

Optimización automatizada del diseño de un ventilador de flujo tubo-axial mediante algoritmos evolutivos



T E S I S

Maestría en Ciencias en Sistemas Inteligentes

Instituto Tecnológico y de Estudios Superiores de Monterrey

Por

Lic. Abel Chávez Morales

Mayo 2009

Optimización automatizada del diseño de un ventilador de flujo tubo-axial mediante algoritmos evolutivos

TESIS

Maestría en Ciencias en
Sistemas Inteligentes

Instituto Tecnológico y de Estudios Superiores de Monterrey

Por

Lic. Abel Chávez Morales

Mayo 2009

Instituto Tecnológico y de Estudios Superiores de Monterrey

División de Graduados en Mecatrónica y Tecnologías de Información

Los miembros del comité de tesis recomendamos que la presente tesis de Abel Chávez Morales sea aceptada como requisito parcial para obtener el grado académico de Maestro en Ciencias en:

Sistemas Inteligentes

Comité de tesis:

Dr. Eduardo Uresti Charre

Asesor de la tesis

Dr. Manuel Valenzuela Rendón

Sinodal

Dr. Santiago E. Conant Pablos

Sinodal

Dr. Joaquín Acevedo Mascarúa
Director de Investigación y Posgrado
Escuela de Ingeniería

Mayo de 2009

Optimización automatizada del diseño de un ventilador de flujo tubo-axial mediante algoritmos evolutivos

Por

Lic. Abel Chávez Morales



TESIS

Presentada a la División de Mecatrónica y Tecnologías de Información
Este trabajo es requisito parcial para obtener el grado académico de Maestro en
Ciencias en Sistemas Inteligentes

Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Monterrey

Monterrey, N.L. Mayo de 2009

A mi hijo Ángel Abel por las alegrías que ha traído a mi vida. A mi esposa Lorena la mujer con quién he decidido compartir todo lo que soy. A mis padres Alfredo y Aurelia por el enorme cariño que siempre me han mostrado. A Josefina Chávez por el lazo especial que nos une. A mis hermanos y demás familiares porque no hay nada mejor que tener una familia.

Reconocimientos

A mi asesor Dr. Eduardo Uresti Charre por mostrarme las herramientas usadas en la presente tesis.

A mis sinodales el Dr. Manuel Valenzuela Rendón y el Dr. Santiago E. Conant Pablos por sus consejos, aportaciones e interés por mi trabajo.

Al Dr. Arturo Galván por mostrarme valiosos conocimientos en el sistema operativo Debian y sus consejos en la simulación de flúidos.

Al MC. Cesar Villarreal por aconsejarme la utilización de *OpenFOAM* y al Dr. Noel León por el equipo de trabajo que formamos.

A todos aquellos que de alguna forma contribuyeron a la realización de esta tesis.

ABEL CHÁVEZ MORALES

Instituto Tecnológico y de Estudios Superiores de Monterrey

Mayo 2009

Optimización automatizada del diseño de un ventilador de flujo tubo-axial mediante algoritmos evolutivos

Abel Chávez Morales, M.C.

Instituto Tecnológico y de Estudios Superiores de Monterrey, 2009

Asesor de la tesis: Dr. Eduardo Uresti Charre

En el presente trabajo se presenta una metodología para una optimización automatizada de un aspa de un ventilador de flujo tubo-axial. La evaluación del desempeño de las distintas aspas se hizo dentro de un software libre para el modelado de fluidos conocido como *OpenFOAM*. El algoritmo evolutivo encargado de la optimización fue programado en un lenguaje de programación conocido como *Python*. El algoritmo utilizado en optimización evolutiva es una variación del algoritmo conocido como *Genitor*. La función objetivo o parámetro a minimizar fue la velocidad promedio de salida en uno de los extremos del túnel donde se encuentra montado el ventilador. Dicho promedio se obtiene al variar la velocidad de rotación del ventilador, en total son siete velocidades distintas las que se promediaron. En este trabajo se ofrece además una recopilación de las leyes que rigen el comportamiento de los ventiladores, sobre fluidos computacionales y sobre algoritmos genéticos. Estas recopilaciones tienen la finalidad de proporcionar la herramienta necesaria para comprender de manera general las implicaciones de un proceso de optimización de esta magnitud. Cabe mencionar que antes que proporcionar la mejor aspa para el ventilador, está presentar el desarrollo del método automatizado, el cuál en un momento determinado puede ser mejorado por alguno de nuestros lectores o futuros tesisistas. Aún así, se presentan resultados de la mejor aspa encontrada,

demostrándose que es posible lograr automatizar este tipo de procesos. Esperamos con el presente trabajo el haber contribuido al avance de la optimización de diseños.

Índice general

Reconocimientos	VI
Resumen	VII
Índice de cuadros	XIII
Índice de figuras	XIV
Capítulo 1. Introducción	1
1.1. Objetivo	2
1.2. Hipótesis	3
1.3. Justificación	3
1.4. Guía del documento	5
Capítulo 2. Fundamentos teóricos de ventiladores	6
2.1. Conceptos del ventilador	6
2.2. Conceptos del sistema	7
2.3. Combinando los conceptos del ventilador y el sistema	10
2.4. Ajustando el funcionamiento del ventilador	11
2.5. Leyes de ventilación	13
2.6. Tipos de abanicos	14
2.6.1. Principio básico	14
2.6.2. Abanicos centrífugos	15
2.6.3. Abanicos de flujo axial	20

2.7. Resumen del capítulo	25
Capítulo 3. Fundamentos teóricos de algoritmos genéticos	26
3.1. Introducción a los algoritmos genéticos	26
3.2. Anatomía de un algoritmo genético	28
3.3. Características de los algoritmos genéticos	28
3.4. Analogía de un algoritmo genético con la naturaleza	30
3.5. Codificación de las variables	31
3.6. Evaluación y selección	32
3.6.1. Métodos de selección	33
3.6.1.1. Hacinamiento determinista	35
3.7. Funcionamiento de un algoritmo genético	36
3.8. Operadores genéticos	37
3.8.1. Crossover (cruce)	37
3.8.2. Mutación	40
3.9. Genitor	41
3.10. Resumen del capítulo	42
Capítulo 4. Fundamentos teóricos de fluidos computacionales	43
4.1. Ecuaciones de conservación	44
4.1.1. Continuidad	44
4.1.2. Cantidad de movimiento	46
4.1.3. Turbulencia	46
4.1.3.1. La hipótesis de Boussinesq	48
4.1.3.2. El modelo $k - \varepsilon$	48
4.1.3.3. El modelo RNG $k - \varepsilon$	50
4.1.3.4. El modelo $k - \varepsilon$ factible	50
4.1.3.5. El modelo RSM	51
4.2. El modelo de corte rotativo	51
4.3. El algoritmo SIMPLE	52

4.4. Resumen del capítulo	53
Capítulo 5. Modelado de un ventilador de flujo tubo-axial en <i>Open-FOAM</i>	54
5.1. Descripción de <i>OpenFOAM</i>	54
5.2. Justificación del uso de <i>OpenFOAM</i>	55
5.3. La geometría básica de OpenFoam	56
5.4. Parámetros del aspa	58
5.5. Visualización de resultados	62
5.6. Resumen del capítulo	64
Capítulo 6. Algoritmos empleados en proceso de optimización	65
6.1. Algoritmo de optimización	65
6.2. Algoritmo de evaluación	69
6.3. Algoritmo de procesado de un cromosoma	71
6.4. Resumen del capítulo	74
Capítulo 7. Resultados	75
7.1. Proceso de optimización automatizado	75
7.2. Codificación del aspa	79
7.3. Comparación entre aspas	81
7.4. Detección de errores	82
7.5. Desempeño del algoritmo de optimización	83
7.6. Relación entre convergencia y velocidad de salida	84
Capítulo 8. Discusión	85
8.1. Conclusiones	85
8.2. Trabajo futuro	86
Apéndice A. POBLACION FINAL DE LA MEJOR CORRIDA (Evaluación y puntos que definen cada aspa)	88

Bibliografia **93**

Vita **95**

Índice de cuadros

3.1. Ejemplo de resultados obtenidos con una función de evaluación	33
--	----

Índice de figuras

2.1. Curva del ventilador [7]	7
2.2. Variaciones en la curva del ventilador [7]	8
2.3. Curva de resistencia del sistema [7]	9
2.4. Variación de la curva de resistencia del sistema [7]	10
2.5. Punto de Operación [7]	11
2.6. Variación de los puntos de operación [7]	12
2.7. Abanico de aspas curvadas hacia adelante [16]	15
2.8. Curva de desempeño de un abanico de aspas curvadas hacia adelante [16]	16
2.9. Abanico centrífugo de aspas radiales [16]	17
2.10. Abanico centrífugo con aspas de punta radial [16]	18
2.11. Abanico centrífugo con aspas radiales [16]	18
2.12. Abanico con aspas inclinadas hacia atrás [16]	19
2.13. Abanico centrífugo con aspas aerodinámicas inclinadas hacia atrás [16]	19
2.14. Curva del abanico con aspas inclinadas hacia atrás [16]	20
2.15. Abanico helicoidal [16]	21
2.16. Curva del abanico helicoidal [16]	22
2.17. Abanico tuboaxial [16]	23
2.18. Curva del abanico tuboaxial [16]	23
2.19. Abanico tipo vaneaxial [16]	24
2.20. Curva del abanico tipo vaneaxial [16]	25
3.1. Procedimiento de un algoritmo genético [12]	36

3.2.	Diagrama de funcionamiento de una posible implementación de algoritmo genético [12]	37
3.3.	<i>Crossover</i> de n puntos	38
3.4.	<i>Crossover</i> uniforme	39
3.5.	<i>Crossover</i> de dos puntos asimétrico	40
4.1.	Un volumen rectangular con fluido entrante y saliente puede ser usado para ilustrar la ecuación de conservación [11]	45
5.1.	Celdas tipo <i>hex</i> : Puntos, aristas y caras definidas en <i>OpenFOAM</i>	56
5.2.	Cuatro celdas tipo <i>hex</i> definiendo el aspa del ventilador.	57
5.3.	Mallado proyectado sobre un aspa del ventilador.	58
5.4.	Vista del aspa hacia el sentido negativo del eje X	60
5.5.	Vista del aspa hacia el sentido positivo del eje Y	60
5.6.	Vista del aspa hacia el sentido negativo del eje Z	61
5.7.	Flujo de aire con velocidad absoluta	62
5.8.	Flujo de aire con velocidad relativa	63
5.9.	Ventilador de flujo axial con ocho aspas	63
6.1.	Algoritmo de optimización	66
6.2.	Algoritmo de evaluación de la población	70
6.3.	Algoritmo de procesado de un cromosoma en <i>OpenFOAM</i>	72
7.1.	Cromosomas evaluados en la primera corrida	76
7.2.	Promedio de la población en cada ciclo de la primera corrida	76
7.3.	Mejores individuos encontrados en la primera corrida	77
7.4.	Cromosomas evaluados en la segunda corrida	77
7.5.	Promedio de la población en cada ciclo de la segunda corrida	78
7.6.	Mejores individuos encontrados en la segunda corrida	78
7.7.	Comparación del promedio de velocidad en cada ciclo de las dos mejores corridas	79

7.8. Mejor aspa encontrada: Vista hacia el sentido negativo del eje X	79
7.9. Mejor aspa encontrada: Vista hacia el sentido positivo del eje Y	80
7.10. mejor aspa encontrada: Vista del aspa hacia el sentido negativo del eje Z	80
7.11. Ventilador con peor desempeño	82
7.12. Ventilador con mejor desempeño	82

Capítulo 1

Introducción

Los algoritmos genéticos han llegado a ser una de las herramientas más utilizadas por la inteligencia artificial. Los algoritmos genéticos se refieren al modelo introducido e investigado por John Holland (1975 [8]) y por los estudiantes de Holland (por ejemplo DeJong, 1975 [5]).

