.m* y otras que también se emplean para facilitar la representación.

### *Selección Aleatoria de un Individuo - genparent*

```

function [preal,pbin] = genparent()
    parent = ceil(N*rand(1));
    preal = real(parent,:);
    pbin = bin(parent,:);
end

```

### *Funciones $\bar{\beta}(u)$ y $\delta_k(r_k)$ – bethaq y deltaq*

```

function bq = bethaq(parent)
    u = rand(1,length(parent));
    if u <= 0.5
        bq = (2*u).^(1/(mu+1));
    else
        bq = (1./(2*(1 - u))).^(1/(mu+1));
    end
end
function delta = deltaq(parent)
    r = rand(1,length(parent));
    if r <= 0.5
        delta = (2*r).^(1/(mum+1)) - 1;
    else
        delta = 1 - (2*(1 - r)).^(1/(mum+1));
    end
end
end

```

### *Recombinación Binaria Simulada - SBX*

```

function [olreal,o2real,p1bin,p2bin] = SBX(vmax,vmin)
    [p1real,p1bin] = genparent();
    [p2real,p2bin] = genparent();
    bq = bethaq(p1real);
    olreal = 0.5*((1 + bq).*p1real + (1 - bq).*p2real);
    olreal = max(olreal,vmin);
    olreal = min(olreal,vmax);

```

```

    o2real = 0.5*((1 - bq).*p1real + (1 + bq).*p2real);
    o2real = max(o2real,vmin);
    o2real = min(o2real,vmax);
end

```

## Mutación Polinomial - PX

```

function [oreal,obin] = PX(vmax,vmin)

    [preal,obin] = genparent();
    delta = deltaq(preal);
    oreal = preal + (vmax - vmin).*delta;
    oreal = max(oreal,vmin);
    oreal = min(oreal,vmax);
end

```

## Cruce de 2 Puntos – f2PX\_V1

```

function [o1bin,o2bin] = f2PX_V1(p1bin,p2bin)

    u = floor(n*rand(1));
    v = floor((n-u)*rand(1));
    if u == 0
        o1bin = [p2bin(:,1:v) p1bin(:,(v+1):n)];
        o2bin = [p1bin(:,1:v) p2bin(:,(v+1):n)];
    else
        o1bin = [p1bin(:,1:u) p2bin(:,(u+1):(u+v)) ...
                p1bin(:,(u+v+1):n)];
        o2bin = [p2bin(:,1:u) p1bin(:,(u+1):(u+v)) ...
                p2bin(:,(u+v+1):n)];
    end
end

```

## Mutación de 1 bit – P1

```

function [obin] = P1(obin)

    u = ceil(n*rand(1));
    if obin(u) == '0'
        obin(u) = '1';
    else
        obin(u) = '0';
    end
end

```

## Corrección de Cadenas No-Factibles - feasbin

```

function [bin] = feasbin(bin,i)

    init = sum(L(1:(i-1))) + 1;
    finit = sum(L(1:i));

```

```

temp = bin2dec(bin(init:finit));
if temp > lmax(i)
    bin(init:finit) = binmax(init:finit);
elseif temp < 1
    bin(init:finit) = binmin(init:finit);
end
end
end

```

- Mecanismo de Selección (+GA/crowded\_comparison\_operator.m)

Esta rutina se encarga de escoger los individuos mejor adaptados de entre la población mixta de padres y vástagos de acuerdo al rango y a la distancia de separación.

```

function [y] = crowded_comparison_operator(x,N)

count = 0;
[a,index] = sort(x.rank);
max_rank = a(size(index,1));

for i = 1:max_rank
    c = find(x.rank == i);
    p = count + 1;
    count = count + size(c,1);
    if count < N
        y.real(p:count,:) = x.real(c,:);
        y.bin(p:count,:) = x.bin(c,:);
        y.adaptation(p:count,:) = x.adaptation(c,:);
        y.rank(p:count,:) = x.rank(c,:);
        y.penalty(p:count,:) = x.rank(c,:);
        y.distance(p:count,:) = x.distance(c,:);
    else
        [restante,indexb] = sort(x.distance(c),'descend');
        j = 1;
        while (p <= N)
            y.real(p,:) = x.real(c(indexb(j)),:);
            y.bin(p,:) = x.bin(c(indexb(j)),:);
            y.adaptation(p,:) = x.adaptation(c(indexb(j)),:);
            y.rank(p,:) = i;
            y.penalty(p,:) = i;
            y.distance(p,:) = restante(j);
            j = j + 1 ;
            p = p + 1;
        end
        break
    end
end
end

```

- Función de Penalización (+GA/penalty.m)

La función de penalización verifica el cumplimiento de las restricciones y asigna un valor proporcional a la violación de las mismas a los individuos no-factibles. De igual forma, a medida que se avanza el algoritmo se vuelve menos receptivo a los individuos no-factibles e incrementa la penalización de manera gradual.

```
function PF = penalty(pot,FS,R,e,b,am,V_ad,area_rotor,r,r_t,Fw,Ew,G)

pf = zeros(1,7); K = R/e;

potmax = 2.5; % HP
potmin = 1.45; % HP
SFmin = 1.2; % 1/1
Kmax = 14; % 1/1
Kmin = 6; % 1/1

if pot < potmin
    pf = 5*(potmin - pot)/potmin;
elseif pot > potmax
    pf = 5*(pot - potmax)/potmax;
end
if FS < SFmin
    pf(2) = (SFmin - FS)/SFmin;
end

if K < Kmin
    pf(3) = 3*(Kmin - K)/Kmin;
elseif K > Kmax
    pf(3) = 3*(K - Kmax)/Kmax;
end

if am < 2*e
    pf(4) = (2*e - am)/(2*e);
end

if V_ad > area_rotor*b
    pf(5) = (V_ad - area_rotor*b)/(area_rotor*b);
end

if r > r_t
    pf(6) = (r - r_t)/r_t;
end

if Ew < (Fw + 2*e)
    pf(7) = (Fw + 2*e - Ew)/(Fw + 2*e);
end

PF = sum(pf);
if G <= 10
```

```

        PF = 2*PF;
elseif (G >= 10) && (G <= 25)
    PF = 5*PF;
elseif (G >= 25) && (G <= 40)
    PF = 10*PF;
else
    PF = 20*PF;
end

```

- Función Objetivo (+GA/*funcion\_objetivo.m*)

La función objetivo es la pieza central del algoritmo y es la que se encarga de asignar el nivel de adaptación a cada individuo evaluado en función de los diferentes criterios seleccionados, además de encargarse de la decodificación de los parámetros binarios y asignarles un valor real.

La función tiene como entradas la población existente en su representación real y binaria, la longitud de los parámetros binarios y el número de la generación actual.

```
function [x,val] = funcion_objetivo(x,L,G)
```

```
import GA.* SimT.* Adaptation.*
```

Determinar el sistema operativo o arquitectura empleada.

```
arch = cellstr(computer);
compare = strcmp(arch, 'GLNXA64');
```

Pre-Indexar las variables.

```

N          = size(x.real,1);

dis        = zeros(N,3);
pot        = zeros(N,1);
eta        = zeros(N,1);
M_total    = zeros(N,1);
PF         = zeros(N,1);
iPF        = zeros(N,1);

```

Decodificar los Parámetros Binarios

```
tabla_e    = [2.4;2.7;3;2.8;3.15;3.5;3.2;3.6;4;3.6;4.05;4.5]*1e-3;
```

```

tabla_b      = linspace(4,7,15)';
tabla_dens   = [2698;2795;2795;2810];
tabla_poisson = [0.33;0.33;0.33;0.33];
tabla_young  = [70;70;70;70]*1e9;
tabla_alpha  = [23.58;23.58;22.32;22.32]*1e-6;
tabla_conduc = [234;193;154;147];
tabla_allow  = [76;97;305;370]*1e6;

```

```

m          = (bin2dec(x.bin(:,1:L(1)))+1);
dis(:,1)   = tabla_e(bin2dec(x.bin(:,1:L(1)))+1);
dis(:,2)   =
dis(:,1).*tabla_b(bin2dec(x.bin(:,(sum(L(1:1))+1):(sum(L(1:2)))))+1)
dis(:,3)   = bin2dec(x.bin(:,(sum(L(1:2))+1):(sum(L(1:3)))))+1;

```

Asignar nombres representativos a las variables de proceso. Este paso no es absolutamente necesario pero facilita las labores de programación y la detección de errores en el código.

```

RPM        = 17000;

R          = x.real(:,1);
r          = x.real(:,2);
a          = x.real(:,3);
theta_s    = convang(x.real(:,4), 'deg', 'rad');
theta_i    = convang(x.real(:,5), 'deg', 'rad');
theta_o    = convang(x.real(:,6), 'deg', 'rad');
na         = round(x.real(:,7));
e          = dis(:,1);
b          = dis(:,2);
young      = tabla_young(dis(:,3));
poisson    = tabla_poisson(dis(:,3));
alpha      = tabla_alpha(dis(:,3));
densidad   = tabla_dens(dis(:,3));
conduc     = tabla_conduc(dis(:,3));
s_allow    = tabla_allow(dis(:,3));

```

Para cada individuo  $i$  de la población  $N$ , la función objetivo se encarga de evaluar las siguientes rutinas:

```

for i = 1:N
    disp(['Evaluando ' num2str(i) ' ° Candidato Solución...'])

```

· **Evaluar el Ciclo Termodinámico (+*SimT/SimT.m*). Ver A.3**

```

[pot(i), eta(i), P, T, Q, H, angle, imep, V_ad, r_t] = SimT(R(i), r(i), ...
    a(i), theta_s(i), theta_i(i), theta_o(i), e(i), b(i));

```

- Determinar el espesor de la carcasa y el número de pernos requerido para las tapas laterales. (+*SimT/epic.m* & +*SimT/ratiolong.m*).

```
[R_ext,R_BC,dn,N] = epic(R(i),e(i),imep);
ratio = ratiolong(R(i),R_ext,e(i),a(i),angle);
```

- Determinar el estado de esfuerzos y temperaturas en ANSYS. Ver A.2

```
CF = [P; T; Q; H; angle];
VD = [R(i),e(i),b(i),R_ext,R_BC,dn,N,length(P),ratio,young(i)...
      ,poisson(i),alpha(i),densidad(i),conduc(i)];
[area_t,T_ext,s_max] = thermoelastic_fem(CF,VD,compare);
FS = s_allow(i)/s_max;
```

- Determinar la longitud de las aletas (+*Adaptation/nageoires.m*).

```
[Lprime,area_fins,kappa] = nageoires(angle,T_ext,Q,R(i),...
e(i),b(i),conduc(i),na(i),R_ext,ratio);
```

- Efectuar la selección del rodamiento (+*Adaptation/peso\_rodamiento.m*).

```
[M_rodamiento,U,Ew,Fw] = peso_rodamiento(e(i));
```

- Determinar las dimensiones del engranaje interno (+*Adaptation/AGMA\_spur.m*).

```
[M_corona,t_engranaje] = AGMA_spur(pot(i),RPM,e(i),m(i));
```

- Determinar el peso del rotor (+*Adaptation/peso\_rotor*).

```
[M_rotor,am,area_rotor] = peso_rotor(R(i),e(i),a(i),b(i),...
U,Ew,densidad(i),V_ad);
```

- Determinar el peso del estator.

```
M_carcasa = (area_t + area_fins).*b(i).*densidad(i);
if (U/1000 + t_engranaje) > b(i)
    M_carcasa = (area_fins.*b(i) + area_t.*(U/1000 + ...
        t_engranaje)).*densidad(i);
end
```

- Evaluar las restricciones y la función de penalización (+*GA/penalty.m*).

```
PF(i) = penalty(pot(i),FS,R(i),e(i),b(i),am,V_ad,...
area_rotor,r(i),r_t,Fw,Ew,G);
```

- Evaluar la masa del conjunto Rotor-Estator.

```
M_total(i) = M_carcasa + M_rodamiento + M_corona + M_rotor;
```

```
if PF(i) ~= 0
```

```

        iPF(i) = 1;
    end
end

```

Por último se afectan las funciones objetivo por la función de penalización y almacenan los valores de la aptitud para ser devueltas al proceso iterativo.

```

f(:,1) = M_total.*(1+PF);

f(:,2) = -eta.*(1 - PF);
x.penalty = iPF;
x.adaptation = f;

```

· Sub-Rutinas empleadas por la Función Objetivo.