Los algoritmos genéticos son una familia de modelos computacionales inspirados en la evolución natural. Estos algoritmos codifican una solución potencial a un problema específico en un cromosoma simple como estructura de datos y aplican operadores de recombinación de estas estructuras para preservar información crítica. Los algoritmos genéticos son a menudo vistos como optimizadores de funciones, aunque el rango de problemas en los cuales los algoritmos genéticos han sido aplicados es extenso. Una implementación de algoritmos genéticos empieza con una población de cromosomas típicamente aleatorios. Uno entonces evalúa estas estructuras y brinda oportunidades de reproducción de tal forma que los cromosomas que representan una mejor solución a la función objetivo tengan mayor oportunidad de reproducirse que los cromosomas con peores soluciones. El mejor comportamiento de una solución es típicamente definido con respecto a la población actual [19].

En los últimos años los algoritmos genéticos han sido utilizados en un gran número de problemas de optimización en diseño, en donde la búsqueda exhaustiva en la forma óptima de la geometría a menudo consume demasiado tiempo y resulta muy cara. Pero con el uso de los algoritmos genéticos combinados con programas de simulación (Método de elemento finito, Dinámica de Fluido Computacional, etc.) se pueden ofrecer una

solución más rápida, aprovechando la aceleración del proceso a través de estos métodos de simulación e imitando los procesos naturales que los sistemas biológicos usan.

Los motores eléctricos son los principales convertidores de energía eléctrica en mecánica y tienen la mayor aplicación dentro de la industria para diferentes tipos de aplicaciones, sin embargo representa un gran consumo de energía. La eficiencia de un motor eléctrico está determinada por la cantidad de pérdidas que ocurre en su interior, en donde un alto porcentaje se pierde por transferencia de calor. Un incremento en la eficiencia en el motor representaría un ahorro económico en la fabricación de los motores al poder reducir la cantidad de los materiales utilizados para su construcción.

La optimización de la forma se refiere a problemas relacionados con una forma definida: Encontrar la forma óptima (en 2 o 3 dimensiones) para cierto problema, siempre que satisfaga ciertos requerimientos. Es decir, la optimización se refiere a la meta de lograr la mejor solución posible para el problema [3].

En esta investigación se combinan las definiciones antes mencionadas. La optimización automatizada de la forma del aspa de un ventilador a través de algoritmos genéticos y la validación de resultados con un programa CFD libre, esto representa una innovación en la búsqueda de una mejor eficiencia en los motores eléctricos.

1.1. Objetivo

El objetivo de esta investigación es el estudio de la optimización de forma del aspa de un ventilador mediante la implementación de algoritmos genéticos como técnica de optimización. Se desea que el aspa trabaje relativamente bien en un amplio rango de velocidades. Se hace uso de la herramienta computacional *OpenFOAM* como método de simulación y validación de los resultados obtenidos por algoritmos genéticos. Así mismo se hace uso del lenguaje Python para lograr integrar nuestros algoritmos evolutivos y *OpenFOAM* en una forma automatizada.

1.2. Hipótesis

Con el avance alcanzado hasta el momento en el área de inteligencia artificial es posible mediante la aplicación de algoritmos genéticos optimizar la forma aerodinámica de un ventilador en un motor de corriente alterna con el propósito de disminuir las pérdidas mecánicas y aumentar el gasto de aire y, como consecuencia, incrementar la eficiencia en el motor. Además es posible lograr que el proceso se realice en forma automática.

1.3. Justificación

Los motores eléctricos son los mayores consumidores de electricidad en la industria y en el comercio. Casi la mitad de la energía eléctrica usada y cerca de las dos terceras partes de la utilizada en la industria es consumida por motores eléctricos.

El número estimado de motores que operan bajo carga variable de entre los motores de tamaño superior a los 50 CV (caballos de vapor) es aproximadamente el 25%. Igualmente se espera un crecimiento de los mismos superior al que experimentará el resto de los motores. De ahí la necesidad de una utilización más eficiente de la energía en este modo de operación cada vez más común en los motores.

Como la mayoría de los sistemas trabajan por debajo de su capacidad nominal durante la mayor parte del tiempo, se han desarrollado métodos para reducir la salida de los motores a un nivel que coincida con lo que se le demanda. Los reguladores de tiro, válvulas de estrangulación y sistemas de recirculación que se usan tradicionalmente para este fin son muy ineficientes desde el punto de vista energético. Reduciendo la velocidad de la carga, por ejemplo de una bomba o un ventilador, se obtendría el mismo objetivo de una manera más adecuada.

Es en este campo donde toman importancia los variadores de velocidad, cuya función principal es ofrecer diferentes velocidades de trabajo. Esto permitirá ahorros energéticos y, por tanto, ahorro en costes en aquellas máquinas accionadas por motores eléctricos que no trabajan a plena carga la mayor parte del tiempo [10].

Una mayor eficiencia en el motor ahorraría energía y reduciría el gasto de una compañía en electricidad.

Existen diferentes formas de aumentar la eficiencia en un motor eléctrico como lo son: disminuir las pérdidas, mejorar el grado de acero en la laminación y diseño de ranura. También con más cobre y un mejor sistema de enfriamiento.

Todas las vías anteriores aumentarían en gran medida el costo de producción de un motor eléctrico, provocando una gran desventaja a la empresa productora frente a sus competidores. Con la técnica de optimización y las herramientas computacionales antes mencionadas se genera un nuevo concepto de optimización, en donde se sigue conservando intacta la parte eléctrica del motor [3].

Hemos optado por utilizar *OpenFOAM* como software de simulación y el lenguaje *Python* como elemento integrador del proceso, tratamos de evitar problemas encontrados en procesos de optimización anteriores. Decidimos entonces seguir la recomendación dada por César Villarreal en su trabajo de tesis relacionado con la optimización de una turbina de aire tipo savonius: “Otra recomendación es evitar el uso de una red inalámbrica si las licencias del software están en un servidor. Si esta red inalámbrica es débil, la conexión puede perderse y el proceso puede interrumpirse debido a esta falla. Para evitar esto, puede usarse una red local segura o licencias dentro de la computadora. Otra forma de evitar esto, es el uso de software libre tal como, OpenFOAM (CFD), DAKOTA (algoritmos) y un lenguaje de programación para integrar el proceso” [17, p. 89-90].

Con el desarrollo de esta tesis se pretende continuar con la brecha abierta por Waldo Arcos en la forma de optimización de la energía consumida por un motor eléctrico. Waldo modeló un aspa de ventilador con un número fijo de revoluciones por minuto, en un proceso que no era automático. Tratamos de ir un paso más allá al utilizar diferentes velocidades de trabajo del ventilador, así como también se ha logrado automatizar el proceso de optimización de la forma del aspa, algo que en la tesis de Waldo no fue posible como lo expresa en su trabajo: “Debido a que el proceso de evaluación es complicado y no puede ser hecho en forma automática fue necesario tener un equipo de

evaluadores” [3, p. 70].

1.4. Guía del documento

En el capítulo 1 se da una descripción general del problema a tratar y el objetivo a lograr. Se plantea la hipótesis propuesta y se da una justificación por la cual se está desarrollando esta investigación. Además se ofrece la guía del presente documento.

En el capítulo 2 se establecen los fundamentos teóricos de ventiladores. Se presenta una descripción de los principales tipos de ventiladores y de las leyes generales de los abanicos.

En el capítulo 3 se da una introducción a los algoritmos genéticos. Se hace mención del algoritmo Genitor, ya que una variación de este algoritmo será usada como método de optimización de la forma del aspa del ventilador en el desarrollo de esta investigación.

En el capítulo 4 se da una introducción a fluidos computacionales. Se hace mención de las principales reglas y modelos que se emplean actualmente en el modelado de fluidos.

En el capítulo 5 se describe brevemente *OpenFOAM* y se presenta la modelación del ventilador de flujo tubo-axial a optimizar dentro de este paquete computacional.

En el capítulo 6 se describen los algoritmos que hacen posible la optimización y la integración de la variación del algoritmo Genitor con *OpenFOAM*.

En el capítulo 7 se describen los resultados obtenidos y se hace una discusión acerca de ellos.

En el capítulo 8 se dan las conclusiones y el trabajo futuro.

Capítulo 2

Fundamentos teóricos de ventiladores

En esta sección se tocarán temas referentes a los ventiladores. Se hablará por separado de los conceptos del ventilador y del sistema, después se combinará ambos conceptos. Se hablará de cómo hacer el ajuste del funcionamiento de un ventilador. Enseguida se tocarán las principales leyes de ventilación. Finalizaremos el capítulo con la descripción los abanicos centrífugos y de tipo axial. De entre los abanicos de tipo axial que describiremos en este capítulo, el abanico de tipo tubo-axial es de especial interés para nosotros, ya que es el tipo de ventilador que hemos modelado.

2.1. Conceptos del ventilador

Un ventilador actúa como una bomba impulsora de aire. La velocidad a la cual un ventilador puede “bombear” aire depende de la presión que éste tenga que superar. Este principio también se refleja en la bombas de agua. Una bomba de agua es capaz de transferir más agua a través de una manguera de 2 pulg. de diámetro que a través de una de 1 pulg. de diámetro, ya que la manguera de 1 pulg. produce más resistencia a la corriente del agua.

Para un ventilador, cada volumen (pcm-Pies Cúbicos por Minuto) corresponde a una resistencia específica en el flujo del aire (P_e -Presión estática). La serie de puntos de pcm y P_e para un ventilador a una rpm (revolución por minuto) constante es llamada, la curva del ventilador. Un ejemplo de ésta es mostrado a 700 rpm. A 0.25 pulg. de columnas de agua de P_e , este ventilador proporcionará 1,000 pcm, ver figura 2.1. Si la

Presión Estática aumenta, los pcm disminuyen. Si la Presión Estática disminuye, los pcm aumentarán.

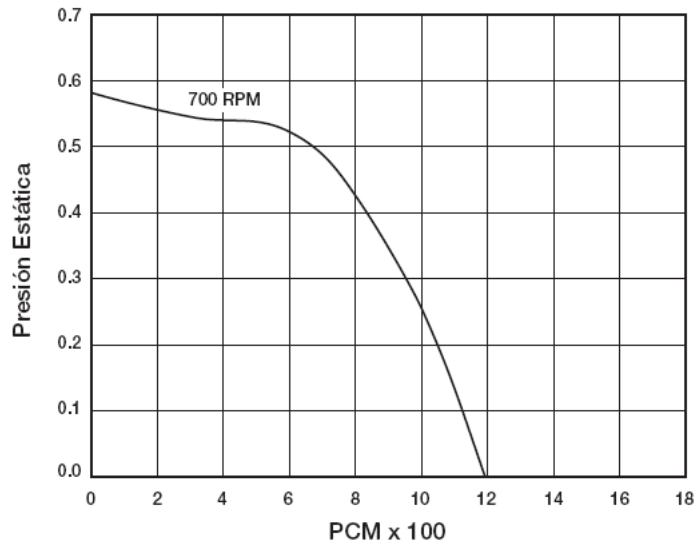


Figura 2.1: Curva del ventilador [7]

A 700 rpm, el punto de operación se deslizará por la curva del ventilador a manera que la Presión Estática va cambiando, pero nunca se desligará de la curva. Para que un ventilador pueda operar a un punto fuera de la curva, las rpm deberán ser cambiadas.

La gráfica 2.2 muestra como las rpm afectan la curva del ventilador. Note que la forma general de las curvas son las mismas. Cambiando las rpm simplemente hace que la curva se mueva hacia afuera o hacia adentro [7].

2.2. Conceptos del sistema

Para un volumen de aire, un sistema de distribución de aire produce una resistencia al flujo del aire (P_e) esta resistencia es la suma de todas las pérdidas de presión estática a medida que el aire pasa a través del sistema. Dentro de los elementos que producen resistencia se incluyen el sistema del ducto, compuertas, rejillas, serpentines, etc.

Un ventilador es simplemente un equipo que crea la diferencia en la presión para mover el aire a través del sistema. Mientras más grande es la diferencia de la presión

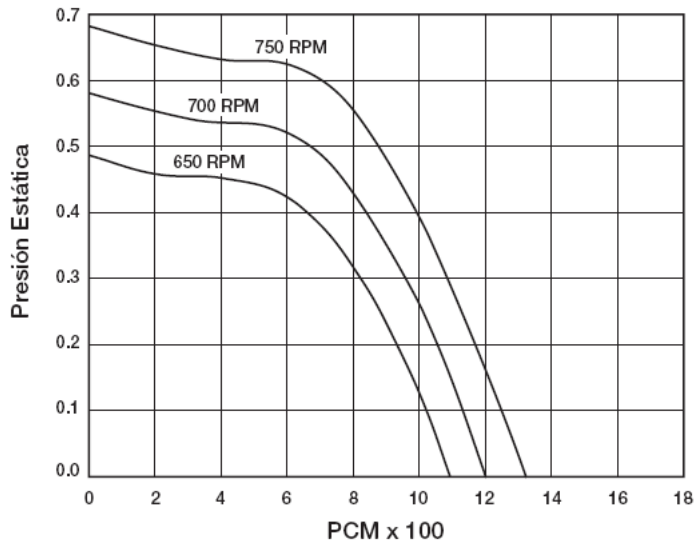


Figura 2.2: Variaciones en la curva del ventilador [7]

creada por el ventilador, más grande será el volumen de aire movido a través del sistema. Éste es el mismo principio relacionado con las bombas de agua. La única diferencia en este caso es que el ventilador bombea aire y no agua. Varias pruebas han establecido una relación entre pcm y Pe. Esta relación es parabólica y obtiene su forma en la siguiente ecuación:

$$Pe = K(pcm)^2$$

La letra K es la constante que refleja lo cerrada que es la parábola. De acuerdo a esta ecuación literariamente afirma que la presión estática (Pe) varía tanto como la elevación cuadrada de los pcm.

Por ejemplo, cuando los pcm duplican su valor, la presión estática aumenta 4 veces. Las gráficas de las figuras 2.3 y 2.4 muestran este concepto.

Ejemplo: Si un sistema es diseñado para mover 1,000 pcm a una resistencia de 0.25 pulg. de Pe. ¿Qué presión estática tendrá que superar el ventilador para producir 2,000 pcm de aire?

Solución: Ya que la presión estática varía, así como la elevación cuadrada de los pcm, podemos resolver la presión estática nueva Pe (Pe₂) con la siguiente ecuación:

$$Pe_2 = Pe_1 \left(\frac{pcm_2}{pcm_1} \right)^2 = 0.25 \text{ pulg.} \left(\frac{2000 \text{ pcm}}{1000 \text{ pcm}} \right)^2 = 1 \text{ pulg.}$$

Si nos referimos a la gráfica 2.3, estos resultados despliegan la curva de resistencia, en el sistema, de un punto A hacia un punto B.

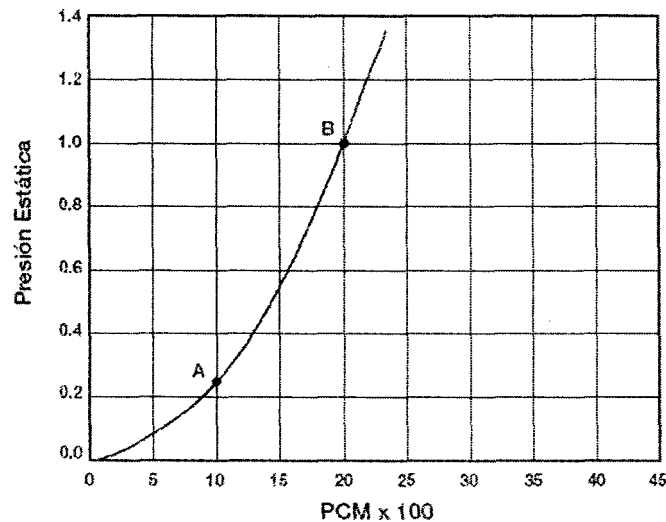


Figura 2.3: Curva de resistencia del sistema [7]

Para este sistema en particular, es imposible mover 2,000 pcm a solamente 0.25 pulg. de Pe. En todos los sistemas de ventilación cada pcm requiere una Pe única. Esta serie de pcm/Pe forma una curva de resistencia como la que se muestra en la gráfica 2.3. Una vez que la curva de resistencia del sistema es definida, al cambiar las rpm del ventilador también cambiarán simultáneamente los pcm y la presión estática, lo cual resultará en un despliegue de la curva de resistencia [7].

Nota: Físicamente cambiando el sistema alteraría la resistencia del mismo. Por ejemplo, cerrando una compuerta de 100 % de apertura a sólo 50 % le dará resistencia y aumentaría la empinada de la curva. El mismo efecto ocurre cuando los filtros se van deteriorando. La gráfica 2.4 muestra este concepto.

La curva A representa a un sistema que requiere 0.5 pulg. de Pe para mover 1,000 pcm. La curva B requiere 0.75 pulg. de Pe para mover la misma cantidad de aire. Esto

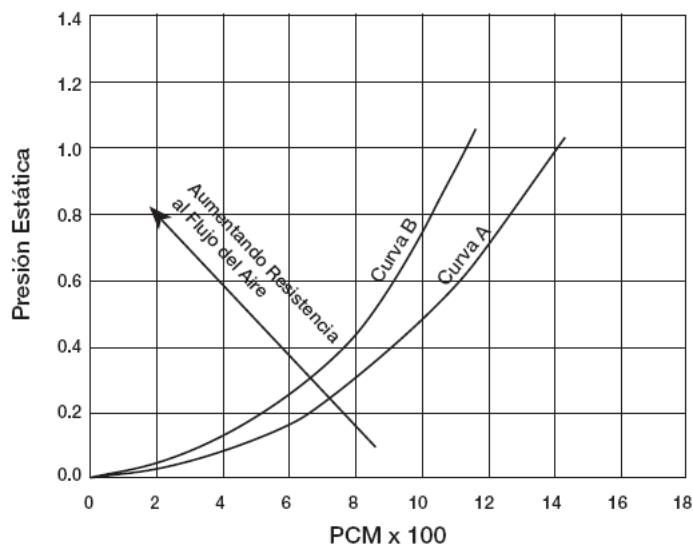


Figura 2.4: Variación de la curva de resistencia del sistema [7]

es como típicamente un sistema reacciona cuando se incrementa la resistencia.

En esta sección, hay tres puntos claves a tomar muy en cuenta:

1. La relación entre el volumen de aire movido y la presión estática.
2. Para un sistema de ventilación estable y bien firme, los puntos de operación deben aferrarse a la curva definiendo las características de los pcm y la presión estática del sistema.
3. Así como los elementos de resistencia cambian, la inclinación de la curva de resistencia en el sistema también cambia.

2.3. Combinando los conceptos del ventilador y el sistema

Las dos secciones previas introducen las curvas del ventilador y las curvas de resistencia en el sistema. Esta sección mostrará cómo éstas se relacionan mutuamente para proporcionar un mejor entendimiento del modo en que el sistema del ventilador opera como una unidad completa.

Recuerde que la curva de un ventilador es la serie de puntos en las cuáles puede el ventilador operar a un rpm constante. De la misma forma, una curva de resistencia en un sistema es la serie de puntos en las cuáles el sistema puede operar. El punto de operación (pcm, Pe) para la combinación del sistema de ventilación se encuentra donde se interceptan estas dos curvas [7] (ver figura 2.5).

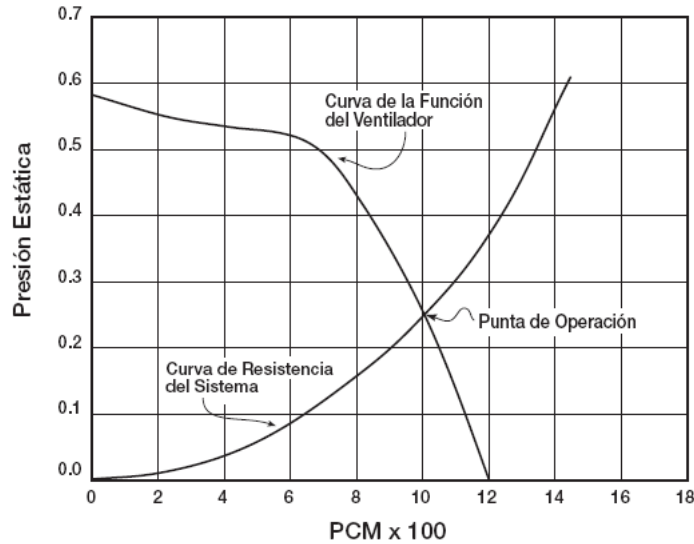


Figura 2.5: Punto de Operación [7]

2.4. Ajustando el funcionamiento del ventilador

Existe una relación directa entre los pcm y las rpm dentro de un sistema de ventilación. Al duplicar las rpm del ventilador también pasará lo mismo con los pcm distribuidos [7].

Ejemplo:

La gráfica 2.3 muestra una curva a 700 rpm con un punto de operación de 1,000 pcm a 0.25 pulg. de Pe. ¿Cuántas rpm serían requeridas para poder mover 2,000 pcm a través del mismo sistema?

Solución:

Dentro de un sistema de ventilación, los pcm son directamente relacionados a las

rpm. Por lo tanto, las nuevas rpm (rpm_2) pueden ser determinadas de acuerdo a la siguiente ecuación:

$$rpm_2 = rpm_1 \left(\frac{pcm_2}{pcm_1} \right) = 700 \text{ rpm} \left(\frac{2000 \text{ pcm}}{1000 \text{ pcm}} \right) = 1400 \text{ rpm}$$

Al referirnos a la gráfica 2.6, este resultado despliega la curva de resistencia en el sistema, de 700 rpm a 1,400 rpm.

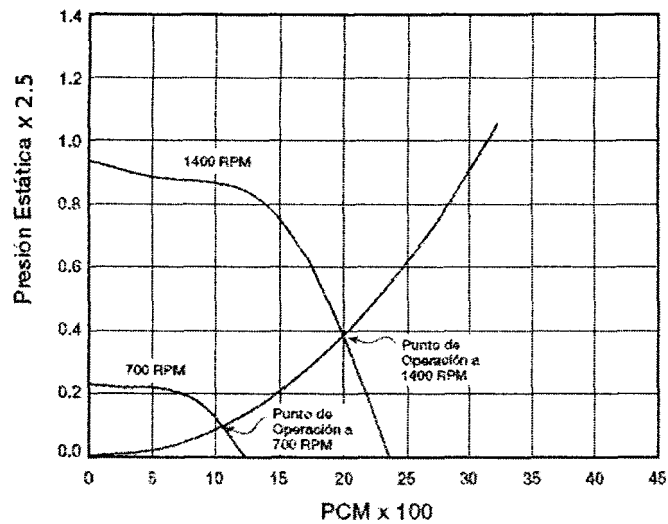


Figura 2.6: Variación de los puntos de operación [7]

Observe que como duplicamos el volumen del aire de 1,000 pcm a 2,000 pcm, la presión estática aumentó de 0.25 pulg. a 1.0 pulg. Debe de tomarse en cuenta que no estamos cambiando el sistema, solamente aumentando la velocidad del ventilador. Por lo tanto, debemos de permanecer en la curva de resistencia del sistema. Dentro de un sistema, la presión estática varía tanto como la elevación cuadrada de los pcm.

Ya que los pcm y las rpm son directamente proporcionales, podemos relacionarlos con la siguiente ecuación

$$Pe_2 = Pe_1 \left(\frac{rpm_2}{rpm_1} \right)^2$$

Por ejemplo:

$$Pe_2 = 0.25 \text{ pulg.} \left(\frac{1400 \text{ rpm}}{700 \text{ rpm}} \right)^2 = 1.0 \text{ pulg.}$$

Esto verifica que el punto de operación en la curva de las 1,400 rpm (2,000 pcm a 1.0 pulg. de Pe). Con este ejemplo, queda claro como los pcm, rpm y la Pe operan unidos en un sistema de ventilación firme y estable.