Determinar el espesor de la carcasa y el número de pernos requerido para las tapas laterales. (+*SimT/epic.m* & +*SimT/ratiolong.m*).

```

function [R_ext,R_BC,dn,N] = epic(R,e,Pmax)

import SimT.*

function [r_eq] = equivalent_radius(x,y)
    dx = diff(x);
    dy = diff(y);
    dl = sqrt(dx.^2 + dy.^2);
    L = sum(dl);
    r_eq = L/(2*pi);
end

dtheta = convang(3,'deg','rad');
theta = 0:dtheta:6*pi;
f_x = @(alpha,R) e*cos(alpha) + R*cos(alpha/3);
f_y = @(alpha,R) e*sin(alpha) + R*sin(alpha/3);
x = f_x(theta,R);
y = f_y(theta,R);

%%% Parámetros de la Empaquetadura
b0 = sqrt((3/16)/8)*25.4/1000; % m.
m = 2.0;
y = 13.7e6; % Pa

%%% Parámetros de los Pernos
dn = [2;2.5;3;3.5;4;5;6]*1e-3; % m.
A = [3.1416;4.9087;7.0686;9.6211;12.566;19.64;28.28]*1e-6; % m^2
B_m = [5.66;6.99;8.31;9.64;10.97;13.63;15.87]*1e-3; % m.
Rh = [4.27;5.08;5.88;6.68;7.50;9.11;10.32]*1e-3; % m.
DB = [3.31;3.91;4.50;5.10;5.69;6.88;7.95]*1e-3; % m.
sigma_res = 225e6; % Pa

```

```

espesor = [3/16;1/4;3/8;1/2;5/8;3/4;7/8;1]*25.4/1000; % m.
r_eq = equivalent_radius(x,y); % m.
G = 2*r_eq + (3/16)*25.4/1000; % m.

for j = 1:size(espesor,1)
    b = espesor(j);
    for i = 1:size(dn,1)
        W(1) = (pi/4)*(G^2)*Pmax + 2*b0*pi*G*m*Pmax; % N
        W(2) = pi*b0*y*G; % N
        A_min = max(W)/sigma_res; % m^2
        N = ceil((A_min/A(i))/4)*4; % u.
        C1 = 2*r_eq + 2*dn(i) + 2*Rh(i); % m.
        C2 = N*B_m(i)/pi; % m.
        BS = C1*pi/N; % m.
        BSmax = 2*dn(i) + (6*b)/(m + 0.5); % m.
        if BS <= BSmax
            while BS >= BSmax
                N = N + 4;
                BS = C1*pi/N;
            end
            if C1 >= C2
                if N <=12
                    break
                end
            end
        end
    end
end

R_ext = R*((C1/2)/r_eq)+1.2*DB(i); % m.
R_BC = R*((C1/2)/r_eq); % m.
end

function [ratio] = ratiolong(R_int,R_ext,e,a,tetha)

x = @(tetha,e,R,a) e*cos(tetha) + R*cos(tetha/3);
y = @(tetha,e,R,a) e*sin(tetha) + R*sin(tetha/3);
dx = diff(x(tetha,e,R_int,a));
dy = diff(y(tetha,e,R_int,a));
dl = sqrt(dx.^2 + dy.^2);
L_i = sum(dl);
dx2 = diff(x(tetha,e,R_ext,a));
dy2 = diff(y(tetha,e,R_ext,a));
dl2 = sqrt(dx2.^2 + dy2.^2);
L_e = sum(dl2);
ratio = L_e/L_i;

```

Determinar la longitud de las aletas (+*Adaptation/nageoires.m*).

```

function [Lprime,area_fins,kappa] =
nageoires(angle,T_ext,Q,R,e,b,conduc,na,R_ext,ratio)

```

```

function dl = fins(tetha,e,Re,b)
    alpha = SimT.convang(tetha,'deg','rad');
    x = @(tetha,e,R) e*cos(tetha) + R*cos(tetha/3);
    y = @(tetha,e,R) e*sin(tetha) + R*sin(tetha/3);
    dx = diff(x(alpha,e,Re));
    dy = diff(y(alpha,e,Re));
    dl = sqrt(dx.^2 + dy.^2)*b;
end

%%% Definir Constantes & Preindexar Variables
T_amb = 303;
h_c = 150;
x_e = 1.5e-3;
L = linspace(4*x_e,R,75);
Tb = zeros(1,na);
Q_b = zeros(1,na);
Lprime = zeros(1,na);
kappa = zeros(1,na+1);
angle2 = linspace(angle(1),angle(size(angle,1)),na*5);
T_ext2 = interp1(angle,T_ext,angle2);
Q2 = -interp1(angle,Q,angle2)/ratio;
for j = 1:na
    Tb(j) = mean(T_ext2((j-1)*5+1:(5*j)));
    Q_b(j) = mean(Q2((j-1)*5+1:(5*j)));
    kappa(j) = mean(angle2((j-1)*5+1:(5*j))) - 2.5*diff(angle2(1:2));
end
kappa(na+1) = kappa(na) + 5*diff(angle2(1:2));
ds = fins(kappa,e,R_ext,b);

%%% Determinar longitud de las Aletas
for j = 1:(na)
    vf = sqrt(1 + (ds(j)./(2*L)).^2);
    m = sqrt((2*h_c*vf)./(conduc*b*ds(j)));
    n_fin = m.*sqrt(L);
    X_prime = 2*n_fin.*sqrt(x_e);
    L_prime = 2*n_fin.*sqrt(L);
    if j ~= na
        C1 = -Q_b(j)./(conduc*n_fin*(0.5*(Tb(j)+Tb(j+1))-T_amb));
    else
        C1 = -Q_b(j)./(conduc*n_fin*(Tb(j)-T_amb));
    end
    C2 = besseli(1,L_prime);
    C3 = besselk(1,L_prime);
    C4 = besseli(0,L_prime);
    C5 = besselk(0,L_prime);
    C6 = (C2.*besselk(1,X_prime) + C3.*besseli(1,X_prime))
    ./(C4.*besselk(1,X_prime) + C5.*besseli(1,X_prime));
    deltaC = abs(C1 - C6); [error,pos] = min(deltaC);
    Lprime(j) = L(pos);
end

posnozero2 = find(Lprime ~= 0);
area = (ds/2).*(Lprime(posnozero2) - (x_e^2)./Lprime(posnozero2));
area_fins = sum(area);

```

```
Lprime = Lprime';
kappa = kappa';
end
```

Efectuar la selección del rodamiento (+*Adaptation/peso\_rodamiento.m*).

```
function [M,U,Ew,Fw,C,Co,Pu,Vel] = peso_rodamiento(e)

tabla_rodamiento(1,:) = [10,14,10,5.61,6.1,0.695,28000,0.0025];
tabla_rodamiento(2,:) = [10,14,13,5.83,6.3,0.72,28000,0.00460];
tabla_rodamiento(3,:) = [10,16,12,7.65,7.20,0.85,28000,0.0055];
tabla_rodamiento(4,:) = [12,17,13,9.13,10.40,1.22,26000,0.0049];
tabla_rodamiento(5,:) = [12,18,12,9.52,10,1.18,26000,0.0060];
tabla_rodamiento(6,:) = [14,20,12,9.90,10.60,1.25,24000,0.0085];
tabla_rodamiento(7,:) = [15,21,15,13.80,16.30,2,24000,0.0110];
tabla_rodamiento(8,:) = [15,21,21,18.70,24.50,3,24000,0.0170];
tabla_rodamiento(9,:) = [16,24,20,20.50,23.60,2.90,22000,0.0220];
tabla_rodamiento(10,:) = [18,28,16,18.70,18.60,2.28,19000,0.0240];
tabla_rodamiento(11,:) = [20,28,20,22.90,28.50,3.45,18000,0.0270];
tabla_rodamiento(12,:) = [20,28,25,29.20,39,4.90,18000,0.0320];
tabla_rodamiento(13,:) = [20,30,30,34.10,41.50,5.20,17000,0.0490];
tabla_rodamiento(14,:) = [23,35,16,24.20,23.20,2.90,15000,0.0290];
tabla_rodamiento(15,:) = [25,35,30,44.60,62,7.80,15000,0.0650];

i = 1;
while (tabla_rodamiento(i,1) <= 5*e*1e3) || (tabla_rodamiento(i,2) <= 7*e*1e3)
    i = i + 1;
end

Fw = tabla_rodamiento(i,1); Ew = tabla_rodamiento(i,2);
U = tabla_rodamiento(i,3); C = tabla_rodamiento(i,4);
Co = tabla_rodamiento(i,5); Pu = tabla_rodamiento(i,6);
Vel = tabla_rodamiento(i,7); M = tabla_rodamiento(i,8);
```

Determinar las dimensiones del engranaje interno (+*Adaptation/AGMA\_spur.m*).

```
function [M_e,EspeSor] = AGMA_spur(Pot,RPM,e,m)

if ceil(m/3) == 1
    P = 25.4/0.6; % in^-1
elseif ceil(m/3) == 2
    P = 25.4/0.7; % in^-1
elseif ceil(m/3) == 3
    P = 25.4/0.8; % in^-1
else
    P = 25.4/0.9; % in^-1
end

if ((m/3) - floor(m/3))*3 + 1 == 1
    J = 0.39;
```

```

elseif ((m/3) - floor(m/3))*3 + 1 == 2
    J = 0.35;
else
    J = 0.37;
end

Td = 5252*Pot/RPM; % ft-lbf
Wt = Td/(3*e/304.8); % lbf
V = (6*e/304.8)*pi*RPM; % ft/min
Qv = 11;
B = 0.25*(12 - Qv)^(2/3);
A = 50 + 56*(1-B);
Kv = ((A + sqrt(V))/A)^B;
Ko = 1.3;
Km = 1.3;
Ks = 1;
K = [Kv Ko Km Ks];

%%% Esfuerzos Admisibles
SF = 1.0;
St = 260000; % psi
sigma_adm = St/SF; % psi
% Esfuerzo a Flexión
F = Wt*P*prod(K)/(sigma_adm*J);
% Aproximación a la Masa del Engranaje
M_e = ((9*e)^2 - (6*e - (7/6)*(25.4/1000/P))^2) ... % kg.
    *pi/4*(F*25.4/1000)*7850;
Espesor = (F*25.4/1000); % m.

```

Determinar el peso del rotor (+Adaptation/peso\_rotor).

```

function [M_rotor,am,AREA] = peso_rotor(R,e,a,b,U,Ew,dens,V_ad)

a_prime = a - 0.05e-3;
x = @(v,R,e) R*cos(2*v) - ((3*e^2)/R)*sin(6*v).*cos(2*v) + ...
    2*e*cos(2*v).*cos(3*v).*sqrt(1 - ((3*e/R)*sin(3*v)).^2);
y = @(v,R,e) R*sin(2*v) - ((3*e^2)/R)*sin(6*v).*cos(2*v) + ...
    2*e*sin(2*v).*cos(3*v).*sqrt(1 - ((3*e/R)*sin(3*v)).^2);
X = @(v,R,e,a1) x(v,R,e) + ((3*e*a1)/(2*R))*(cos(5*v) - ...
    cos(v)) + a1*cos(2*v).*sqrt(1 - ((3*e/R)*sin(3*v)).^2);
Y = @(v,R,e,a1) y(v,R,e) + ((3*e*a1)/(2*R))*(sin(5*v) - ...
    sin(v)) + a1*sin(2*v).*sqrt(1 - ((3*e/R)*sin(3*v)).^2);
v = linspace(pi/6,pi/2);
X1 = X(v,R,e,a_prime);
Y1 = Y(v,R,e,a_prime);
AREA = polyarea([0 X1 0],[0 Y1 0]);

if U*1e-3 > b
    Vol = (pi/4)*(e*(9*e)^2 + (b-e)*(Ew*1e-3)^2 - (U*1e-3 - (b-e))*((9*e)^2 -
    (Ew*1e-3)^2));
else

```

```
    Vol = (pi/4)*(e*(9*e)^2 + (b-e)*(Ew*1e-3)^2);  
end  
  
M_rotor = (3*(AREA*b - V_ad) - Vol)*dens;  
v_prime = pi/3;  
X2 = X(v_prime,R,e,a_prime);  
Y2 = Y(v_prime,R,e,a_prime);  
am = sqrt(X2^2 + Y2^2);
```

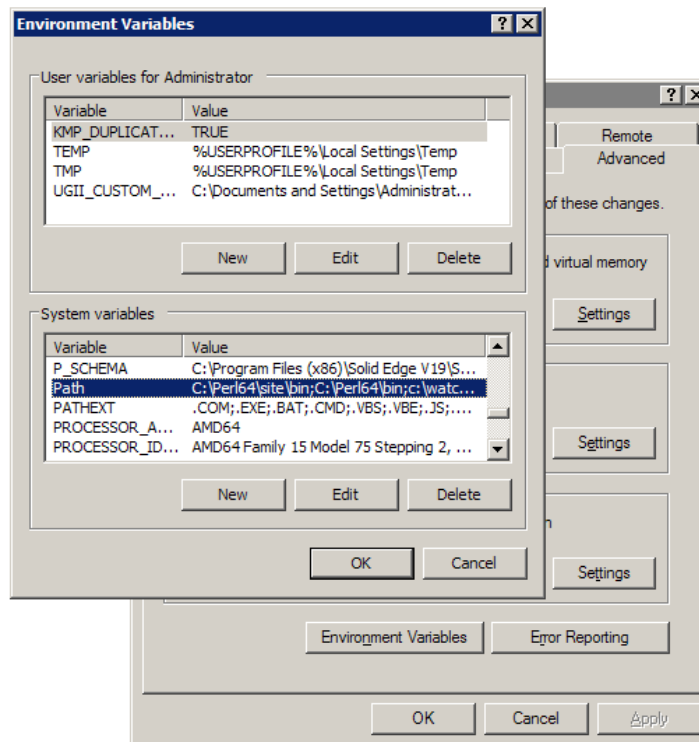
## A.2. IMPLEMENTACIÓN EN ANSYS-BATCH (*froid.inp*)

Se optó por efectuar un análisis termoelástico bidimensional que permitiese observar los efectos combinados de la transferencia de calor con la presión de los gases al interior del recipiente. Para esto empleamos un elemento Plane223, que tiene desplazamientos en las direcciones **x** y **y** (UX, UY) y la temperatura en el nodo (TEMP) como grados de libertad, en un enmallado que tiene en promedio unos 4200 nodos, luego tendríamos alrededor de 12600 DOF's en esta prueba.

Para ejecutar ANSYS en modo batch, primero debemos añadir el directorio que contiene el ejecutable ANSYS110.exe al path o rutas de búsqueda de comandos (*command search path*) de modo que podamos ejecutar la rutina sin importar la ubicación relativa de los archivos.

Para Windows XP y Windows NT, esto se hace haciendo un cambio en las variables del entorno como muestra la figura 58.

Figura 58. Modificar variables del sistema



Seleccionamos la variable Path y damos clic en editar. Una vez ahí añadimos el directorio que contiene el ejecutable de ANSYS – por ejemplo, *C:\Program Files\ANSYS Inc.lv110\ANSYS\bin\winx64\* - al final del texto separado por un semicolon o punto y coma (;). Damos clic en Aceptar y procedemos a reiniciar el PC.

La rutina se ejecuta desde MATLAB empleando el comando (!) que permite ejecutar comandos del símbolo del sistema desde una función o rutina .m.

El comando para ejecutar ANSYS en modo batch desde el símbolo del sistema en este caso sería:

```
> ansys110.exe -p ANE3FL -dir "X:\dir1\" -s read -np 2 -b -i input.file -o output.file
```

Esto a grandes rasgos quiere decir que se ejecute ANSYS con la licencia especificada (-p ANE3FL), en el directorio indicado (dir "X:\dir1\ "), que emplee procesamiento paralelo con memoria compartida SMP (-np 2). El archivo de entrada de parámetros, es decir donde se encuentra definida la prueba en lenguaje APDL se llama input.file (-i input.file) y la captura de pantalla de la prueba, es decir los datos de salida sean almacenados en el archivo output.file (-o output.file).

En el procesamiento paralelo en sistemas con más de un procesador lógico, se establece la comunicación entre dichos procesadores para coordinar la solución de un problema. El procesamiento paralelo con memoria compartida **SMP**, permite la comunicación inter-procesador a través de variables almacenadas en un espacio compartido. Otra forma de resolver estos problemas emplea un modulo de memoria para cada uno de los procesadores y estos se conectan entre sí mediante una red de alta velocidad. Esta técnica se conoce como computación multi-procesador con memoria distribuida y emplea ficheros **MPI** para la comunicación de los equipos, y permite

Para ejecutar *ANSYS Distributed* se requiere de un protocolo MPI funcional, que en nuestro caso consiste en la implementación de código abierto del estándar MPI

realizada en la Universidad de Chicago MPICH2 (Argonne National Laboratory Group, 2002) y que puede ser descargada de manera gratuita en la pagina del proyecto el laboratorio nacional de Argonne (<http://www.mcs.anl.gov/research/projects/mpich2/index.php>).

Para mayor información se recomienda consultar la documentación incluida en el paquete de ANSYS (ANSYS, Inc., 2005).

El comando para ejecutar *ANSYS Distributed* en un equipo local en modo de ejecución por lotes desde el símbolo del sistema en este caso sería:

```
> ansys110.exe -p ANE3FL -dir "X:\dir1\" -s read -dis -mpi MPICH -np 2 -b -i input.file -o output.file
```

Para nuestro caso en particular, tendríamos el archivo se ejecuta a partir del directorio X:\WNKL\+ANSYS, donde X es la letra de la unidad de disco a partir de la cual ejecutamos el programa. La línea de código ejecutada por MATLAB es la siguiente:

```
! ansys110.exe -p ANE3FL -dir "X:\WNKL\+ANSYS" -s read -dis -mpi MPICH -np 2 -b -i "+ANSYS\froid.inp" -o "+ANSYS\WNKL.out"
```

- Implementación en Linux (GLNX86 & GLNXA64).

La implementación en Linux se plantea con el objeto de aprovechar arreglos en cluster, que se caracterizan por el alto rendimiento, alta disponibilidad y escalabilidad en aplicaciones de computación en paralelo, como son en este caso los algoritmos genéticos. Para este proyecto queda planteada la utilización de procesamiento paralelo distribuido en un computador local, y se propone como proyecto futuro su implementación en un arreglo en cluster. Para lo cual se requieren instalaciones funcionales de ANSYS y de MATLAB para Linux.

La ejecución de comandos UNIX/LINUX de MATLAB emplea rutas diferentes a las del sistema operativo, lo que necesariamente nos obliga a referenciar la dirección absoluta de la instalación de ANSYS en el código.

Para ejecutar el análisis en modo batch desde el Shell de Linux empleamos el siguiente comando:

```
> /ansys110 -p ANE3FL -dir "/dir1/" -s read -dis -np 2 -b < "/input.file" >
"/output.file" &
```

Los comandos tienen concordancia con los explicados en la sección anterior.

La línea de código ejecutada por MATLAB sería entonces:

```
! sudo nice -20 /ansys_inc/v110/ansys/bin/ansys110 -p ANE3FL -dir
"/WNKL/+ANSYS" -j "WNKL" -np 2 -b < "/WNKL/+ANSYS/froid.inp" >
"/WNKL/+ANSYS/WNKL.out" &
```

Donde *sudo* se emplea para otorgar privilegios de lectura y escritura – en caso de ser necesarios – y el comando *nice* sirve para asignar un valor de prioridad al proceso de mayor (-20) a menor (+20).

- Ejecución del Análisis Termoelástico (+*Adapation/thermoelastic\_fem.m*).

Se encarga de la selección de la arquitectura, la ejecución del análisis por elementos finitos y de devolver la temperatura externa y el esfuerzo máximo a MATLAB.

```
function [area_t,T_ext,s_max] = thermoelastic_fem(CF,VD,compare)
```

```
if compare == 0
    fid = fopen('+ANSYS\designvar.dat','wt');
    fprintf(fid, '%20.6f \n',VD); fclose('all');
    fid = fopen('+ANSYS\condition.dat','wt');
    fprintf(fid, '%20.6f \n',CF); fclose('all');
    ! ansys110.exe -p ANE3FL -dir "C:\WNKL\+ANSYS" -j "WNKL" -s read -dis -mpi
    MPICH -np 2 -b -i "+ANSYS\froid.inp" -o "+ANSYS\WNKL.out"

    fid = fopen('\+ANSYS\results.dat', 'r');
    A = fscanf(fid,'%g', [1 inf]);
else
    fid = fopen('+ANSYS/designvar.dat','wt');
    fprintf(fid, '%20.6f \n',VD); fclose('all');
    fid = fopen('+ANSYS/condition.dat','wt');
    fprintf(fid, '%20.6f \n',CF); fclose('all');
    ! sudo nice -20 /ansys_inc/v110/ansys/bin/ansys110 -p ANE3FL -dir
    "/WNKL/+ANSYS" -j "WNKL" -np 2 -b < "/WNKL/+ANSYS/froid.inp" >
    "/WNKL/+ANSYS/WNKL.out" 2>&1

    fid = fopen('/+ANSYS/results.dat', 'r');
```

```
A = fscanf(fid, '%g', [1 inf])';  
end
```

```
area_t = A(1); s_max = A(2);  
T_ext = A(3:size(A,1));
```

Existen aproximadamente unos 15000 comandos en ANSYS, cada uno con una función y una sintaxis específica. Sin embargo, para resolver un problema típico solo se emplean unos cuantos comandos muy comunes, como senos, cosenos, comparaciones lógicas y bucles.

El archivo *froid.inp* puede dividirse en secciones según la función y naturaleza de sus comandos como se muestra a continuación:

- Lectura y obtención de parámetros.
- Generación de la geometría.
- Enmallado.
- Restricciones y condiciones de frontera.
- Solución, Post-proceso y almacenamiento de resultados.
- Lectura y Obtención de Parámetros

Los parámetros podemos clasificarlos en 14 parámetros escalares, contenidos en el archivo *designvar.dat*, y 5 parámetros vectoriales almacenados en el archivo *condition.dat*. Para efectuar la lectura y almacenar los datos en una variable interna – VD para los escalares y CF para los vectoriales - empleamos el comando \*VREAD.

```
*set,VD  
  
*dim,VD,array,14  
*create,ansuitmp  
*vread,VD,'designvar','dat','',ijk,14,1,0  
(E20.6)  
*end  
/input,ansuitmp  
  
*SET,CF  
*DIM,CF,ARRAY,5*long  
*CREATE,ansuitmp
```

```

*VREAD,CF,'condition','dat','','IJK,5*long,1,0
(E16.3)
*END
/INPUT,ansuitmp

```

Podemos descomponer estas variables en otras más pequeñas. Si bien no es absolutamente necesario, nos facilita el trabajo asignar nombres a los parámetros que podamos asociar con facilidad.

```

Ri          = VD(1)      ! Radio Generador
e           = VD(2)      ! Excentricidad del Rotor
b           = VD(3)      ! Ancho de la Carcasa
Re          = VD(4)      ! Radio Externo
Rb          = VD(5)      ! Radio del Círculo de Pernos
d           = VD(6)      ! Diámetro de los Pernos
N           = VD(7)      ! Numero de Pernos
long        = VD(8)      ! Numero de Datos
ratio       = VD(9)      ! Relación de Áreas
young       = VD(10)     ! Modulo de Elasticidad
poisson     = VD(11)     ! Modulo de Poisson
alpha       = VD(12)     ! Coeficiente de Expansión Térmica
densidad    = VD(13)     ! Densidad
conduc      = VD(14)     ! Conductividad Térmica
*set,P
*set,T
*set,H
*set,Q
*set,ANG
*dim,P,array,long
*dim,T,array,long
*dim,H,array,long
*dim,Q,array,long
*dim,ANG,array,long
*do,i,1,long
  P(i) = CF(i)           ! Presión de los Gases
  T(i) = CF(i+long)     ! Temperatura de los Gases
  Q(i) = CF(i+2*long)   ! Flujo de Calor por Unidad de Área
  H(i) = CF(i+3*long)   ! Coeficiente de Transf. por Convección
  ANG(i) = CF(i+4*long) ! Ángulo de Referencia
*enddo

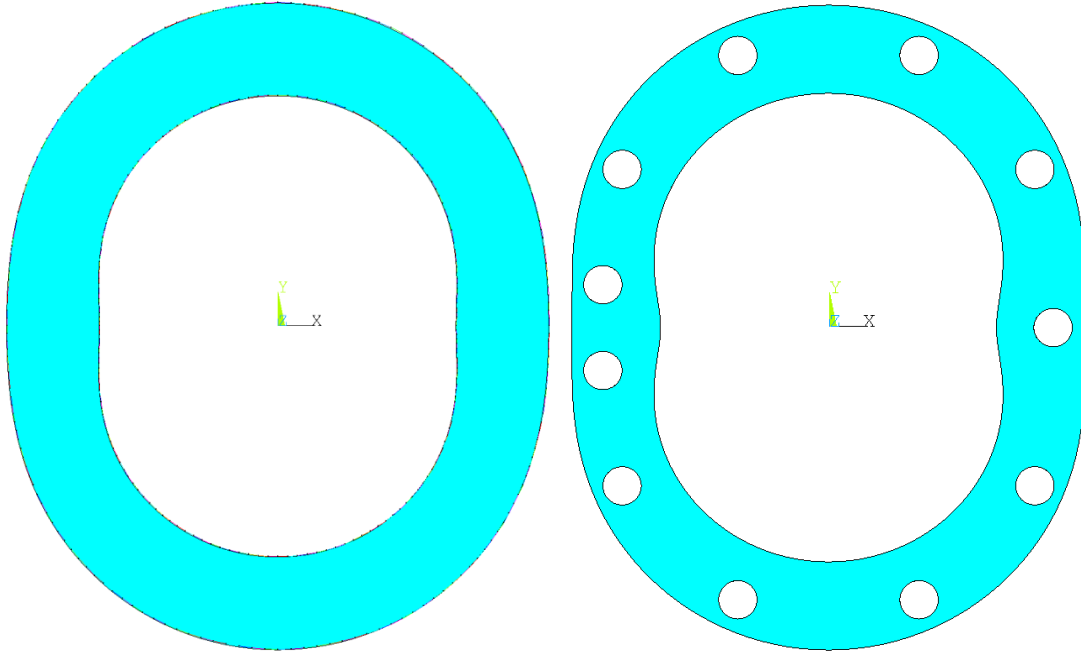
```

#### · Generación de la Geometría.

La generación de la geometría, el enmallado y la aplicación de las cargas se efectúa en la fase de pre-procesamiento, luego es preciso emplear el comando /PREP7 antes de empezar a trabajar.

La carcasa se genera a partir de puntos o KP, cada cierto número de grados a partir de los cuales se generan unas áreas trapezoidales entre la epitrocoide interna y la superficie externa de la carcasa, a partir de las ecuaciones paramétrica obtenidas en (Yamamoto, 1981) y procedemos a ubicar las perforaciones de los pernos, como se muestra en la figura 59.

**Figura 59.** Generación de la Geometría



```

/prep7
*afun,deg
inc      = 6          ! Incremento cada 6° de Cigüeñal
nDIV     = 1080/inc   ! Numero de Divisiones
angle    = ANG(1)

/com, GENERAR LA GEOMETRIA
*do,j,1,180
  Yi = e*cos(angle) + Ri*cos(angle/3)
  Xi = e*sin(angle) + Ri*sin(angle/3)
  Ye = e*cos(angle) + Re*cos(angle/3)
  Xe = e*sin(angle) + Re*sin(angle/3)
  k,j,Xi,Yi
  k,j+180,Xe,Ye
  *if,j,ne,1,then
    l,j-1,j
    l,j+179,j+180
  *endif
  angle = angle + inc
*enddo

```

```

1,180,1
1,360,181
al,all

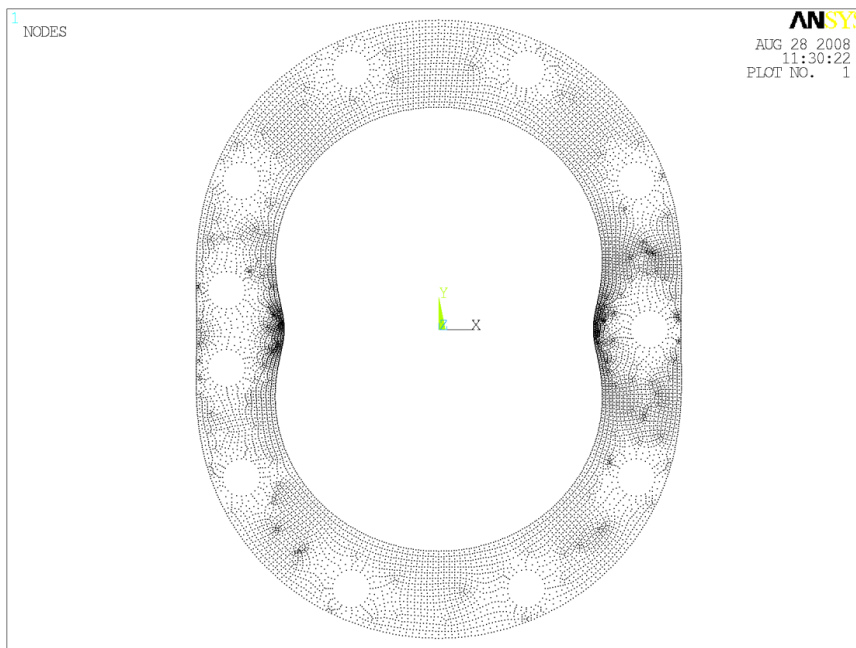
/com, Ubicar las Perforaciones de los Pernos
numstr,line,1000
inc2 = (1080/N)
angle2 = 270 - inc2*3/2
*do,iJ,1,N-2
  Xp = e*sin(angle2) + Rb*sin(angle2/3)
  Yp = e*cos(angle2) + Rb*cos(angle2/3)
  cyl4,Xp,Yp,d
  angle2 = angle2 - inc2
*enddo
cyl4,Rb-e,0,d
asba,1,all,,delete,delete
numcmp,area

```

- Enmallado.