2.5. Leyes de ventilación

En un sistema de ventilación estable y firme, a medida que cambia las rpm del ventilador, la presión estática y el Bhp (punto de la potencia del caballaje del motor) también cambian. Las ecuaciones siguientes, mejor conocidas como leyes de ventilación, muestran la relación entre estos parámetros de operación [7].

$$\begin{aligned} pcm_{nuevo} &= \frac{rpm_{nueva}}{rpm_{anterior}} pcm_{anterior} \\ Pe_{nuevo} &= \left(\frac{rpm_{nueva}}{rpm_{anterior}} \right)^2 Pe_{anterior} \\ Bhp_{nuevo} &= \left(\frac{rpm_{nueva}}{rpm_{anterior}} \right)^3 Bhp_{anterior} \end{aligned}$$

Las primeras dos ecuaciones ya han sido estudiadas en la sección de conceptos del sistema y el ventilador. Es necesario estudiar los ejemplos anteriores para poder entender más a fondo la aplicación de estas ecuaciones.

La tercera ecuación relaciona el caballaje de fuerza, del motor con las rpm del ventilador. El cambio en el caballaje del motor puede ser determinado cuando las rpm son aumentadas alrededor de un 25 %. Esto es mostrado a continuación:

$$Bhp_{nuevo} = (1.25)^3 Bhp_{anterior} = 1.95 Bhp_{anterior}$$

NOTA: Un aumento de un 25% en las rpm resultaría en un 95% de aumento en el caballaje del motor. Tomando esto en consideración, en las primeras selecciones del ventilador, los motores deben, de ser escogidos con un caballaje más grande

de lo requerido, en caso de que surja un aumento en las rpm del ventilador en un futuro.

2.6. Tipos de abanicos

2.6.1. Principio básico

Los abanicos pueden ser clasificados básicamente en dos tipos diferentes: de flujo axial y centrífugos. Los abanicos de flujo axial actúan como propulsores, generando flujo de aire en la dirección del eje del abanico. Los abanicos centrífugos generan corrientes de viento acelerando el flujo de aire radialmente y convirtiendo la energía cinética en una diferencia de presión. Los abanicos de flujo axial y centrífugos tienen capacidades similares en términos de presión, flujo de aire, y eficiencia; sin embargo, usualmente no son intercambiables.

Los puntos clave que determinan cuál tipo de abanico es el más apropiado para una cierta tarea incluyen atributos técnicos y no técnicos. Las consideraciones técnicas incluyen presión, tasa de flujo, eficiencia, restricciones de espacio, generación de ruido, configuración de la unidad, rango de temperatura, variaciones en las condiciones de operación y tolerancia a corrosión o partículas de aire cargadas. Las consideraciones no técnicas incluyen costo, tiempo de entrega, disponibilidad, y familiaridad diseño/operador con un modelo de abanico.

Entender los principios de selección de abanicos puede ser útil en la corrección de un sistema de desempeño pobre, especialmente durante retroadaptación u oportunidades de actualización. Si los niveles de ruido, costos de energía, requisitos de mantenimiento, o desempeño del abanico no cumplen las expectativas, entonces un tipo diferente de abanico necesita ser considerado [16].

2.6.2. Abanicos centrífugos

Los abanicos centrífugos son el tipo de abanicos industriales comúnmente más usados. Los abanicos centrífugos son capaces de generar altas diferencias de presión con altas eficiencias, y pueden ser construidos para adaptarse a condiciones de operación duras. Los abanicos centrífugos tienen varios tipos de formas de aspa, incluyendo aspas curvadas hacia adelante, aspas radiales, aspas de punta radial, aspas inclinadas hacia atrás, aspas curvadas hacia atrás, y aspas aerodinámicas. Algunos tipos de abanicos centrífugos son capaces de servir ampliamente variando las condiciones de operación, lo cual puede representar una ventaja significativa.

Aspas curvadas hacia adelante. Este tipo de abanico, mostrado en la figura 2.7, tiene aspas que se curvan en la dirección de rotación. Este tipo de abanico es usado típicamente en aplicaciones que requieren de bajos a medios volúmenes de aire a baja presión. Se caracterizan por bajas eficiencias relativas (entre 55 y 65 por ciento). Este tipo de abanico puede operar a relativamente bajas velocidades, lo cual se traduce a bajos niveles de ruido. Las abanicos de aspas curvadas hacia adelante son seleccionados comúnmente por su tamaño relativamente pequeño respecto a otros tipos de abanicos.

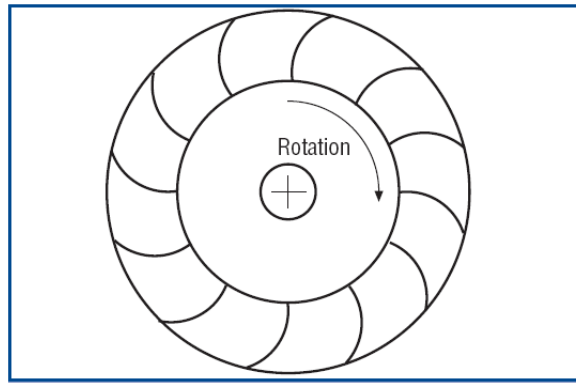


Figura 2.7: Abanico de aspas curvadas hacia adelante [16]

Los niveles de estrés en los abanicos están muy relacionados a la velocidad de operación; consecuentemente, los abanicos de aspas curvadas hacia adelante no requieren atributos de diseño de alta resistencia. Su baja velocidad de operación también hace que sean silenciosos y a la medida para aplicaciones de calefacción residencial, venti-

lación, y aire acondicionado (HAVC). Una curva típica de desempeño se muestra en la figura 2.8. La depresión en la curva de desempeño representa una región crítica que puede crear problemas de operación a bajas tasas de flujo de aire.

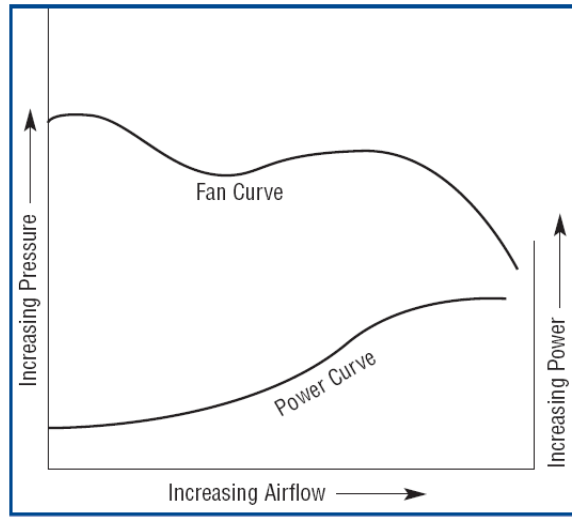


Figura 2.8: Curva de desempeño de un abanico de aspas curvadas hacia adelante [16]

Los abanicos de aspas curvadas hacia adelante regularmente están limitados a aplicaciones de servicios de limpieza. Estos abanicos típicamente no están construidos para altas presiones ni servicio duro. Además, la salida del abanico es difícil de ajustar con precisión (notar como la curva del abanico es en cierto modo horizontal), y estos abanicos no son usados en lugares donde el aire debe ser estrechamente controlado. Los abanicos de aspas curvadas hacia adelante tienen una curva de potencia que se incrementa de manera constante con el flujo del aire hacia la entrega libre; consecuentemente, un controlador de selección es requerido para evitar sobrecarga del motor del abanico.

Aspas radiales. Se muestran en la figura 2.9, este tipo de abanico es usado típicamente en aplicaciones que requieren de bajos a medios volúmenes de aire a altas presiones. La forma plana del asa limita la acumulación de material; consecuentemente, estos abanicos son capaces de manejar corrientes de partículas grandes, incluyendo polvo, virutas de madera, y desechos de metales. Este tipo de abanico es característicamente accidentado. El diseño simple de estos abanicos permite a muchas tiendas de

metalurgia pequeñas el construir pequeñas unidades para aplicaciones especiales. En muchos casos, las hojas pueden ser recubiertas económicamente con componentes protectores para mejorar la resistencia a la erosión y a la corrosión. Los grandes espacios entre las hojas también permiten a estos abanicos operar a bajas corrientes de aire sin los problemas de vibración que usualmente acompañan la operación en la región de depresión. La durabilidad característica de este tipo de abanico es clave de porque es considerado un arduo trabajador de la industria.

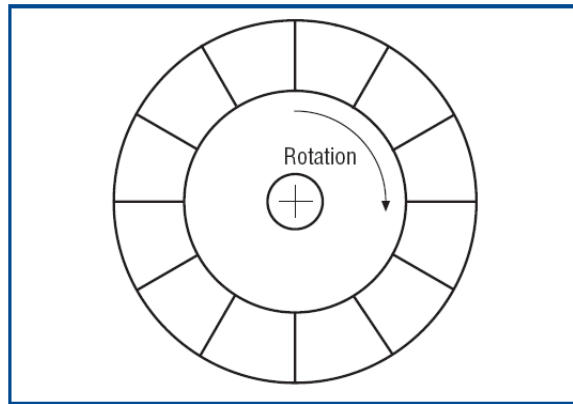


Figura 2.9: Abanico centrífugo de aspas radiales [16]

Aspas de punta radial. Este tipo de abanico llena el espacio entre los abanicos de aire limpio y los más ásperos abanicos de aspa radial. Los abanicos de punta radial se caracterizan por un pequeño ángulo de ataque entre las hojas y el aire entrante, lo cual promueve baja turbulencia. Un abanico de punta radial se muestra en la figura 2.10.

Los abanicos con aspas de punta radial tienen muchas de las características de los abanicos con aspas radiales y son bien adaptados para su uso con flujos de aire que tienen pequeñas partículas en concentraciones moderadas y flujos de aire con alto contenido de humedad. Estos abanicos pueden tener eficiencias por encima del 75 por ciento. Estos abanicos son usados comúnmente en servicios de manejo de transporte de sólidos porque tienen grandes corridas de eliminación de partículas. Una curva típica para abanicos radiales se muestra en la figura 2.11.

Abanicos con aspas inclinadas hacia atrás. Este tipo de abanico se caracte-

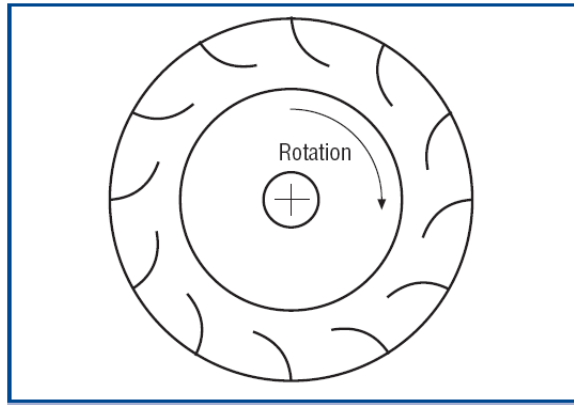


Figura 2.10: Abanico centrífugo con aspas de punta radial [16]

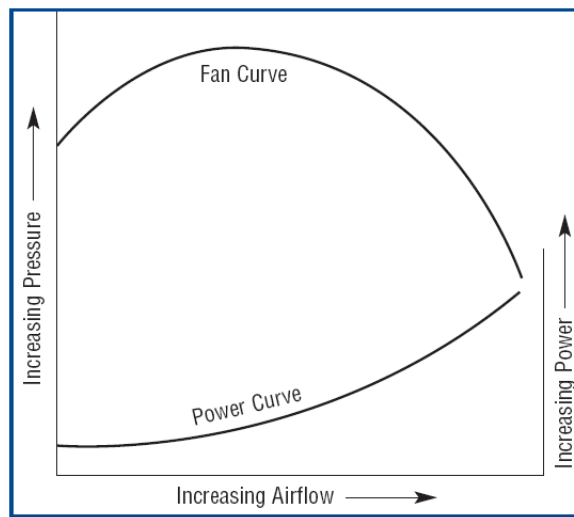


Figura 2.11: Abanico centrífugo con aspas radiales [16]

riza por aspas que se inclinan al contrario de la dirección de rotación. Dentro de los abanicos con aspas inclinadas hacia atrás hay tres diferentes formas de hojas: planas, curvadas, y aerodinámicas. Los de tipo de hoja plana, que se muestran en la figura 2.12, son los más robustos. Los abanicos de hoja curvada tienden a ser más eficientes. Las de hojas aerodinámicas, mostrados en la figura 2.13, son los más eficientes de todos, capaces de alcanzar eficiencias que exceden el 85 por ciento. A causa de que las aspas aerodinámicas dependen del incremento creado por cada aspa, este tipo de abanico es altamente susceptible a la inestabilidad de operación a causa de variaciones en el flujo

de aire.

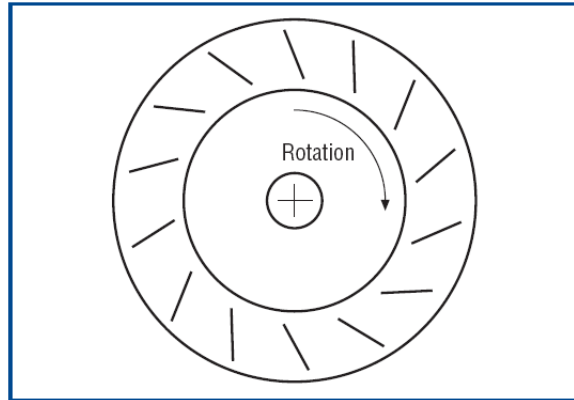


Figura 2.12: Abanico con aspas inclinadas hacia atrás [16]

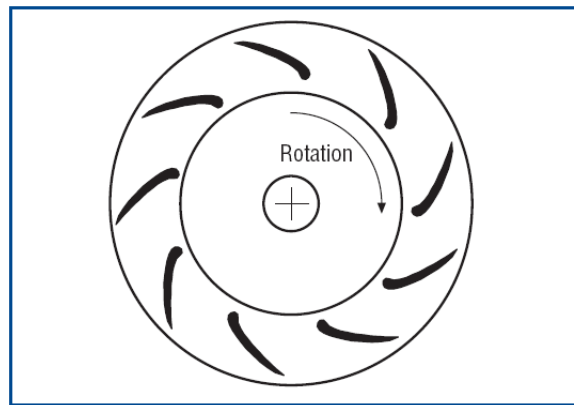


Figura 2.13: Abanico centrífugo con aspas aerodinámicas inclinadas hacia atrás [16]

Una consecuencia de la orientación de las aspas inclinadas hacia atrás es un ángulo pequeño de ataque con el flujo de aire. Esto promueve la acumulación de partículas en las aspas del abanico, lo cual puede crear problemas de desempeño. Las aspas delgadas son más eficientes que los otros tipos a causa de su menor masa rotativa. Sin embargo, este adelgazamiento característico de las paredes hace a este tipo de abanico altamente susceptible a problemas de erosión. El adelgazamiento de las paredes puede llevar a la formación de cavidades en las aspas, lo cual puede interferir severamente con el desempeño del abanico.

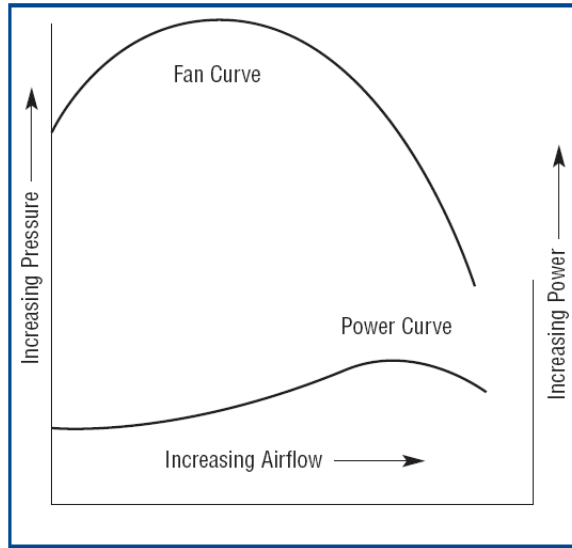


Figura 2.14: Curva del abanico con aspas inclinadas hacia atrás [16]

Una aplicación común para abanicos con aspas inclinadas hacia atrás es el servicio de aspiración mecánica. En estas aplicaciones, el abanico es expuesto al flujo de aire relativamente limpio en la parte alta del flujo. Las altas eficiencias de operación proporcionadas por este tipo de abanico pueden dar sistemas con ciclos de vida de bajos costos. Una curva de desempeño típico se muestra en la figura 2.14. La potencia efectiva del motor se incrementa con el flujo de aire para la mayor parte de la curva de desempeño pero, cae a altas tasas de flujo de aire, a causa de esta característica de no sobrecarga del motor, este tipo de abanico es seleccionado a menudo cuando el comportamiento del sistema a tasas altas de flujo de aire es incierto.

2.6.3. Abanicos de flujo axial

Las ventajas clave de los abanicos con flujo axial son tamaño compacto, bajo costo, y bajo peso. Los abanicos con flujo axial se usan frecuentemente en aplicaciones de escape donde el tamaño de las partículas arrastradas por el aire es pequeño, tales como polvo, humo, y vapor. Los abanicos de flujo axial son también útiles en aplicaciones de ventilación que requieren la habilidad para generar corrientes de aire en sentido contrario. Aunque los abanicos están diseñados típicamente para generar el flujo en

un solo sentido, éstos pueden operar en el sentido contrario. Esta característica es útil cuando un lugar puede requerir que el aire contaminado sea eliminado o aire fresco sea suplido.

Los abanicos de flujo axial tienen una severa región crítica que los hace particularmente inadecuados para sistemas con una amplia variedad de condiciones de operación. En esta región crítica, el flujo de aire es insuficiente para llenar las aspas, causando que el abanico opere en forma inestable. Las consecuencias de la operación inestable incluyen molestos patrones de ruido, desempeño ineficiente, y comportamiento acelerado de la maquinaria. Este problema de mal funcionamiento puede ser resuelto en muchos abanicos de flujo axial seleccionando un abanico con un mecanismo regulador. Este mecanismo altera los patrones de flujo de aire alrededor de las aspas, permitiendo la estabilidad de operación en el rango completo de flujo de aire y presión.

Los abanicos de flujo axial deben rotar más rápido que sus abanicos centrífugos equivalentes para lograr la misma capacidad de flujo de aire. Esta característica los hace más ruidosos que sus abanicos centrífugos equivalentes; sin embargo, la señal de ruido está dominada por las altas frecuencias, las cuales son fáciles de atenuar [16].

Abanicos helicoidales. La versión más simple de un abanico de flujo axial es el tipo helicoidal, que se muestra en la figura 2.15.

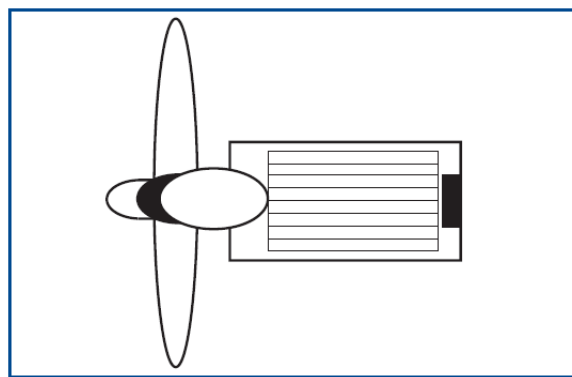


Figura 2.15: Abanico helicoidal [16]

Los abanicos helicoidales generan altas tasas de flujo de aire a bajas presiones. A causa de que los abanicos helicoidales no generan mucha presión, usualmente no

se combinan con muchos trabajos en tubos. Los abanicos helicoidales tienden a tener eficiencias relativamente bajas, pero no son caros gracias a su construcción simple. Los abanicos helicoidales tienden a ser comparablemente ruidosos, reflejando su operación ineficiente.

Como se muestra en la figura 2.16, los requerimientos de potencia de los abanicos helicoidales decrece con el incremento del flujo de aire. Estos alcanzan la máxima eficiencia, cerca de la descarga libre, y son usados a menudo en aplicaciones de ventilación en lo alto del techo.

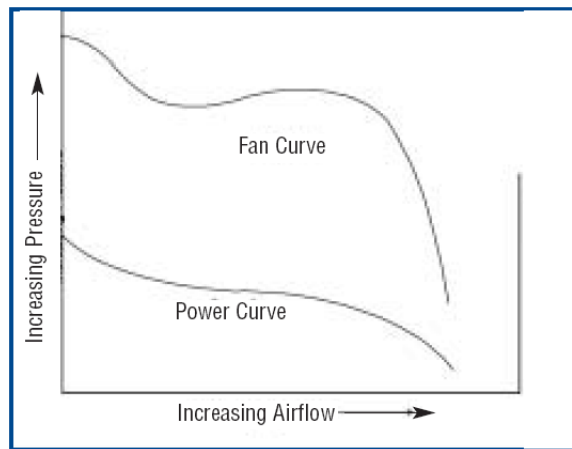


Figura 2.16: Curva del abanico helicoidal [16]

Abanicos tuboaxiales. Una versión más compleja de un abanico helicoidal es el abanico tuboaxial. Este tipo, que se muestra en la figura 2.17, es esencialmente un abanico helicoidal colocado en el interior de un cilindro. Mejorando las características del flujo de aire, los abanicos tuboaxiales alcanzan altas presiones y mejores eficiencias de operación que los abanicos helicoidales.

Los abanicos tuboaxiales son usados en presión media, tasas altas de flujo de aire y son adecuados para instalaciones HVAC entubadas. El perfil del flujo de aire hacia abajo es disperejo, con una gran componente rotacional. Este flujo de aire característico es acompañado por ruido de flujo de aire moderado. Los abanicos tuboaxiales son usados frecuentemente en aplicaciones de escape porque crean suficiente presión para superar pérdidas en tubos y son relativamente eficientes en espacio. También, a causa de su

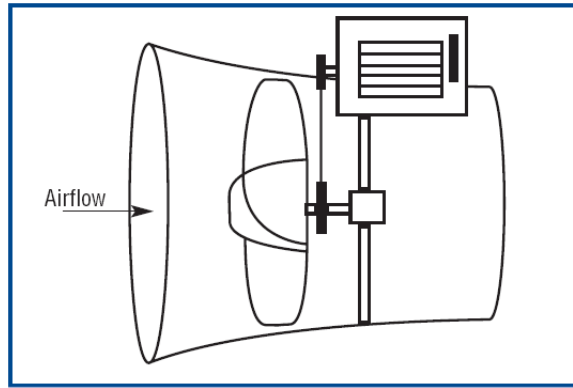


Figura 2.17: Abanico tuboaxial [16]

baja masa rotativa, pueden acelerar rápidamente a velocidad elevada, lo cual es útil en muchas aplicaciones de ventilación.

La curva de desempeño para abanicos tuboaxiales se muestra en la figura 2.18. Muy parecidos a los abanicos helicoidales, los abanicos tuboaxiales tienen una región de inestabilidad pronunciada que debe ser evitada.