Antes de efectuar el enmallado es preciso definir el tipo de elemento y los grados de libertad del mismo, y definir las propiedades mecánicas y térmicas que caracterizan al material en cuestión. Se emplean elementos Plane223 Quad con 8 nodos cada uno y hasta 4 grados de libertad por nodo y el enmallado por barrido como se observa en la Figura 60.

**Figura 60.** Enmallado por Barrido



```

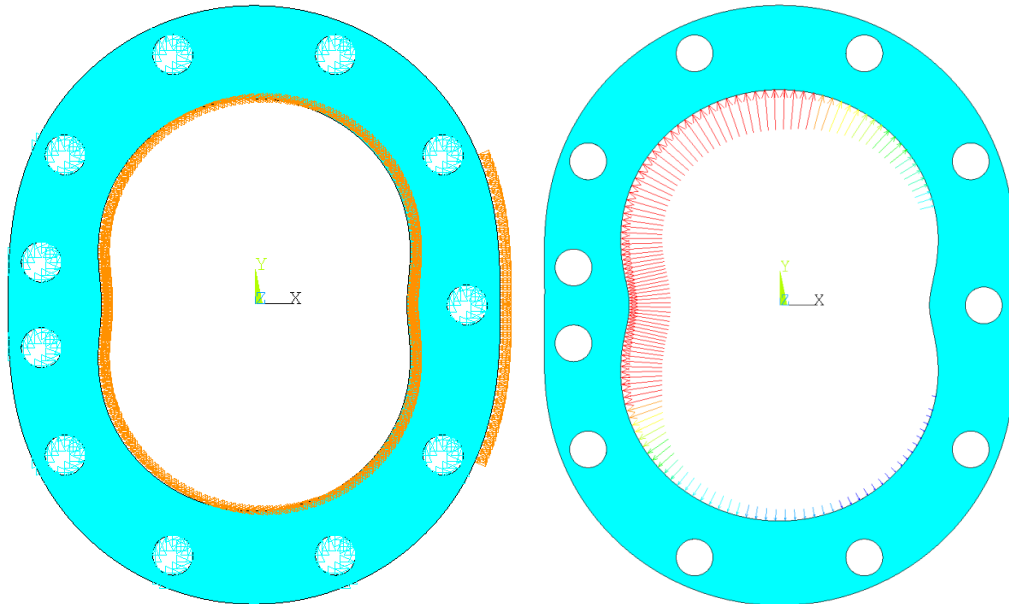
/com, PROPIEDADES DEL ELEMENTO
et,1,223,11
mp,ex,1,young
mp,nuxy,1,poisson
mp,dens,1,densidad
mp,alpx,1,alpha
mp,kxx,1,conduc

/com, ENMALLADO
cm,_y,area
asel,,,,1
cm,_y1,area
chkmsch,'area'
cmsel,s,_y
!*
mshkey,2
amesh,_y1
mshkey,0
!*
cmdele,_y
cmdele,_y1
cmdele,_y2
!*

```

· Restricciones y Condiciones de Frontera.

**Figura 61.** Restricciones en los pernos y Presión al interior de la carcasa



```

/com, APLICAR CONDICIONES DE FRONTERA
tref,320

```

```

/com, Restricción en los Pernos
LFIN = (N-1)*4+999

```

```

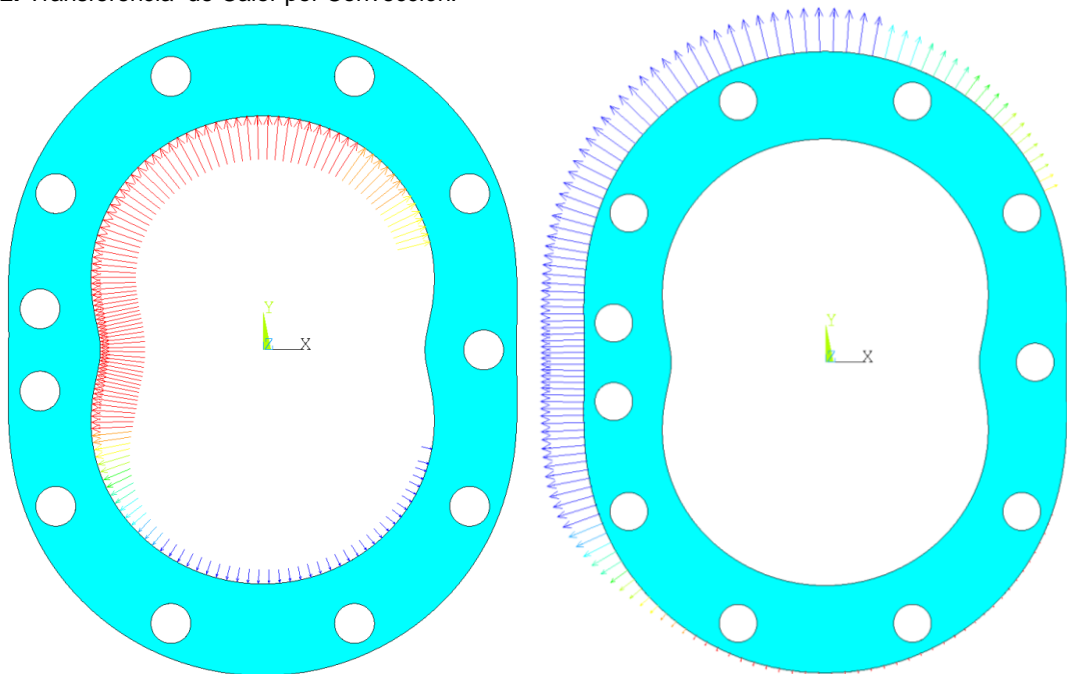
lsel,s,,,1000,LFIN
dl,all,,uy,0
dl,all,,ux,0
dl,all,,temp,400,0
lsel,all

/com, Temperaturas Constantes
ksel,s,,,1,180
dk,all,temp,453.5
ksel,all

ksel,s,,,long+180,360
dk,all,temp,325
ksel,all

```

**Figura 62.** Transferencia de Calor por Convección.



```

/com, Presion Interna y Conveccion Interna
*do,i,1,long
  lsel,s,,, (i*2)-1
  sfl,all,pres,P(i)
  sfl,all,conv,H(i),,T(i)
*enddo

```

Es preciso tener en cuenta que parte del calor es evacuado hacia el rotor y por las tapas laterales, luego consideraremos que solo un porcentaje (60%) escapa por a través de la carcasa.

```

/com, Flujo de Calor en la base de las aletas

```

```

*do,i,1,long
  lsel,s,,, (i*2)
  sfl,all,hflux,-1*Q(i)/ratio
*enddo
lsel,all

```

· Solución, Post-Proceso y Resultados.

Los resultados del análisis que sirven como datos de entrada para el diseño de las aletas y para la evaluación de la función peso serian: el área de la sección transversal, el esfuerzo máximo equivalente por efectos combinados y el vector de temperaturas en el exterior de la carcasa, es decir en la base de las aletas.

Estos datos se almacenan en el archivo results.dat que será leído posteriormente por MATLAB como texto ASCII.

```

/sol
antype,0
/status,solu
solve
/post1

/com, Determinar Área de la Sección Transversal
*get,A_seccion,area,1,area

/com, Obtener Esfuerzo Equivalente Máximo
nsort,s,eqv,0,0,1,select
*get,nmax,node,0,num,max
*get,smax,node,nmax,s,eqv
nset,all
*set,RES
*dim,RES,array,long+2,1
RES(1,1) = A_seccion
RES(2,1) = smax
*do,i,1,long
  X0 = KX(i+180)
  Y0 = KY(i+180)
  Z0 = KZ(i+180)
  NUMERO = NODE(X0,Y0,Z0)
  RES(i+2,1) = TEMP(NUMERO)
*enddo
*cfopen,results,dat
*vwrite,res(1,1,1)
(f20.6)
*cfclos

```

El archivo *froid\_ver.inp* en general posee la misma estructura, pero hace lecturas a dos archivos adicionales que contienen la información geométrica de las aletas.

- Lectura y Obtención de Parámetros

Se hacen lecturas adicionales a los archivos *designvar\_ver.dat* y *condition\_ver.dat*, que contienen el número de puntos para la construcción de las aletas y las coordenadas de los mismos, respectivamente.

```
*set,long2
*dim,long2,array,1
*create,ansuitmp
*vread,long2,'designvar_ver','dat','','ijk,1,1,0
(E20.6)
*end
/input,ansuitmp
long2a = long2(1)
*set,CXY
*dim,CXY,array,2*long2a
*create,ansuitmp
*vread,CXY,'condition_ver','dat','','IJK,2*long2a,1,0
(E20.6)
*end
/input,ansuitmp
*set,XX
*set,YY
*dim,XX,array,long2a
*dim,YY,array,long2a
```

- Generación de la Geometría.

La geometría cambia sustancialmente para incluir las aletas como fueron estimadas por la función *nageoires.m*, como se muestra en la Figura 63.

```
/prep7

*afun,deg
inc      = 6           ! Incremento cada 6° de Cigüeñal,
nDIV     = 1080/inc    ! Numero de Divisiones
angle    = ANG(1)
/com, GENERAR LA GEOMETRIA
*do,j,1,180
  Yi = e*cos(angle) + Ri*cos(angle/3)
  Xi = e*sin(angle) + Ri*sin(angle/3)
  Ye = e*cos(angle) + Re*cos(angle/3)
  Xe = e*sin(angle) + Re*sin(angle/3)
  k,j,Xi,Yi
  k,j+180,Xe,Ye
  *if,j,ne,1,then
    l,j-1,j
    *if,j,gt,long+1,then
      l,j+179,j+180
    *endif
  *endif
```

```

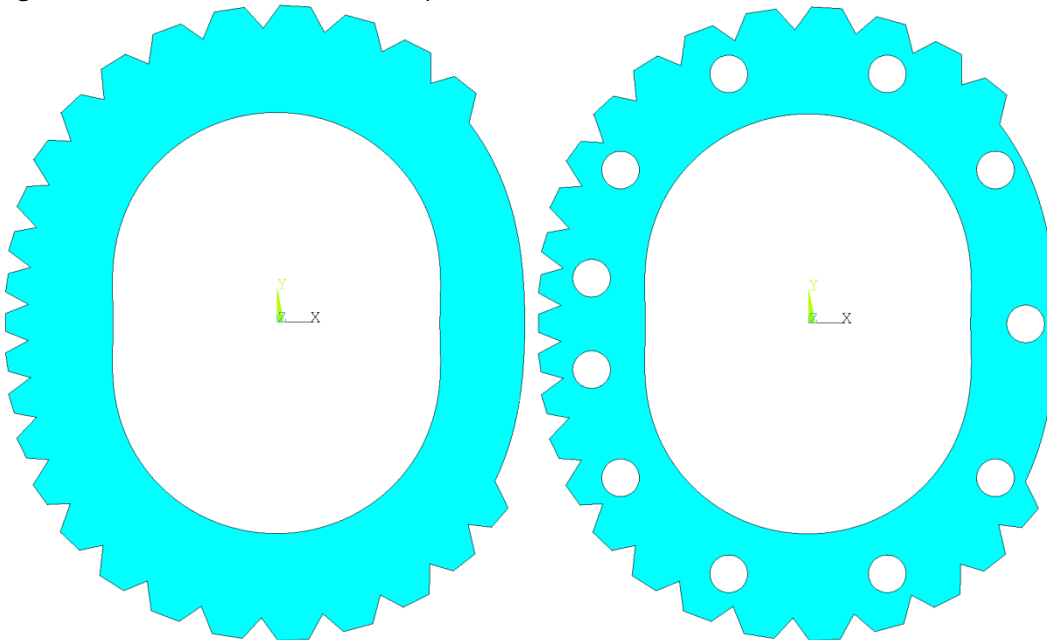
    angle = angle + inc
*enddo
1,180,1

*do,k,1,long2a
    XX(k) = CXY(k)
    YY(k) = CXY(k+long2a)
    k,k+1000,YY(k),XX(k)
*enddo
numstr,line,10000
na = (long2a-1)/3
*do,k,1,na
    1,998+k*3,999+k*3
    1,999+k*3,1000+k*3
    1,1000+k*3,1001+k*3
*enddo
1,360,1001
1,(long2a+1000),(long+181)
al,all

/com,Ubicar las Perforaciones de los Pernos
numstr,line,1000
inc2 = (1080/N)
angle2 = 270 - inc2*3/2
*do,iJ,1,N-2
    cyl4,e*sin(angle2) + Rb*sin(angle2/3),e*cos(angle2) + Rb*cos(angle2/3),d
    angle2 = angle2 - inc2
*enddo
cyl4,Rb-e,0,d
asba,1,all,,delete,delete
numcmp,area

```

**Figura 63.** Generación de la Geometría para el modelo de verificación



- Enmallado

El enmallado se realiza igual que en el modelo preliminar y luego se procede a hacer un refinamiento del enmallado en las proximidades de las aletas.

```

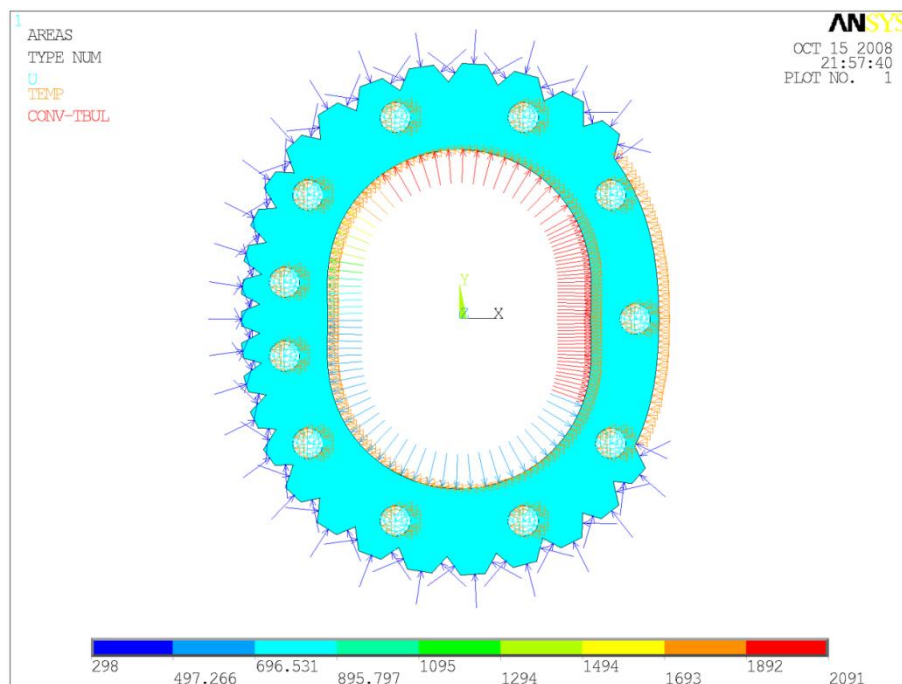
cm, _y, line
lsel, s, , , 10000, 20000
cm, _y1, line
cmsel, s, _y
cmdele, _y
!*
!*
lrefine, _y1, , , 2, 1, 1, 1
cmdele, _y1
!*

```

- Restricciones y Condiciones de Frontera.

Se omite el flujo de calor en la pared externa y se reemplaza por la transferencia por convección con el aire ambiental, como se puede observar en la Figura 64.

**Figura 64.** Condiciones de frontera para el modelo de verificación.



```

/com, Conveccion Externa
lsel, s, , , 10000, 20000
sfl, all, conv, 160, , 298
lsel, all

```

### A.3. SIMULACIÓN DEL CICLO EN MATLAB (*SimT.m*)

Las funciones de MATLAB aquí descritas se encuentran basadas en buena medida en los programas de FORTRAN para motores recíprocos presentados por (Ferguson, 1986) y en la adaptación de dichas rutinas a MATLAB tomada de (Buttsworth, 2002). Las funciones en general conservan la estructura de los archivos originales.

- Información Termodinámica Básica (*airdata.m* & *fueldata.m*)

Como se menciona en la sección 2.4.2, para el cálculo de propiedades del aire y los productos de la combustión son empleadas las ecuaciones empíricas propuestas por (Gordon & McBride, 1971), mientras que para el combustible empleamos las ecuaciones propuestas por (Heywood, 1988).

Para el aire y los productos de la combustión empleamos la rutina *airdata.m* para obtener los coeficientes  $a_1$  al  $a_7$  para la Ecuación 13 en cada las siguientes especies químicas y en el orden mostrado:  $CO_2$ ,  $H_2O$ ,  $N_2$ ,  $O_2$ ,  $CO$ ,  $H_2$ ,  $H$ ,  $O$ ,  $OH$  y  $NO$ .

Como conocemos de antemano el tipo de combustible a utilizar – metanol es para todos los efectos prácticos el combustible más común en los aeromodelos -, limitamos la función *fueldata.m* a suministrar los coeficientes  $a_1$  al  $a_7$  para la Ecuación 14 y los coeficientes  $\alpha$ ,  $\beta$ ,  $\gamma$  y  $\delta$ , requeridos para obtener la relación estequiométrica.

```
function [alpha,beta,gamma,delta,Afuel] = fueldata()

alpha    = 2;
beta     = 6;
delta    = 0;
gamma    = 1;
Afuel    = [3.5176, 0.0200, -6.0015e-6, 0, 0, -3.0301e4, 8.0162];
```

- Fuel, Air & Residual Gases (*farg.m*)

La función FARG se encarga de determinar la fracción molar y las propiedades termodinámicas de la mezcla de aire, combustible y residuos de la combustión al interior de la cámara.

```
function [h,u,v,s,Y,cp,dlv1T,dlv1p] = farg(p,T,phi,f)
```

## Obtener Coeficientes $a_1$ a $a_7$

```
import SimT.airdata SimT.fueldata

A = airdata('GMcB_low');
[alpha,beta,gamma,delta,Afuel] = fueldata();
Ru      = 8314.34;                               % J/(kmolK)
table   = [-1 1 0 0 1 -1]';
M       = [44.01 18.02 28.008 32.000 28.01 2.018]'; % kg/kmol

MinMol  = 1e-25;
dlv1T   = 1;
dlvlp   = -1;
eps     = 0.210/(alpha+0.25*beta-0.5*gamma);
```

## Determinar el número de moles de cada especie producto de la reacción.

```
if phi <= 1.0                                     % Mezcla Pobre o Estequiometrica
    nu = [alpha*phi*eps beta*phi*eps/2 (0.79 + delta*phi*eps/2) ...
          0.21*(1 - phi) 0 0]';
    dcdT= 0;
else                                              % Mezcla Rica en Combustible
    z = 1000/T;
    K = exp(2.743 + z*(-1.761 + z*(-1.611 + z*0.2803)));
    dKdT= -K*(-1.761 + z*(-3.222 + z*0.8409))/1000;
    a = 1 - K;
    b = 0.42 - phi*eps*(2*alpha-gamma) + K*(0.42*(phi-1) + alpha*phi*eps);
    c = -0.42*alpha*phi*eps*(phi-1)*K;
    nu5 = (-b + sqrt(b^2 - 4*a*c))/2/a;
    dcdT = dKdT*(nu5^2 - nu5*(0.42*(phi - 1)+alpha*phi*eps)+ ...
            0.42*alpha*phi*eps*(phi-1))/(2*nu5*a + b);
    nu = [(alpha*phi*eps - nu5) (0.42 - phi*eps*(2*alpha-gamma) + nu5) ...
           (0.79 + delta*phi*eps/2) 0 nu5 (0.42*(phi-1) - nu5)]';
end
```

## Fracción Molar de los Productos de la Combustión

```
tmoles = sum(nu);                               % Mol
Y       = nu/tmoles;                             % 1/1
Mres    = sum(Y.*M);                             % kg/kMol
```

## Fracción Molar y Peso Molecular de la Mezcla Aire-Combustible

```
fuel    = eps*phi/(1 + eps*phi);
o2      = 0.21/(1 + eps*phi);
n2      = 0.79/(1 + eps*phi);
Mfa     = fuel*(12.01*alpha + 1.008*beta + 16*gamma + 14.01*delta)
        ... + 32*o2 + 28.02*n2;                 % kg/kMol
Mfuel   = 12.01*alpha + 1.008*beta + 16.000*gamma + 14.01*delta;
```

## Fracción Molar del Combustible-Aire-Gas Residual

```
Yres = f/(f + Mres/Mfa*(1 - f));  
Y = Y*Yres;  
Yfuel = fuel*(1 - Yres);  
Y(3) = Y(3) + n2*(1 - Yres);  
Y(4) = Y(4) + o2*(1 - Yres);
```

## Propiedades de cada una de las especies

```
Tcp0 = [1 T T^2 T^3 T^4]';  
Th0 = [1 T/2 T^2/3 T^3/4 T^4/5 1/T]';  
Ts0 = [log(T) T T^2/2 T^3/3 T^4/4 1]';  
cp0 = A(1:6,1:5)*Tcp0;  
h0 = A(1:6,1:6)*Th0;  
s0 = A(1:6,[1:5 7])*Ts0;  
cpfuel = Afuel(1:5)*[1 T T^2 T^3 1/T^2]';  
hfuel = Afuel(1:6)*[1 T/2 T^2/3 T^3/4 -1/T^2 1/T]';  
s0fuel = Afuel([1:5 7])*[log(T) T T^2/2 T^3/3 -1/T^2/2 1]';  
  
Yfuel = max(Yfuel,MinMol);  
i = find(Y < MinMol);  
Y(i) = ones(length(i),1)*MinMol;
```

## Propiedades de la mezcla

```
h = hfuel*Yfuel + sum(h0.*Y);  
s = (s0fuel - log(Yfuel))*Yfuel + sum((s0 - log(Y)).*Y);  
cp = cpfuel*Yfuel + sum(cp0.*Y) + sum(h0.*table*T*dcdT*Yres/tmoles);  
MW = Mfuel*Yfuel + sum(Y.*M);  
R = Ru/MW;  
h = R*T*h;  
u = h-R*T;  
v = R*T/p;  
s = R*(-log(p/101.325e3) + s);  
cp = R*cp;
```

### · Productos de la Combustión (*ecp.m*)

La función ECP utiliza el método propuesto por (Olikara & Borman, 1975) para resolver la composición, calores específicos en equilibrio y otras derivadas parciales que resultan útiles en el análisis termodinámico.

```
function [h,u,v,s,Y,cp,dlv1T,dlv1p] = ecp(p,T,phi)
```

Determinar los coeficientes  $a_1$  al  $a_7$  para el combustible y el aire, y los parámetros ajustados  $A$  al  $E$  para la Ecuación 15.

```

import SimT.fueldata import SimT.airdata import SimT.farg
A0 = airdata('GMcB_hi'); [alpha,beta,gamma,delta] = fueldata();

Kp = [0.432168E+0 -0.112464E+5 0.267269E+1 -0.745744E-4 0.242484E-8;
      0.310805E+0 -0.129540E+5 0.321779E+1 -0.738336E-4 0.344645E-8;
      -0.141784E+0 -0.213308E+4 0.853461E+0 0.355015E-4 -0.310227E-8;
      0.150879E-1 -0.470959E+4 0.646096E+0 0.272805E-5 -0.154444E-8;
      -0.752364E+0 0.124210E+5 -0.260286E+1 0.259556E-3 -0.162687E-7;
      -0.415302E-2 0.148627E+5 -0.475746E+1 0.124699E-3 -0.900227E-8];

MinMol = 1e-25;
tol     = 3e-12;
Ru      = 8314.34; % J/(kMolK)
M = [44.01 18.02 28.008 32 28.01 2.018 1.009 16 17.009 30.004]'; % kg/kMol

```

## Pre-indexar Variables

```

dcdT = zeros(4,1);

dcdp = zeros(4,1);
dfdT = zeros(4,1);
dfdp = zeros(4,1);
dYdT = zeros(10,1);
dYdp = zeros(10,1);
B = zeros(4,1);

```

## Revisar si se forma Carbono Solido

```

eps = 0.210/(alpha + 0.25*beta - 0.5*gamma);

if phi>(0.210/eps/(0.5*alpha - 0.5*gamma))
    error('El valor de phi es demasiado alto - Se formara C(s) y otras
especies'); %#ok<ERTAG>
end

```

## Estimación de las Fracciones Parciales

```

[h,u,v,s,Y] = farg(p,T,phi,1);
Y(7:10) = ones(4,1)*MinMol;

```

## Evaluar las Constantes de Equilibrio

```

patm = p/101.325e3; % Convertir de Pa a ATM

TKp = [log(T/1000) 1/T 1 T T^2]';
K = 10.^(Kp*TKp);
c = K.*[1/sqrt(patm) 1/sqrt(patm) 1 1 sqrt(patm) sqrt(patm)]';