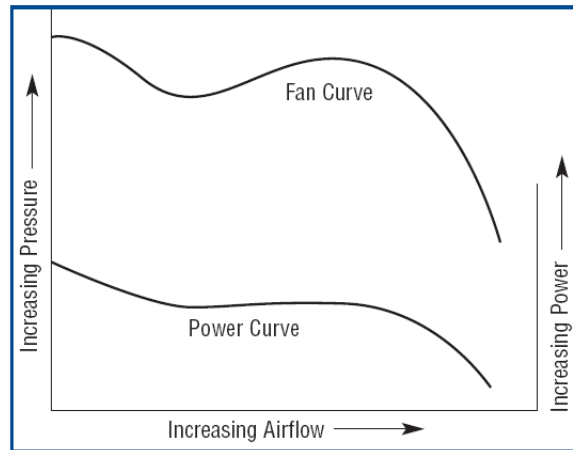


Figura 2.18: Curva del abanico tuboaxial [16]

Los abanicos tuboaxiales pueden ser conectados directamente a un motor o manejados por una configuración de banda. A causa de las altas velocidades de operación de los motores de 2-, 4-, y 6-tiempos, la mayoría de los abanicos tuboaxiales usa bandas de operación para alcanzar velocidades debajo de 1100 revoluciones por minuto.

Abanicos tipo vaneaxial. Un refinamiento del abanico de flujo axial es el aba-

nico tipo vaneaxial. Como se muestra en la figura 2.19, un abanico tipo vaneaxial es esencialmente un abanico tuboaxial con veletas de salida que mejoran el patrón de flujo de aire, convirtiendo la energía cinética del flujo de aire en presión. Estas veletas crean un perfil de corriente de aire que es comparativamente uniforme.

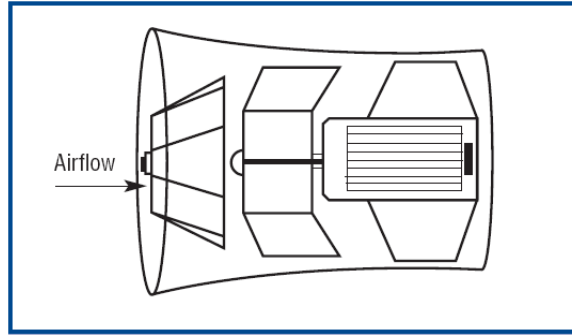


Figura 2.19: Abanico tipo vaneaxial [16]

Los abanicos tipo vaneaxial son usados típicamente en aplicaciones de mediana a alta presión, tales como servicio de corrientes de aire inducidas para escape de calderas. Similar a los abanicos tuboaxiales, los abanicos tipo vaneaxial tienden a tener masa de rotación baja, lo cual les permite alcanzar velocidades de operación relativamente altas. Esta característica es útil en aplicaciones de ventilación de emergencia donde la rápida eliminación del aire o su suplemento es requerido. También, parecido a otros abanicos de flujo axial, los abanicos tipo vaneaxial pueden generar corriente en el sentido opuesto, lo cual es también útil en aplicaciones de ventilación. Dependiendo de las circunstancias, estas aplicaciones pueden requerir el suplemento de aire fresco o la remoción del aire contaminado.

Los abanicos tipo vaneaxial están a menudo equipados con aspas de ángulo variable, las cuales pueden ser ajustadas para enfrentar el ángulo de ataque del flujo de aire. Las aspas de ángulo variable pueden cambiar la carga del abanico, proporcionando un método efectivo y eficiente de control del flujo de aire.

Como se muestra en la figura 2.20, los abanicos de tipo vaneaxial tienen curvas de desempeño que tienen regiones inestables a la izquierda del pico de presión. Estos abanicos son altamente eficientes. Cuando están equipados con aspas aerodinámicas

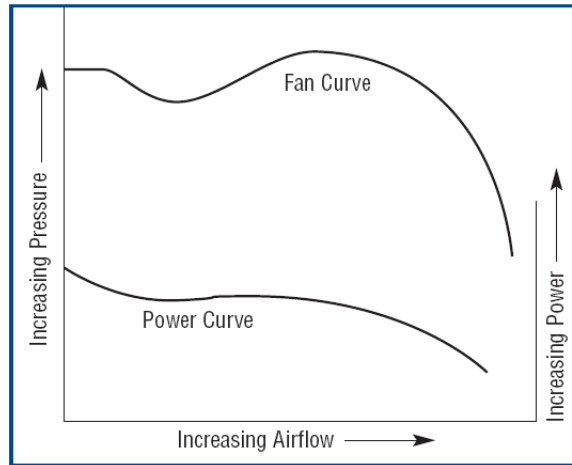


Figura 2.20: Curva del abanico tipo vaneaxial [16]

y construidas con pequeños claros, pueden alcanzar eficiencias por encima del 85 por ciento. Los abanicos de tipo vaneaxial están frecuentemente conectados al eje del motor.

2.7. Resumen del capítulo

En esta sección se habló acerca de los ventiladores. Se describieron los conceptos del ventilador y del sistema, después se combinó ambos conceptos. Se habló de como hacer el ajuste del funcionamiento de un ventilador. Se describieron las principales leyes de ventilación. Finalmente se describieron los abanicos centrífugos y de tipo axial. Se mostraron ejemplos de aplicación práctica y en general se muestra de manera gráfica las relaciones que nos permiten entender mejor el funcionamiento de los ventiladores.

Capítulo 3

Fundamentos teóricos de algoritmos genéticos

En este capítulo se hará una introducción a los algoritmos genéticos, se mencionaran sus principales características como son: creación de una población inicial, evaluación de la población inicial, selección de padres, cruce de los padres, mutación sufrida por el hijo o los hijos resultantes, evaluación de los hijos, reintegración de los hijos en la población y si no se cumple una condición de paro se continúa con un ciclo establecido. Se hablará también de la importancia de la codificación de los cromosomas y de la similitud que existe entre los algoritmos genéticos y la naturaleza. Se finalizará con una breve descripción del algoritmo genitor, el cual fue usado en nuestra investigación con algunas variaciones pequeñas.

3.1. Introducción a los algoritmos genéticos

Un algoritmo genético (AG) es una técnica heurística que imita a la evolución biológica como estrategia para resolver problemas. Inicialmente desarrollada por John Holland (1975 [8]) y por los estudiantes de Holland (por ejemplo DeJong, 1975 [5]). Dado un problema específico a resolver, la entrada del AG es un conjunto de soluciones potenciales a ese problema, codificadas de alguna manera, y una métrica llamada función de aptitud que permite evaluar cuantitativamente a cada candidata. Estas candidatas pueden ser soluciones que ya se sabe factibles, con el objetivo de que el AG las mejore, pero se suelen generar aleatoriamente.

Los algoritmos genéticos son algoritmos de búsqueda basados en los mecanismos de

la selección natural que combinan la supervivencia de las secuencias mejores adaptadas con cambios aleatorios de información.

Los algoritmos genéticos pueden verse como una familia de procedimientos de búsqueda adaptativos. Su nombre se deriva de que están basados en modelos de cambio genético en una población de individuos o cromosomas. Esto es:

- Noción Darwiniana de aptitud (fitness) que influye en generaciones futuras.
- Apareamiento que produce descendientes en generaciones futuras.
- Operadores genéticos que determinan la configuración genética de los descendientes.

Un punto clave de estos modelos, es que el proceso de adaptación no se hace cambiando incrementalmente una sola estructura, sino manteniendo una población de estructuras a partir de las cuales se generan nuevas estructuras usando los operadores genéticos.

Cada estructura en la población está asociada con una aptitud y los valores se usan en competencia para determinar qué estructuras serán usadas para formar nuevas estructuras.

Una de sus características es su habilidad de explotar información acumulada acerca de un espacio de búsqueda inicialmente desconocido para guiar la búsqueda subsecuente a subespacios útiles. Su aplicación está enfocada sobre todo a espacios de búsqueda grandes, complejos y poco entendidos. El precio es que se pueden necesitar un número grande de muestras para que se tenga suficiente información para guiar muestras subsecuentes a subespacios útiles.

En su forma más simple, un algoritmo genético está orientado hacia el desempeño (por ejemplo, hacer cambios estructurales para mejorar el desempeño).

Una de las ideas más importantes es definir estructuras admisibles en el sentido que estén bien definidas y puedan ser evaluadas. Es decir, que tales estructuras codifiquen soluciones factibles.

Los algoritmos genéticos difieren con los métodos tradicionales de búsqueda y optimización en los siguientes aspectos:

- Trabajan con un conjunto de parámetros codificados y no con los parámetros mismos.
- Inician la búsqueda desde un conjunto de puntos, no de uno solo.
- Usan una función a optimizar en lugar de la derivada u otro conocimiento adicional.
- Usan reglas de transición probabilísticas no determinísticas.

3.2. Anatomía de un algoritmo genético

Un algoritmo genético esta compuesto por:

- **Módulo evolutivo:** mecanismo de decodificación (interpreta la información de un cromosoma) y función de evaluación (mide la calidad del cromosoma). Sólo aquí existe información del dominio.
- **Módulo poblacional:** tiene una representación poblacional y técnicas para manipularla (técnica de representación, técnica de arranque, criterio de selección y de reemplazo). Aquí también se define el tamaño de la población y la condición de terminación.
- **Módulo reproductivo:** contiene los operadores genéticos.

3.3. Características de los algoritmos genéticos

Algunas de las características de los algoritmos genéticos son:

- Son algoritmos estocásticos. Es decir, dos ejecuciones distintas pueden dar dos soluciones distintas.

- Son algoritmos de búsqueda múltiple, por lo tanto dan varias soluciones. Aunque habitualmente las evaluaciones de los individuos de la población final es similar, los individuos suelen ser distintos entre sí.
- Son los algoritmos que hacen una barrida amplia al subespacio de posibles soluciones válidas. De hecho, se considera que, de todos los algoritmos de optimización estocásticos, los algoritmos genéticos son de los más exploratorios disponibles [12].
- La convergencia del algoritmo es poco sensible a la población inicial si ésta se escoge de forma aleatoria y es lo suficientemente grande.
- La curva de convergencia asociada al algoritmo presenta una convergencia excepcionalmente rápida al principio, que casi enseguida se bloquea. Esto se debe a que el algoritmo genético es excelente descartando subespacios realmente malos. Cada cierto tiempo, la población vuelve dar el salto evolutivo, y se produce un incremento en la velocidad de convergencia excepcional. La razón de esto es que algunas veces aparece una mutación altamente beneficiosa, o un individuo excepcional, que propaga algún conjunto de cromosomas excepcional al resto de la población.
- La optimización es función de la representación de los datos. Una buena codificación puede hacer la programación y resolución muy sencillas, mientras que una codificación errada obligará a estudiar que los individuos cumplan las restricciones del problema. Además, la velocidad de convergencia va a estar fuertemente influenciada por la representación.
- Es una búsqueda paramétricamente robusta. Eso quiere decir que los parámetros del algoritmo escogidos deben ser realmente malos para que no converja.
- Los algoritmos genéticos son intrínsecamente paralelos. Esto significa que, independientemente de que se hayan implementado de forma paralela o no, buscan en distintos puntos del espacio de soluciones de forma simultánea.

3.4. Analogía de un algoritmo genético con la naturaleza

La idea básica de un algoritmo genético es la siguiente: generar un conjunto con algunas de las posibles soluciones. Cada una va a ser llamada individuo, y a dicho conjunto se le denominará población.

Cada individuo tiene una información asociada a él. En un problema de optimización corresponde a las variables libres, es decir, aquéllas a las que el algoritmo tiene que asignar un valor para que una función sea mínima o máxima para esos valores. Esa función es la denominada función de aptitud y determina el grado de adaptación de un individuo. A dicha información se la va a denominar código genético.

Las características de los individuos, sean beneficiosas o no, se van a denominar fenotipos. La información asociada a un individuo se compone de partes indivisibles denominadas cromosomas.

Un fenotipo puede estar asociado a más de un cromosoma, en cuyo caso puede ser que el hijo herede un fenotipo que no tenía ni el padre ni la madre, sino una combinación de ambos. Un ejemplo en el humano es el color de la piel o la estructura del cráneo. En caso de que el hijo tenga parte de los genes del padre y parte de los genes de la madre que intervienen en un fenotipo, se va a crear una característica nueva asociada a ese fenotipo.