```

```

d = [beta/alpha (gamma+0.42/eps/phi)/alpha (delta+1.58/eps/phi)/alpha]';

i = find(Y < MinMol);
Y(i)= ones(length(i),1)*MinMol;
DY3to6 = 2*tol*ones(4,1);
MaxIter = 100;
MaxVal = max(abs(DY3to6));
Iter = 0;
some = 0;

while (Iter < MaxIter) && ((MaxVal > tol) || (some < 1))
    Iter = Iter+1;
    if Iter>2,
        some = 1;
    end
    D76 = 0.5*c(1)/sqrt(Y(6));
    D84 = 0.5*c(2)/sqrt(Y(4));
    D94 = 0.5*c(3)*sqrt(Y(6)/Y(4));
    D96 = 0.5*c(3)*sqrt(Y(4)/Y(6));
    D103 = 0.5*c(4)*sqrt(Y(4)/Y(3));
    D104 = 0.5*c(4)*sqrt(Y(3)/Y(4));
    D24 = 0.5*c(5)*Y(6)/sqrt(Y(4));
    D26 = c(5)*sqrt(Y(4));
    D14 = 0.5*c(6)*Y(5)/sqrt(Y(4));
    D15 = c(6)*sqrt(Y(4));

    A(1,1) = 1+D103;
    A(1,2) = D14+D24+1+D84+D104+D94;
    A(1,3) = D15+1;
    A(1,4) = D26+1+D76+D96;

    A(2,1) = 0;
    A(2,2) = 2*D24+D94-d(1)*D14;
    A(2,3) = -d(1)*D15-d(1);
    A(2,4) = 2*D26+2+D76+D96;

    A(3,1) = D103;
    A(3,2) = 2*D14+D24+2+D84+D94+D104-d(2)*D14;
    A(3,3) = 2*D15+1-d(2)*D15-d(2);
    A(3,4) = D26+D96;

    A(4,1) = 2+D103;
    A(4,2) = D104-d(3)*D14;
    A(4,3) = -d(3)*D15-d(3);
    A(4,4) = 0;

    B(1) = -(sum(Y)-1);
    B(2) = -(2*Y(2)+2*Y(6)+Y(7)+Y(9)-d(1)*Y(1)-d(1)*Y(5));
    B(3) = -(2*Y(1)+Y(2)+2*Y(4)+Y(5)+Y(8)+Y(9)+Y(10)-d(2)*Y(1)-d(2)*Y(5));
    B(4) = -(2*Y(3)+Y(10)-d(3)*Y(1)-d(3)*Y(5));

    DY3to6 = A\B;
    MaxVal = max(abs(DY3to6));
    Y(3:6) = Y(3:6)+DY3to6/10;

```

```

i = find(Y<MinMol);

Y(i) = ones(length(i),1)*MinMol;
Y(7) = c(1)*sqrt(Y(6));
Y(8) = c(2)*sqrt(Y(4));
Y(9) = c(3)*sqrt(Y(4)*Y(6));
Y(10)= c(4)*sqrt(Y(4)*Y(3));
Y(2) = c(5)*sqrt(Y(4))*Y(6);
Y(1) = c(6)*sqrt(Y(4))*Y(5);
end

TdKdT = [1/T -1/T^2 1 2*T]';
dKdT = 2.302585*K.*(Kp(:, [1 2 4 5])*TdKdT);

dcdT(1:2) = dKdT(1:2)./sqrt(patm);
dcdT(3:4) = dKdT(3:4);
dcdT(5:6) = dKdT(1:2).*sqrt(patm);

dcdp(1:2) = -0.5.*c(1:2)./p;
dcdp(5:6) = 0.5.*c(5:6)./p;

x1 = Y(1)/c(6);
x2 = Y(2)/c(5);
x7 = Y(7)/c(1);
x8 = Y(8)/c(2);
x9 = Y(9)/c(3);
x10 = Y(10)/c(4);

dfdT(1) = dcdT(6)*x1 + dcdT(5)*x2 + dcdT(1)*x7 + dcdT(2)*x8 + ...
dcdT(3)*x9 + dcdT(4)*x10;
dfdT(2) = 2*dcdT(5)*x2 + dcdT(1)*x7 + dcdT(3)*x9 - d(1)*dcdT(6)*x1;
dfdT(3) = 2*dcdT(6)*x1 + dcdT(5)*x2 + dcdT(2)*x8 + dcdT(3)*x9+ ...
dcdT(4)*x10 - d(2)*dcdT(6)*x1;
dfdT(4) = dcdT(4)*x10 - d(3)*dcdT(6)*x1;

dfdp(1) = dcdp(6)*x1 + dcdp(5)*x2 + dcdp(1)*x7 + dcdp(2)*x8;
dfdp(2) = 2*dcdp(5)*x2 + dcdp(1)*x7 - d(1)*dcdp(6)*x1;
dfdp(3) = 2*dcdp(6)*x1 + dcdp(5)*x2 + dcdp(2)*x8 - d(2)*dcdp(6)*x1;
dfdp(4) = -d(3)*dcdp(6)*x1;

B = -dfdT;
dYdT(3:6) = A\B;
dYdT(1) = sqrt(Y(4))*Y(5)*dcdT(6) + D14*dYdT(4) + D15*dYdT(5);
dYdT(2) = sqrt(Y(4))*Y(6)*dcdT(5) + D24*dYdT(4) + D26*dYdT(6);
dYdT(7) = sqrt(Y(6))*dcdT(1) + D76*dYdT(6);
dYdT(8) = sqrt(Y(4))*dcdT(2) + D84*dYdT(4);
dYdT(9) = sqrt(Y(4)*Y(6))*dcdT(3) + D94*dYdT(4) + D96*dYdT(6);
dYdT(10)= sqrt(Y(4)*Y(3))*dcdT(4) + D104*dYdT(4) + D103*dYdT(3);

B = -dfdp;
dYdp(3:6) = A\B;
dYdp(1) = sqrt(Y(4))*Y(5)*dcdp(6) + D14*dYdp(4) + D15*dYdp(5);
dYdp(2) = sqrt(Y(4))*Y(6)*dcdp(5) + D24*dYdp(4) + D26*dYdp(6);

```

```

dYdp(7) = sqrt(Y(6))*dcdp(1) + D76*dYdp(6);
dYdp(8) = sqrt(Y(4))*dcdp(2) + D84*dYdp(4);
dYdp(9) = D94*dYdp(4) + D96*dYdp(6);
dYdp(10)= D104*dYdp(4) + D103*dYdp(3);

```

### Calculo de Propiedades Termodinámicas

```

Tcp0= [1 T T^2 T^3 T^4]';
Th0 = [1 T/2 T^2/3 T^3/4 T^4/5 1/T]';
Ts0 = [log(T) T T^2/2 T^3/3 T^4/4 1]';
cp0 = A0(:,1:5)*Tcp0;
h0 = A0(:,1:6)*Th0;
s0 = A0(:, [1:5 7])*Ts0;
Y(1)= (2*Y(3) + Y(10))/d(3) - Y(5);
Y(2)= (d(1)/d(3)*(2*Y(3) + Y(10)) - 2*Y(6) - Y(7) - Y(9))/2;
i = find(Y<MinMol);
Y(i)= ones(length(i),1)*MinMol;

```

### Determinación de las Propiedades de la Mezcla

```

h = sum(h0.*Y);
s = sum((s0 - log(Y)).*Y);
cp = sum(Y.*cp0 + h0.*dYdT*T);
MW = sum(Y.*M);
MT = sum(dYdT.*M);
Mp = sum(dYdp.*M);
R = Ru/MW;
v = R*T/p;
cp = R*(cp - h*T*MT/MW);
h = R*T*h;
s = R*(-log(patm)+s);
u = h - R*T;
dlvlT = 1 + max(-T*MT/MW, 0);
dlvlp = -1 - max(p*Mp/MW, 0);

```

### · Calculo de la Temperatura Adiabática (*Tadiabatic.m*)

```

import SimT.farg
import SimT.ecp

MaxIter = 50;
Tol = 0.00001; % 0.001°C Error Permissible de la Temperatura
Tb = 2000; % Estimado Inicial
DeltaT = 2*Tol*Tb;

Iter = 0;
[hu] = farg(p,Tu,phi,f);
while (Iter < MaxIter) && (abs(DeltaT/Tb) > Tol)
    Iter = Iter + 1;
    [hb,u,v,s,Y,cp] = ecp(p,Tb,phi); %#ok<ASGLU>
    DeltaT = (hu - hb)/cp;
    Tb = Tb + DeltaT;

```

```
end
```

```
if Iter >= MaxIter
    warning('Numero de Iteraciones Máximo Alcanzado'); %#ok<WNTAG>
end
```

· Etapas de Funcionamiento del Motor (*RatesComp.m*, *RatesComb.m* & *RatesExp.m*)

Estas funciones contienen el sistema de ecuaciones diferenciales ordinarias de primer orden, planteados en la pág. 45, para ser resueltas mediante un integrados numérico ode45 o ode23.

```
function yprime=RatesComp(theta,y)
```

```
global m_1 theta_i Cblowby omega phi V_ad ...
    VTDC e R1 R2 b R Lcte Lmax Lmin T_w f
```

```
import SimT.farg
```

```
p = y(1);
T_u = y(2);
yprime = zeros(5,1);
```

```
% Masa en la Cámara por Concepto de Blow-by
mass = m_1*exp(-Cblowby*(theta - theta_i)/omega);
```

```
% Volumen de la Cámara
V = VTDC + (sqrt(3)/2)*e*(2*R1+R2)*b*(1-sin(2/3*theta + pi/2))+ V_ad;
```

```
% Derivada del Volumen
dVdtheta = sqrt(3)/3*e*(2*R1 + R2)*b*sin(2/3*theta);
```

```
upmean = pi/3*R*omega; % Velocidad Promedio del Rotor
Const1 = 1/omega/mass;
C1 = 2.28;
hcoeff = 131*R^(-0.2)*T_u^(-0.53)*(p/100e3)^(0.8)*(C1*upmean)^0.8;
Qconv = hcoeff*(2*V/b + Lcte*b + b*(Lmin + (Lmax - Lmin)*(1 - sin(2/3*theta...
+ pi/2))))*(T_u - T_w);
[h,u,v,s,Y,cp,dvlvT,dvlvp] = farg(p,T_u,phi,f);
```

```
A = 1/mass*(dVdtheta + V*Cblowby/omega);
B = Const1*v/cp*dvlvT*Qconv/T_u;
C = 0;
D = 0;
E = v^2/cp/T_u*dvlvT^2+v/p*dvlvp;
```

```
yprime(1) = (A + B + C)/(D + E);
yprime(2) = -Const1/cp*Qconv + v/cp*dvlvT*yprime(1);
yprime(3) = p*dVdtheta;
```

```

yprime(4) = Qconv/omega;
yprime(5) = Cblowby*mass/omega*h;

function yprime = RatesComb(theta,y)

global m_1 theta_i theta_s theta_b omega Cblowby VTDC ...
    R1 R2 e b R V_1 p_1 T_1 T_w Lcte Lmax Lmin VH f phi V_ad
import SimT.*

p = y(1);
T_b = y(2);
T_u = y(3);
yprime = zeros(6,1);

% Masa en la Cámara por Concepto de Blow-by
mass = m_1*exp(-Cblowby*(theta - theta_i)/omega);

% Volumen de la Cámara
V = VTDC + (sqrt(3)/2)*e*(2*R1+R2)*b*(1-sin(2/3*theta + pi/2)) + V_ad;

% Derivada del Volumen
dVdtheta = sqrt(3)/3*e*(2*R1 + R2)*b*sin(2/3*theta);

% Fracción de Gases Quemados
x = 0.5*(1 - cos(pi*(theta - theta_s)/theta_b));
dxdtheta = pi/2/theta_b*sin(pi*(theta - theta_s)/theta_b);
x = max(x,0.0001);
x = min(x,0.9999);

upmean = pi/3*R*omega; % Velocidad Media del Rotor
C1 = 2.28;
C2 = 3.24e-3;
k = 1.3;
pm = p_1*(V_1/V)^k;
Const1 = 1/omega/mass;
hcoeffu = 131*R^(-0.2)*T_u^(-0.53)*(p/100e3)^(0.8)*(C1*upmean + ...
    C2*VH*T_1/p_1/V_1*(p - pm))^(0.8);
hcoeffb = 131*R^(-0.2)*T_b^(-0.53)*(p/100e3)^(0.8)*(C1*upmean + ...
    C2*VH*T_1/p_1/V_1*(p - pm))^(0.8);
Qconvu = hcoeffu*(2*V/b + Lcte*b + b*(Lmin + (Lmax - Lmin)*(1 - ...
    sin(2/3*theta + pi/2))))*(1 - sqrt(x))*(T_u - T_w);
Qconvb = hcoeffb*(2*V/b + Lcte*b + b*(Lmin + (Lmax - Lmin)*(1 - ...
    sin(2/3*theta + pi/2))))*sqrt(x)*(T_b - T_w);

[h_u,u_u,v_u,s,Y,cp_u,dvlTu,dvlpu]= farg(p,T_u,phi,f);
[h_b,u_b,v_b,s,Y,cp_b,dvlTb,dvlpb]= ecp(p,T_b,phi);

A = 1/mass*(dVdtheta + V*Cblowby/omega);
B = Const1*(v_b/cp_b*dvlTb*Qconvb/T_b + v_u/cp_u*dvlTu*Qconvu/T_u);
C = -(v_b - v_u)*dxdtheta - v_b*dvlTb*(h_u - h_b)/cp_b/T_b*(dxdtheta ...
    -(x - x^2)*Cblowby/omega);
D = x*(v_b^2/cp_b/T_b*dvlTb^2 + v_b/p*dvlpb);
E = (1 - x)*(v_u^2/cp_u/T_u*dvlTu^2 + v_u/p*dvlpu);