De todas formas, no es un enfoque muy frecuente, ya que debemos asegurar que el conjunto de los fenomas tendrá ley de composición interna respecto al operador de cruce definido sobre el alfabeto cromosómico. Por otro lado, que un cromosoma codifique más de un fenotipo es más raro todavía. El cromosoma debe tener en dicho caso tantos valores como el producto del número de valores posibles que tenga cada fenotipo del cromosoma.

Es de vital importancia la forma de codificar los fenotipos en los cromosomas y la determinación de qué es fenotipo, es decir, como la información va a ser almacenada en el código genético. Escoger equivocadamente la forma de almacenar la información

puede lentificar la convergencia, es decir, que se tardará más en encontrar la solución o nunca convergerá en una solución debido a que la población está errando aleatoriamente por efecto de las mutaciones y de los cruzamientos sin llegar nunca a un punto estable, a este fenómeno se le denomina deriva genética.

3.5. Codificación de las variables

Los algoritmos genéticos requieren que el conjunto de parámetros de los que depende un problema se codifique en un cromosoma. Cada cromosoma tiene varios genes, que corresponden a parámetros del problema. Para poder trabajar con estos genes en un programa computacional, es necesario codificarlos en una cadena de símbolos (números o letras) que generalmente va a estar compuesta de ceros y unos.

Los tres métodos principales de codificación de variables son:

- Cadenas binarias, secuencias de unos y ceros donde el dígito de cada posición representa el valor de algún aspecto de la solución.
- Cadenas de enteros o números decimales donde cada posición representa algún aspecto particular de la solución.
- Cadenas de letras donde cada letra representa un aspecto específico de la solución.

El método elegido para codificar las variables debe facilitar la definición de operadores que causen los cambios aleatorios en los individuos seleccionados: cambiar un 0 por un 1 o viceversa, sumar o restar al valor de un número una cantidad elegida al azar, o cambiar una letra por otra.

Otra estrategia, desarrollada principalmente por John Koza, de la Universidad de Stanford, y denominada programación genética, representa a los programas informáticos como estructuras de datos ramificadas llamadas árboles [9, p. 35]. En este método, los cambios aleatorios pueden generarse cambiando el operador o alterando el valor de un cierto nodo del árbol, o sustituyendo un subárbol por otro.

La elección de la codificación de las variables no es trivial, especialmente si estamos tratando un problema de optimización. Por ejemplo, una codificación directa de los números enteros puede dar problemas a la hora de que el algoritmo converja, ya que números consecutivos, como por ejemplo el 15 y el 16, al pasarlos a binario son muy diferentes (10000, 01111) con lo que una solución con el valor 15 en un campo, difícilmente llegará a evolucionar a una solución con el valor 16 (ya que debería cambiar simultáneamente todos los bits). A este problema se le conoce como “Picos de Hamming”.

Para disminuir el efecto de los “Picos de Hamming” se pueden utilizar otras codificaciones. Por ejemplo, codificando según los “Códigos de Gray” asegura que enteros consecutivos sólo se diferencien en un bit.

3.6. Evaluación y selección

Durante la evaluación se decodifica el cromosoma y se convierte en una serie de parámetros de un problema. Luego se halla la solución del problema a partir de esos parámetros y se le da una puntuación en función de lo cerca que esté de la mejor solución. A esta puntuación se le llama *fitness* o evaluación.

El siguiente ejemplo ilustra el proceso de evaluación de un algoritmo genético. Se quiere hallar el máximo de una parábola invertida con el máximo en $x = 1$. El único parámetro del problema es la variable x . La optimización consiste en hallar un x tal que $F(x)$ sea máximo. Se crea entonces una población de cromosomas, cada uno de los cuales contiene una codificación binaria del parámetro x . Cada byte, cuyo valor está comprendido entre 0 y 255, se transformará para ajustarse al intervalo $[-1,1]$, donde el objetivo es hallar el máximo de la función. La tabla 3.1 muestra los resultados obtenidos para cuatro cromosomas.

La aptitud determina siempre los cromosomas que se van a reproducir, y aquéllos que se van a eliminar.

Valor binario	Decodificación	Evaluación $f(x)$
10010100	21	0.9559
10010001	19	0.9639
00101001	-86	0.2604
01000101	-58	0.6636

Cuadro 3.1: Ejemplo de resultados obtenidos con una función de evaluación

3.6.1. Métodos de selección

Entre las técnicas que un algoritmo genético puede utilizar para seleccionar a los individuos que deben copiarse hacia la siguiente generación están:

- **Selección elitista:** se garantiza la selección de los miembros más aptos de cada generación. La mayoría de los algoritmos genéticos no utilizan elitismo puro, sino que usan una forma modificada por la que el individuo mejor, o algunos de los mejores, son copiados hacia la siguiente generación en caso de que no surja nada mejor.
- **Selección proporcional a la aptitud:** los individuos más aptos tienen más probabilidad de ser seleccionados, pero no la certeza.
- **Selección por rueda de ruleta:** una forma de selección proporcional a la aptitud en la que la probabilidad de que un individuo sea seleccionado es proporcional a la diferencia entre su aptitud y la de sus competidores.
- **Selección escalada:** al incrementarse la aptitud media de la población, la fuerza de la presión selectiva también aumenta y la función de aptitud se hace más discriminadora. Este método puede ser útil para seleccionar más tarde, cuando todos los individuos tengan una aptitud relativamente alta y sólo les distinguen pequeñas diferencias en la aptitud.
- **Selección por torneo:** se eligen subgrupos de individuos de la población, y los

miembros de cada subgrupo compiten entre ellos. Sólo se elige a un individuo de cada subgrupo para la reproducción.

- **Selección por estado estacionario:** la descendencia de los individuos seleccionados en cada generación vuelven al acervo genético preexistente, reemplazando a algunos de los miembros menos aptos de la siguiente generación. Se conservan algunos individuos entre generaciones.
- **Selección por rango:** a cada individuo de la población se le asigna un rango numérico basado en su aptitud, y la selección se basa en este rango, en lugar de las diferencias absolutas en aptitud. La ventaja de este método es que puede evitar que individuos muy aptos ganen dominancia al principio a expensas de los menos aptos, lo que reduciría la diversidad genética de la población y podría obstaculizar la búsqueda de una solución aceptable.
- **Selección generacional:** la descendencia de los individuos seleccionados en cada generación se convierte en toda la siguiente generación. No se conservan individuos entre las generaciones.
- **Selección jerárquica:** los individuos atraviesan múltiples rondas de selección en cada generación. Las evaluaciones de los primeros niveles son más rápidas y menos discriminatorias, mientras que los que sobreviven hasta niveles más altos son evaluados más rigurosamente. La ventaja de este método es que reduce el tiempo total de cálculo al utilizar una evaluación más rápida y menos selectiva para eliminar a la mayoría de los individuos que se muestran poco o nada prometedores, y sometiendo a una evaluación de aptitud más rigurosa y computacionalmente más costosa sólo a los que sobreviven a esta prueba inicial.

Algunos de estos métodos son mutuamente exclusivos, pero otros pueden utilizarse en combinación, algo que se hace a menudo.

3.6.1.1. Hacinamiento determinista

El hacinamiento determinista ó DC por sus siglas en inglés (*Deterministic Crowding*) se basa en el principio de competición estricta. Cualquier individuo puede ser recombinado con otro pero únicamente el hijo que más se asemeje al padre puede reemplazarlo [20].

De esta manera los individuos tienden a competir entre sí con otros que tienen el mismo puntaje obteniendo varios óptimos locales. Esta característica es útil para prevenir una convergencia prematura ya que explora diferentes espacios en paralelo.

Otra ventaja es que existe un mecanismo elitista inherente en el método. Los algoritmos genéticos que no tienen una buena diversidad genética convergen rápidamente en un óptimo local o no pueden mantener un óptimo global durante la evolución.

Un algoritmo genético con hacinamiento determinista estándar trabaja de la siguiente manera:

1. Inicializar la población aleatoriamente.
2. Aleatoriamente formar parejas de individuos.
3. Después de la recombinación, dos padres producen dos hijos.
4. De acuerdo a una regla de paridad, cada hijo es comparado con un padre.
5. Si el puntaje del hijo es superior que el del padre, el padre es reemplazado.
6. Ir al paso 2 hasta que se satisfaga el criterio de parada.

Cada hijo es comparado con el padre con que tiene mayor similitud. Si los dos padres son idénticos, cada hijo es comparado con un padre. Si los dos hijos son más similares a un padre y ambos puntajes son superiores a los del padre, el hijo más similar al padre gana.

Para medir la similitud entre un hijo y un padre se emplea la distancia de *Hamming*. Si dos listas de elementos son comparadas, la distancia de *Hamming* esta dada por el número de elementos que no son iguales [4].

3.7. Funcionamiento de un algoritmo genético

Luego de haber codificado las variables y haber definido el método de selección, el algoritmo genético procede de la siguiente manera:

1. Se generan aleatoriamente una serie de cromosomas.
2. Se evalúa la puntuación de cada uno de los individuos.
3. Se permite a cada uno de los individuos reproducirse de acuerdo con su puntuación.
4. Se empareja a los individuos de la nueva población, haciendo que intercambien material genético y que alguno de los bits de un gen se vea alterado debido a una mutación espontánea.

Un algoritmo genético también tiene parámetros que se tienen que fijar para cada ejecución:

- Tamaño de la población: debe de ser suficiente para garantizar la diversidad de las soluciones y tiene que crecer más o menos con el número de bits del cromosoma.
- Condición de terminación: lo más habitual es que la condición de terminación sea la convergencia del algoritmo genético o un número prefijado de generaciones.

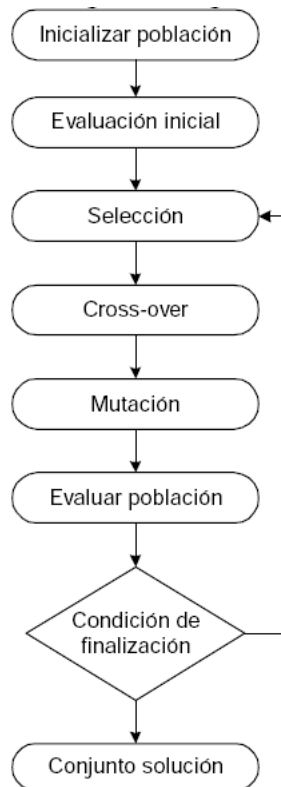
El procedimiento que sigue un algoritmo genético puede resumirse en el algoritmo de la figura 3.1.

Figura 3.1: Procedimiento de un algoritmo genético [12]

```
Procedimiento AG
Tiempo = 0
Inicializa Población (tiempo)
Evalúa Población (tiempo)
Mientras no condición de terminación
    Tiempo = tiempo + 1
    Construye Población (tiempo)
    a partir de población (tiempo - 1) usando:
    o Selección
    o Modifica Población (tiempo) usando Operadores Genéticos
    o Evalúa Población (tiempo)
    o Reemplaza
Fin Mientras
```

Un posible esquema para representar la implementación de un algoritmo genético se muestra en la figura 3.2.

Figura 3.2: Diagrama de funcionamiento de una posible implementación de algoritmo genético [12]



3.8. Operadores genéticos

3.8.1. Crossover (cruce)

Es el principal operador genético y consiste en el intercambio de material genético entre dos cromosomas.

Para aplicar el cruce, entrecruzamiento o recombinación, se escogen aleatoriamente dos miembros de la población. No pasa nada si se emparejan dos descendientes de los mismos padres, ello garantiza la perpetuación de un individuo con buena puntuación. Sin embargo, si esto sucede demasiado a menudo, puede crear problemas: toda

la población puede aparecer dominada por los descendientes de algún individuo que puede tener características no deseadas. Esto se suele denominar en otros métodos de optimización atranque en un mínimo local, y es uno de los principales problemas con los que se enfrentan los que aplican algoritmos genéticos.