```

```

yprime(1) = (A + B + C)/(D + E);
yprime(2) = -Const1/cp_b/x*Qconvb + v_b/cp_b*dlvlTb*yprime(1)+ ...
    (h_u - h_b)/cp_b*(dxidtheta/x - (1 - x)*Cblowby/omega);
yprime(3) = -Const1/cp_u/(1 - x)*Qconvu + v_u/cp_u*dlvlTu*yprime(1);
yprime(4) = p*dVdtheta;
yprime(5) = (Qconvb + Qconvu)/omega;
yprime(6) = Cblowby*mass/omega*((1 - x^2)*h_u+x^2*h_b);

function yprime=RatesExp(theta,y,flag)
global m_1 theta_i Cblowby omega ...
    VTDC e R1 R2 b R Lcte Lmax Lmin T_w ...
    phi V_ad

import SimT.*

p = y(1);
T_b = y(2);
yprime = zeros(5,1);

% Masa en la Camara por Concepto de Blow-by
mass = m_1*exp(-Cblowby*(theta - theta_i)/omega);

% Volumen de la Cámara
V = VTDC + (sqrt(3)/2)*e*(2*R1+R2)*b*(1-sin(2/3*theta + pi/2)) + V_ad;

% Derivada del Volumen
dVdtheta = sqrt(3)/3*e*(2*R1 + R2)*b*sin(2/3*theta);
upmean = pi/3*R*omega; % Velocidad Promedio del Rotor
C1 = 2.28;
hcoeff = 131*R^(-0.2)*T_b^(-0.53)*(p/100e3)^(0.8)*(C1*upmean)^0.8;
Qconv = hcoeff*(2*V/b + Lcte*b + b*(Lmin + (Lmax - Lmin)*(1 - sin(2/3*theta +
pi/2))))*(T_b - T_w);
Const1 = 1/omega/mass;

if T_b < 1000
    [h,u,v,s,Y,cp,dlvlT,dlvlp]=farg(p,T_b,phi,1);
else
    [h,u,v,s,Y,cp,dlvlT,dlvlp]=ecp(p,T_b,phi);
end

A = 1/mass*(dVdtheta + V*Cblowby/omega);
B = Const1*v/cp*dlvlT*Qconv/T_b;
C = 0;
D = v^2/cp/T_b*dlvlT^2 + v/p*dlvlp;
E = 0;

yprime(1) = (A + B + C)/(D + E);
yprime(2) = -Const1/cp*Qconv + v/cp*dlvlT*yprime(1);
yprime(3) = p*dVdtheta;
yprime(4) = Qconv/omega;
yprime(5) = Cblowby*mass/omega*h;

```

· Rutina Principal (*SimT.m*)

```
function [Pot,eta,nuP2,nuT2,nuQ2,nuH2,nuA2,imep,V_ad,rel,error] =
SimT(R,r,a,theta_s_p,theta_i_p,theta_o_p,e,b)

global m_1 theta_i theta_s theta_b Cblowby omega VTDC e R1 R2 ...
      b R Lcte Lmax Lmin T_w f phi V_ad V_1 p_1 T_1 VH

import SimT.*
%% 0. Preparación del Algoritmo
%%% 0.1. Parámetros Geométricos del Motor
Sp      = 1e-4;           % Holgura mínima           % m.
ap      = a - Sp;        % Distancia Proyectada % m.
R1      = R + a;         % Radio 1           % m.
R2      = R + ap;        % Radio 2           % m.
VH      = sqrt(3)*e*(2*R1 + R2)*b; % Volumen Desplazado % m^3
VTDC    = VH/(r - 1);    % Volumen en el PMS % m^3
VBCD    = r/(r - 1)*VH;  % Volumen en el PMI % m^3
rel     = VBCD/VTDC;     % Relación de Compresión % 1/1

[Lcte,ang_min,ang_max] = SideArc(R,e,ap);           % m. | Rad. | Rad.
[Lmin,Lmax]           = ArcLength(R,e,a,ang_min,ang_max); % m. | m.

%%% 0.2. Parámetros Termofluidos
phi      = 0.77;         % Relación de Equivalencia
theta_b  = convang(80,'deg','rad'); % Duración de la Combustión
RPM      = 17000;        % Velocidad Nominal
T_w      = 450;          % Temperatura Superficial
Cblowby  = 0.9;          % Constante de Blow-by

omega    = convangvel(RPM,'rpm','rad/s');           % Rad./s

%%% 0.3. Determinar el Volumen para Compensar
V_ad     = (VBCD - r*VTDC)/(r-1);                   % m^3

%%% 0.4. Parámetros de Integración
dtheta   = convang(3,'deg','rad'); % Intervalo de Integración

function angle_norm = norm(angle)
    angle_norm = round((angle)/dtheta)*dtheta;
end

theta_i  = norm(theta_i_p);           % Rad.
theta_s  = norm(theta_s_p);           % Rad.
theta_o  = norm(theta_o_p);           % Rad.
theta_b  = norm(theta_b);             % Rad.
theta_e  = theta_s + theta_b;         % Rad.
options  = odeset('RelTol',1e-3);    % Rad.

%% 1. Integración Numérica
%%% 1.1. Condiciones Iniciales
```

```

p_atm    = 101.3e3;                                % Pa
p_1      = 0.9*p_atm;                              % Pa
T_atm    = 320;                                    % K
T_res    = 1400;                                    % K
f        = 1/(1 + (T_res/T_atm)*(r*(0.9) - (0.9)^0.23)); % 1/1
T_1      = T_res*r*f^1.2;                          % K
[h_1,u_1,v_1] = farg(p_1,T_1,phi,f);               % J/kg | J/kg |
V_1      = VBCD;                                    % m^3
m_1      = V_1/v_1;                                  % kg
U_1      = u_1*m_1;                                  % J

%% 1.2. Integración durante la etapa de Compresión
[theta_comp,table_comp] = ode23('RatesComp',(theta_i:dtheta:theta_s),[p_1 T_1
0 0 0],options);

%% 1.3. Condiciones al inicio de la Combustión
p_b      = table_comp(size(theta_comp,1),1);        % Pa
Tu_b     = table_comp(size(theta_comp,1),2);        % K
W_b      = table_comp(size(theta_comp,1),3);        % J
Ql_b     = table_comp(size(theta_comp,1),4);        % J
Hl_b     = table_comp(size(theta_comp,1),5);        % J
Tb_b     = Tadiabatic(p_b,Tu_b,phi,f);              % K

%% 1.4. Integración durante la etapa de Combustión
[theta_comb,table_comb] = ode23('RatesComb',(theta_s:dtheta:theta_e),[p_b Tb_b
Tu_b W_b Ql_b Hl_b],options);

%% 1.5. Condiciones al inicio de la Expansión
p_e      = table_comb(size(theta_comb,1),1);        % Pa
Tb_e     = table_comb(size(theta_comb,1),2);        % K
W_e      = table_comb(size(theta_comb,1),4);        % J
Ql_e     = table_comb(size(theta_comb,1),5);        % J
Hl_e     = table_comb(size(theta_comb,1),6);        % J

%% 1.6. Integración durante la etapa de Expansión
[theta_exp,table_exp] = ode23('RatesExp',(theta_e:dtheta:theta_o),[p_e Tb_e
W_e Ql_e Hl_e],options);

%% 2. Analizar Resultados
%% 2.1. Verificar Conservación de Masa
m_4      = m_1*exp(-Cblowby*(2*pi/omega));          % kg.
p_4      = table_exp(size(theta_exp,1),1);          % Pa
T_4      = table_exp(size(theta_exp,1),2);          % K
W_4      = table_exp(size(theta_exp,1),3);          % J
Ql_4     = table_exp(size(theta_exp,1),4);          % J
Hl_4     = table_exp(size(theta_exp,1),5);          % J
[h_4,u_4,v_4] = farg(p_4,T_4,phi,1);               % J/kg | J/kg | m^3/kg
U_4      = u_4*m_4;                                  % J
error(1) = 1 - v_4*m_4/VBCD;                        % 1/1
error(2) = 1 + W_4/(U_4 - U_1 + Ql_4 + Hl_4);       % 1/1
error    = max(error)*100;                          % (%)

%% 2.2. Fracción de mezcla quemada

```

```

theta = theta_s:dtheta:(theta_e - dtheta); % Rad.
x = 0.5*(1-cos(pi*(theta-theta_s)/theta_b))'; % 1/1

%%% 2.3. Presión Media Efectiva y Eficiencia Térmica
imep = W_4/VH; % Pa
eta = W_4*(9*(1/phi) + 1)/(m_1*(1 - f)*29.7e6)*100; % (%)
Pot = (W_4*RPM/60)*0.00134102209; % HP

%%% 2.4. Coeficiente de Transferencia por Convección Interna
h1 = calcq(theta_comp(1:(size(theta_comp,1)-1)),table_comp(1:...
(size(theta_comp,1)-1),:),'comp');
h2u = calcq(theta_comb(1:(size(theta_comb,1)-1)),table_comb(1:...
(size(theta_comb,1)-1),:),'combu');
h2b = calcq(theta_comb(1:(size(theta_comb,1)-1)),table_comb(1:...
(size(theta_comb,1)-1),:),'combb');
h2 = (1-x).*h2u + x.*h2b;
h3 = calcq(theta_exp,table_exp,'exp');
H = [h1;h2;h3]; % W/(m^2K)

%%% 2.5. Temperatura y Presión de los Gases
ang = [theta_comp(1:(size(theta_comp,1)-1),1);theta_comb(1:...
(size(theta_comb,1)-1),1);theta_exp];
P = [table_comp(1:(size(theta_comp,1)-1),1);table_comb(1:...
(size(theta_comb,1)-1),1);table_exp(:,1)]; % Pa
T_a = (1-x).*table_comb(1:(size(theta_comb,1)-1),3) + ...
x.*table_comb(1:(size(theta_comb,1)-1),2);
T = [table_comp(1:(size(theta_comp,1)-1),2);T_a;table_exp(:,2)]; % K
W = [table_comp(1:(size(theta_comp,1)-1),3);table_comb(1:...
(size(theta_comb,1)-1),4);table_exp(:,3)];

%%% 2.6. Flujo de Calor por Unidad de Área
A1 = 2*VH/b; % m^2
A2 = Lcte*b; % m^2
A3 = (Lmin + (Lmax - Lmin).*(1 - sin(2.*ang/3 + pi/2))).*b; % m^2
Aratio = A3./(A1 + A2 + A3); % 1/1
Q = Aratio.*H.*(T-T_w); % W/(m^2)

%%% 2.7. Condiciones de Frontera
nuA = ((theta_comp(1,1)+pi/2):dtheta:(theta_exp(size(theta_exp,1))-pi/2))';
nuA = nuA/1.5; % Rad.
nuP = interp1(ang/1.5,P,nuA); [Pmax,Ppos] = max(nuP); % Pa
nuT = interp1(ang/1.5,T,nuA); [Tmax,Tpos] = max(nuT); % K
nuQ = interp1(ang/1.5,Q,nuA); [Qmax,Qpos] = max(nuQ); % W/(m^2)
nuH = interp1(ang/1.5,H,nuA); [Hmax,Hpos] = max(nuH); % W/(m^2K)
nuA2 = ang/1.5; % Rad.
nuP2 = Pmax*ones(size(ang)); nuP2(1:Ppos) = nuP(1:Ppos); % Pa
nuT2 = Tmax*ones(size(ang)); nuT2(1:Tpos) = nuT(1:Tpos); % K
nuQ2 = Qmax*ones(size(ang)); nuQ2(1:Qpos) = nuQ(1:Qpos); % W/(m^2)
nuH2 = Hmax*ones(size(ang)); nuH2(1:Hpos) = nuH(1:Hpos); % W/(m^2K)
nuP2(size(nuP2,1)-(size(nuP,1)-Ppos):size(ang,1)) = nuP(Ppos:size(nuP,1));
nuT2(size(nuT2,1)-(size(nuT,1)-Tpos):size(ang,1)) = nuT(Tpos:size(nuT,1));
nuQ2(size(nuQ2,1)-(size(nuQ,1)-Qpos):size(ang,1)) = nuQ(Qpos:size(nuQ,1));
nuH2(size(nuH2,1)-(size(nuH,1)-Hpos):size(ang,1)) = nuH(Hpos:size(nuH,1));
nuA2 = convang(nuA2*3 - 3*pi/2,'rad','deg');
end

```

#### A.4. SOPORTE TEÓRICO COMPLEMENTARIO

- Aleta Longitudinal de Perfil Trapecial (Fernández, 2002)

Asumimos que las aletas se diseñan de modo que el flujo de calor se comporte como flujo modo-dimensional en la dirección radial a la carcasa del motor.

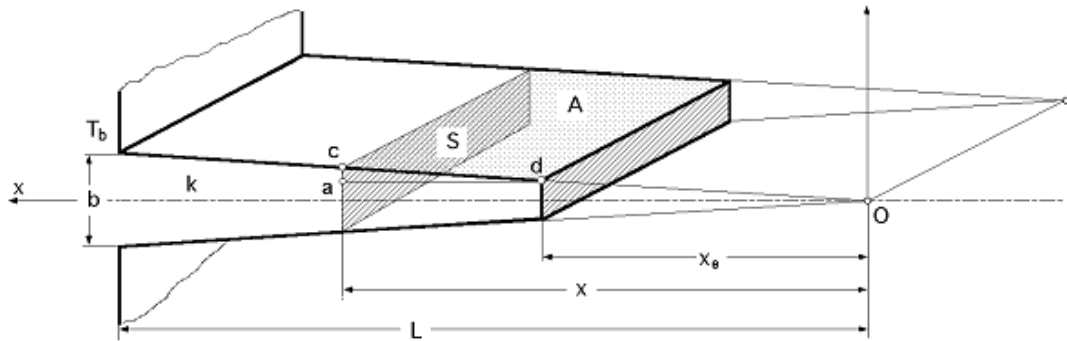
Definiendo  $\Phi$  como la diferencia entre la temperatura de la aleta en ese punto y la temperatura del aire atmosférico, podemos escribir la ecuación diferencial del flujo en las aletas como:

Ecuación 56.

$$\Phi = T - T_a$$

$$\frac{d^2\Phi}{dx^2} + \frac{1}{S} \frac{dS}{dx} \frac{d\Phi}{dx} - \frac{h_c}{k} \left( \frac{1}{S} \frac{dA}{dx} \right) \Phi = 0$$

Figura 65. Aleta Recta de Perfil Trapecial, imagen tomada de (Fernández, 2002)



Para la aleta longitudinal de anchura  $\Delta s$ , en la que se pueden despreciar las pérdidas laterales, el área de las secciones lateral  $A$  y transversal  $S$ , viene dada por:

Ecuación 57.

$$S = \frac{bx}{L} \Delta s \rightarrow \frac{dS}{dx} = \frac{b}{L} \Delta s$$

$$A = 2(x - x_e) \sqrt{1 + \left(\frac{b}{2L}\right)^2} \rightarrow \frac{dA}{dx} = 2 \sqrt{1 + \left(\frac{b}{2L}\right)^2}$$

Si  $L \gg b$ , nos aproximaremos a la condición de flujo mono dimensional  $1 + \frac{b}{2L} \cong 1$ , luego podemos aproximar  $A = 2(x - x_e)$ . Sustituyendo estos valores en la ecuación diferencial, se tiene que:

**Ecuación 58.**

$$\begin{aligned} \frac{d^2\Phi}{dx^2} + \left(\frac{L}{bx\Delta s}\right) \left(\frac{b}{L}\Delta s\right) \frac{d\Phi}{dx} - \frac{h_c}{k} \left(\frac{L}{bx\Delta s}\right) \left(2\sqrt{1 + \left(\frac{b}{2L}\right)^2}\right) \Phi &= 0 \\ \frac{d^2\Phi}{dx^2} + \frac{1}{x} \frac{d\Phi}{dx} - \frac{h_c}{k} \left(\frac{L}{bx\Delta s}\right) \left(2\sqrt{1 + \left(\frac{b}{2L}\right)^2}\right) \Phi &= 0, \\ \frac{d^2\Phi}{dx^2} + \frac{1}{x} \frac{d\Phi}{dx} - \frac{n^2}{x} \Phi &= 0, \text{ Con } n^2 = \frac{h_c}{k} \frac{2L}{b\Delta s} \sqrt{1 + \left(\frac{b}{2L}\right)^2} \end{aligned}$$

Siendo la solución de esta ecuación diferencial – EDO *Emden-Fowler* – de la forma:

**Ecuación 59.** EDO *Emden-Fowler*

$$\Phi = C_1 I_0(2n\sqrt{x}) + C_2 K_0(2n\sqrt{x})$$

Siendo  $I_0$  la función de Bessel modificada de primera especie y orden cero, y  $K_0$  la función de Bessel modificada de segunda especie y orden cero.

Lo siguiente es determinar las constantes de integración -  $C_1$  y  $C_2$  - a partir de las condiciones de contorno.

$$\begin{aligned} 1. \quad x = L \rightarrow \Phi_b = T_b - T_a &= C_1 I_0(2n\sqrt{L}) + C_2 K_0(2n\sqrt{L}) \\ 2. \quad x = x_e \rightarrow \left(\frac{d\Phi}{dx}\right)_{x_e} &= C_1 \frac{n}{\sqrt{x_e}} I_1(2n\sqrt{x_e}) - C_2 \frac{n}{\sqrt{x_e}} K_1(2n\sqrt{x_e}) = 0 \end{aligned}$$

Tenemos que:

**Ecuación 60.**

$$\begin{aligned} C_1 &= \frac{(T_b - T_a)K_1(2n\sqrt{x_e})}{K_1(2n\sqrt{x_e})I_0(2n\sqrt{L}) + K_0(2n\sqrt{L})I_1(2n\sqrt{x_e})} \\ C_2 &= \frac{T_b - T_a}{\frac{K_1(2n\sqrt{x_e})I_0(2n\sqrt{L})}{I_1(2n\sqrt{x_e})} + K_0(2n\sqrt{L})} \end{aligned}$$

El calor disipado por la aleta sería entonces:

**Ecuación 61.**

$$\dot{Q} = -k S_b \left( \frac{d\Phi}{dx} \right)_L = C_1 \frac{n}{\sqrt{x}} I_1(2n\sqrt{L}) - C_2 \frac{n}{\sqrt{x}} K_1(2n\sqrt{L})$$

$$\dot{Q} = k S_b n (T_b - T_a) \frac{-K_1(2n\sqrt{x_e}) I_1(2n\sqrt{L}) + I_1(2n\sqrt{x_e}) K_1(2n\sqrt{L})}{K_1(2n\sqrt{x_e}) I_0(2n\sqrt{L}) + K_0(2n\sqrt{L}) I_1(2n\sqrt{x_e})}$$

· Distribución de los Pernos en las Tapas Laterales (+*Adaptation/epic.m*)

Para las tapas laterales se optó por una distribución circular de los pernos, con sellos de grafito flexible, por ofrecer un buen sellado y resistencia a altas presiones y temperatura, de los cuales se seleccionó un empaque GRAPH-LOCK® 3123 de 1/16" de espesor, con un factor de empaquetadura *m* y un esfuerzo mínimo de asentamiento *y* de 2.0 y 2000 *psi* respectivamente.

Para el diseño de las tapas laterales tenemos:

1. Se asume un tamaño de perno, partiendo del más pequeño al más grande.
2. Estimar el punto de aplicación de la presión interna, *G*, que puede estimarse como:

**Ecuación 62.**

$$G = (R + e) + 2d_p$$

3. Determinar el ancho efectivo del empaque. Asumiendo un ancho del empaque *N* = 3/16", tendremos un ancho efectivo *b* =  $\sqrt{N/8}$ , es decir unos 0.0039 m.
4. Determinar las cargas de apriete  $W_m^1$  y  $W_m^2$ .

**Ecuación 63.**

$$W_m^1 = \frac{\pi}{4} G^2 P_{max} + 2\pi b m G P_{max}$$

$$W_m^2 = b\pi G y$$

El valor de la carga de apriete corresponde al mayor valor.

5. Se determina el tamaño del área apernada como:

Ecuación 64.

$$A_b = W_m / S_{per}$$

6. Se determina el número de pernos requerido, y se redondea este valor a un múltiplo de 4.

Ecuación 65.

$$N_{per} = A_b / A_{raiz}$$

El área raíz del perno, se puede obtener de la siguiente tabla.

**Tabla 8.** Dimensiones de los Pernos

Diámetro del Perno	Roscado		Dimensiones de la Tuerca		Espaciado de los Pernos $B$	Distancia Radial $R_h$	Distancia al Borde $E$
	Paso	Área (mm <sup>2</sup> )	A Cara	A Esquina			
<b>M2</b>	0,40	2,07	--	--	5,66	4,27	3,31
<b>M2.5</b>	0,45	3,39	--	--	6,99	5,08	3,91
<b>M3</b>	0,50	5,04	--	--	8,31	5,88	4,50
<b>M3.5</b>	0,60	6,78	--	--	9,64	6,68	5,10
<b>M4</b>	0,70	8,78	--	--	10,97	7,50	5,69
<b>M5</b>	0,80	14,18	--	--	13,63	9,11	6,88
<b>M6</b>	1,00	20,12	--	--	15,87	10,32	7,95

7. Estimar un valor inicial para el círculo de pernos, que sea un poco mayor a  $C_{min}$  y muy cercano a  $C_1$ .

Ecuación 66.

$$C_{min} = \frac{N_{per} B}{\pi}$$

$$C_1 = (R + e) + 2d_{per} + 2R_h$$

Este círculo de pernos se transforma en una peritrocoide de perímetro equivalente sobre la cual irán distribuidos los pernos seleccionados.

8. A esta distancia  $C$  le sumamos la distancia mínima al borde, para obtener el espesor sólido de la carcasa, y así estimar la temperatura en la base de las aletas.

- Diseño del Engranaje Interno

Los parámetros geométricos del motor nos restringen de manera automática las dimensiones de los engranajes, sin embargo, para efectos de cálculo del peso resulta necesario un mayor detalle sobre el engranaje interno y esto se hace de acuerdo a las especificaciones de la norma AGMA, para determinar el espesor mínimo del engranaje y seleccionar los materiales adecuados.

Los tipos de falla que se pueden presentar son fallas por flexión en el diente, y fallas en la superficie del diente.

- Esfuerzo por Flexión – Norma AGMA.

Esta ecuación considera los efectos de la fuerza radial y normal aplicada al engranaje, los concentradores de esfuerzos en la raíz del diente y el efecto de tener varios dientes en contacto.

**Ecuación 67.**

$$\sigma = \frac{W_t P K_v K_o K_m K_s K_b}{F J}$$

Donde  $\sigma$  es el esfuerzo en cuestión,  $W_t$  es la fuerza tangencial aplicada al diente,  $P$  corresponde al paso diametral del engranaje,  $F$  es el ancho de la cara del diente,  $J$  es un factor de geometría propio de la norma AGMA y los factores  $K_v$ ,  $K_o$ ,  $K_m$  y  $K_s$  y  $K_b$  corresponden a correcciones por concepto de velocidad, sobrecarga, montaje, tamaño y por existencia de nervaduras, respectivamente.

La fuerza transmitida se obtiene a partir del valor del par de diseño y en función del diámetro primitivo de la corona  $dp_2$ .

**Ecuación 68.**

$$dp_2 = mZ_2$$
$$W_t = \frac{T_d}{\frac{dp_2}{2}}$$

Donde  $T_d$  es el par de diseño,  $m$  es el modulo estandarizado del engranaje y  $Z_2$  es el numero de dientes de la corona.

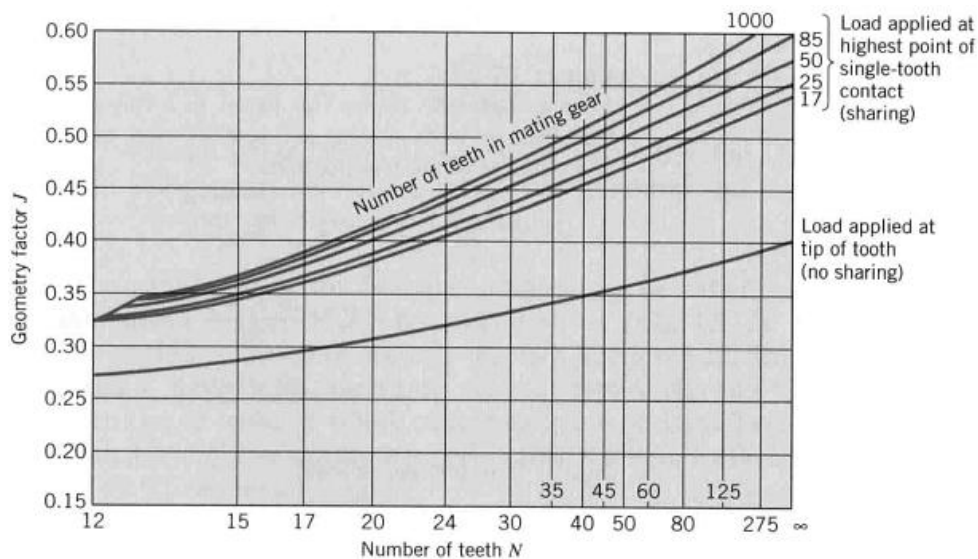
El paso diametral  $P$  es el reciproco del modulo, teniendo en cuenta que este último se encuentra en unidades métricas.

**Ecuación 69.**

$$P = \frac{1}{m}$$

El factor de geometría  $J$  es función del número de dientes de los dos engranajes, del ángulo de presión y del punto de aplicación de la carga como se muestra en la figura para un ángulo de presión de  $25^\circ$ . Para la combinación de 16, 18 y 20 dientes tenemos un factor de geometría  $J$  de 0.35, 0.37 y 0.39 respectivamente.

**Figura 66.** Factor de Geometría  $J$



La norma AGMA establece una serie de grados de calidad del engranaje  $Q_v$ , este valor iría desde 3 hasta 12, donde los engranajes con calidad de 8 a 12 se consideran elementos de alta precisión. El factor de corrección por velocidad  $K_v$  se encuentra en función de la calidad  $Q_v$  y la velocidad en la línea de contacto  $V$  medida en  $\left(\frac{ft}{min}\right)$  como se muestra en la ecuación.

**Ecuación 70.**

$$K_v = ((A + \sqrt{V})/A)^B$$

$$A = 50 + 56(1 - B) \quad B = 0.25(12 - Q_v)^{\frac{2}{3}} \quad V = \pi d p_2 \text{RPM}_2$$

Para efectos prácticos y dada la escala del engranaje se optó hacer las estimaciones con un grado de calidad  $Q_v$ , igual a 11.

El factor de sobrecarga  $K_o$  es altamente subjetivo y varía en buena medida dependiendo del tipo de máquina que se esté empleando, para nuestro caso asumiremos vibración ligera con un factor  $K_o = 1.2$

El factor de distribución de carga  $K_m$  depende en buena medida de la calidad y la precisión del montaje, que para un montaje adecuado con tolerancias pequeñas y engranajes de precisión se estima  $K_m$  alrededor de 1.3.

El factor de tamaño  $K_s$  se encuentra en función del paso diametral y para engranajes con módulo pequeño es un valor que tiende a 1 y el factor de corrección por nervaduras  $K_b$  es igual a 1 por ausencia total de nervadura en los engranajes.

· Esfuerzos Permisibles – Norma AGMA. (Daly, 2002)

La norma AGMA define dos esfuerzos permisibles  $\sigma_{adm}$  y  $\sigma_{c adm}$  para esfuerzos a flexión y por contacto respectivamente.

**Ecuación 71.**

$$\sigma_{adm} = \frac{S_t Y_n}{S_F K_T K_R}$$

Donde  $S_t$  es el esfuerzo permisible a compresión,  $S_F$  es el factor de seguridad empleado,  $Y_n$  es el factor de ciclo de vida,  $K_T$  y  $K_R$  son los factores por temperatura y confiabilidad.

En ausencia de condiciones de operación particulares podemos asumir que los factores por temperatura y confiabilidad del material,  $K_T$  y  $K_R$  respectivamente, son igual a 1.

El punto de diseño busca que  $\sigma$  sea igual a  $\sigma_{adm}$ , Podríamos reescribir las ecuaciones en función de los parámetros de diseño y eliminando los factores que sean igual a 1.

**Ecuación 72.**

$$F = \frac{W_t P K_v K_o K_m K_s K_b}{\sigma_{adm} J}$$