El cruce se usa en conjunto con el teorema de los esquemas el cual se basa en la noción de bloques de construcción, así una buena solución a un problema está constituida por buenos bloques. El cruce es el encargado de mezclar bloques buenos que se encuentren en los diversos progenitores. La presión selectiva se encarga de que sólo los buenos bloques se perpetúen y poco a poco vayan formando una buena solución. El teorema de los esquemas viene a decir que la cantidad de buenos bloques se va incrementando con el tiempo de ejecución de un algoritmo genético, y es el resultado teórico más importante en algoritmos genéticos.

El intercambio genético se puede llevar a cabo de muchas formas:

- **Crossover en n puntos:** los dos cromosomas se cortan por n puntos, y el material genético situado entre ellos se intercambia. Lo más habitual es un *crossover* de un punto o de dos puntos. Un ejemplo de *crossover* de dos puntos se muestra en la figura 3.3.

Figura 3.3: *Crossover* de n puntos

Padre	0	0	0	1	0	1	0	1	0	1	0	1	0	1
Madre	1	0	1	1	1	0	0	1	1	1	0	1	1	1
Hijo	0	0	0	1	0	0	0	1	1	1	0	1	0	1

- **Crossover uniforme:** se genera un patrón aleatorio de 1s y 0s, y se intercambian los bits de los dos cromosomas que coincidan donde hay un 1 en el patrón. O bien se genera un número aleatorio para cada bit, y si supera una determinada

probabilidad se intercambia ese bit entre los dos cromosomas. Un ejemplo de *crossover* uniforme se muestra en la figura 3.4, donde se ha tomado un bit del padre cuando el valor del patrón es cero y un bit de la madre cuando el valor del patrón es uno.

Figura 3.4: *Crossover* uniforme

Padre	0	0	0	1	0	1	0	1	0	1	0	1	0	1
Madre	1	0	1	1	1	0	0	1	1	1	0	1	1	1
Patrón	0	0	1	0	0	1	0	0	1	0	0	1	1	0
Hijo	0	0	1	1	0	0	0	1	1	1	0	1	1	1

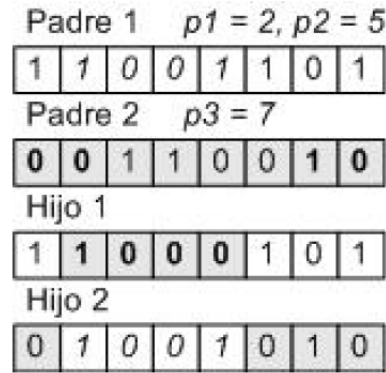
- ***Crossover* especializados:** en algunos problemas, aplicar aleatoriamente el *crossover* da lugar a cromosomas que codifican soluciones inválidas, en este caso hay que aplicar el *crossover* de forma que genere siempre soluciones válidas.

Existe una extensión del *crossover* de dos puntos en el cual los puntos de cruce pueden ser diferentes en los padres, este es el *crossover* asimétrico de dos puntos ó ATC por sus siglas en inglés (*asymmetric two-point crossover*). El ATC funciona de la siguiente manera:

- Seleccionar dos puntos de cruce p1 y p2 en el padre.
- Seleccionar un punto de cruce p3 en la madre.
- Crear el primer hijo: Reemplazar los genes entre p1 y p2 en el padre con los genes de la madre iniciando en p3.
- Crear el segundo hijo: Reemplazar los genes entre p1 y p2 en la madre con aquellos entre p1 y p2 en el padre.

Cada individuo padre es tratado como una lista circular de genes, como se muestra en la figura 3.5 al llegar al último gen en la madre se continúa en el inicio [20].

Figura 3.5: *Crossover* de dos puntos asimétrico



3.8.2. Mutación

En la evolución una mutación es un suceso poco común (sucede aproximadamente una de cada mil replicaciones). En la mayoría de los casos las mutaciones son letales, pero en promedio contribuyen a la diversidad genética de la especie. En un algoritmo genético tendrán el mismo papel y la misma frecuencia (muy baja).

Una vez establecida la frecuencia de mutación, por ejemplo uno por mil, se examina cada bit de cada cadena cuando se vaya a crear el nuevo individuo a partir de sus padres (normalmente de forma simultánea al *crossover*), si un número generado aleatoriamente está por debajo de la frecuencia de mutación, se cambiará el bit (de 0 a 1 o de 1 a 0), si no se dejará como está. Dependiendo del número de individuos que haya y del número de bits por individuo, puede resultar que las mutaciones sean extremadamente raras en una sola generación.

No conviene abusar de la mutación, es cierto que es un mecanismo generador de diversidad y por tanto, la solución cuando un algoritmo genético está estancado, pero también es cierto que reduce el algoritmo genético a una búsqueda aleatoria. Siempre es más conveniente usar otros mecanismos de generación de diversidad, como aumentar el tamaño de la población o garantizar la aleatoriedad de la población inicial. El operador

mutación junto con el *crossover* y el método de selección de ruleta constituyen un algoritmo genético simple, SGA, introducido por Goldberg en su libro [6].

3.9. Genitor

El algoritmo Genitor (Whitley 1988: 1989 [18]) fue el primero de los cuales Syswerda (1989 [15]) ha llamado algoritmos genéticos de “estado estable”. El nombre “estado estable” es de alguna forma erróneo, debido a que este algoritmo muestra más variación que los algoritmos genéticos ortodoxos en términos del comportamiento de hiperplanos (Syswerda, 1991 [14, p. 94-101]) y por lo tanto es más susceptible a error de muestreo y deriva genética. La ventaja es que los mejores individuos encontrados en la búsqueda se mantienen en la población. Esto resulta en una mayor agresividad de búsqueda que en la práctica es a menudo bastante efectiva.

Hay tres diferencias entre algoritmos tipo Genitor y algoritmos genéticos ortodoxos. Primero, la reproducción produce un descendiente a la vez. Dos padres son seleccionados para reproducción y producen un descendiente que inmediatamente es puesto de regreso en la población. La segunda gran diferencia es como los descendientes son puestos de regreso en la población. Los descendientes no reemplazan a los padres, sino al miembro menos apto de la población. En el Genitor, el peor individuo de la población es reemplazado. La tercera diferencia entre el Genitor y la mayoría de otras formas de algoritmos genéticos es que la aptitud es asignada de acuerdo a ranqueo antes que por rueda de ruleta. El ranqueo ayuda a mantener una presión selectiva más constante sobre el curso de la búsqueda.

Goldberg y Deb (1991 [14, p. 69-93]) han demostrado que reemplazando el peor miembro de la población genera una presión selectiva más alta que el reemplazo aleatorio. Pero una presión selectiva más grande no es la única diferencia entre el Genitor y el algoritmo genético ortodoxo. Usando terminología usada por la comunidad de Estrategia Evolutiva (como sugiere Larry Eshelman), el genitor es una estrategia $(\mu + \lambda)$ mientras que el algoritmo genético ortodoxo es una estrategia (μ, λ) . Así, la acumulación

de cadenas mejoradas en la población es monotónica [19].

3.10. Resumen del capítulo

En este capítulo se dio una introducción a los algoritmos genéticos, se mencionaron sus principales características como son: creación de una población inicial, evaluación de la población inicial, selección de padres, cruce de los padres, mutación sufrida por el hijo o los hijos, evaluación de los hijos, reintegración de los hijos en la población y si no se cumple una condición de paro se continúa con un ciclo establecido. Se habló también de la importancia de la codificación de los cromosomas y de la similitud que existe entre los algoritmos genéticos y la naturaleza. Se finalizó con una breve descripción del algoritmo genitor, el cual fue usado en nuestra investigación con algunas variaciones pequeñas.

Capítulo 4

Fundamentos teóricos de fluidos computacionales

Dinámica de fluidos computacionales, o CFD, es la simulación numérica de movimiento de fluido. Mientras que el movimiento de fluidos en una mezcla es una aplicación obvia de CFD, hay cientos de otras, que van desde flujo de sangre en las arterias, fluido supersónico sobre un plano aerodinámico, hasta el modelado de caucho en la manufactura de partes automotrices. Numerosos modelos y técnicas de solución han sido desarrolladas a lo largo de los años para ayudar a describir una gran variedad de movimientos de fluido [11]. En la siguiente sección se presentan las ecuaciones fundamentales para el flujo de fluidos.

Este capítulo es importante, ya que los modelos empleados por *OpenFOAM* se basan en las teorías aquí examinadas. Aunque actualmente *OpenFOAM* puede utilizar nuevos modelos, es recomendable utilizar en un principio los ya reconocidos. Los modelos aquí descritos no son exhaustivos, sino sólo una recopilación de los que consideramos más importantes. El algoritmo SIMPLE y el modelo de corte rotativo descritos en el presente capítulo son en especial importantes ya que el solver utilizado por *OpenFOAM* recibe el nombre de *simpleSRFFoam* por utilizar ambos (SRF viene del inglés *single reference frame*). Por ahora nos concentraremos en la dinámica de los fluidos computacionales y dejaremos la descripción de *OpenFOAM* para el siguiente capítulo.

4.1. Ecuaciones de conservación

Si un volumen pequeño, o elemento de fluido en movimiento es considerado, principalmente dos cambios del elemento tendrán lugar. Primero, el elemento de fluido se trasladará y posiblemente rotará en el espacio, y segundo, se deformará, ya sea por un simple alargamiento a lo largo de uno o más ejes, o por una distorsión angular que causa que cambie de forma. El proceso de translación a menudo se refiere a una convección, mientras que el proceso de distorsión se relaciona con la presencia de gradientes en el campo de velocidad y un proceso llamado difusión. En el caso más simple, estos procesos gobiernan la evolución del fluido de un estado a otro. En los sistemas más complicados, puede haber fuentes que incrementen o proporcionen cambios adicionales en el fluido. Muchos otros fenómenos pueden contribuir también en la manera como un elemento de fluido cambia con el tiempo. El calor puede causar que un gas se expanda, y las reacciones químicas pueden causar que la viscosidad cambie, por ejemplo. Muchos de los procesos tales como los envueltos en la descripción generalizada del movimiento de fluidos se describen por un conjunto de ecuaciones de conservación o de transporte. Este conjunto de ecuaciones, con el tiempo, cambian en el fluido que resulta desde convección, difusión, y aumento o disminución de la cantidad conservada o transportada. Aún más, estas ecuaciones se acoplan, o sea que cambios en una variable (digamos temperatura) pueden producir cambios en otras variables (digamos presión). Las ecuaciones discutidas abajo describen muchos de estos fenómenos de acoplamiento, enfatizando en esos procesos que son típicos en aplicaciones de mezclado.

4.1.1. Continuidad

La ecuación de continuidad es un enunciado de la conservación de la masa. Para entender su origen, consideremos el flujo de un fluido o densidad ρ a través de seis caras de un bloque rectangular, como se muestra en la figura 4.1.

El bloque tiene lados de longitud Δx_1 , Δx_2 y Δx_3 y componentes de velocidad u_1 , u_2 , y u_3 en cada una, de las tres direcciones. Para asegurar la conservación de la masa,

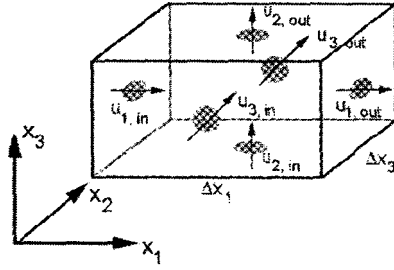


Figura 4.1: Un volumen rectangular con fluido entrante y saliente puede ser usado para ilustrar la ecuación de conservación [11]

la suma, de las masas que fluyen a través de las seis caras debe ser cero.

$$\rho(U_{1,out} - U_{1,in})(\Delta x_2 \Delta x_3) + \rho(U_{2,out} - U_{2,in})(\Delta x_1 \Delta x_3) + \rho(U_{3,out} - U_{3,in})(\Delta x_1 \Delta x_2) = 0 \quad (4.1)$$

Dividiendo la ecuación 4.1 por $(\Delta x_1 \Delta x_2 \Delta x_3)$ la ecuación puede ser escrita como:

$$\rho \frac{\Delta U_1}{\Delta x_1} + \rho \frac{\Delta U_2}{\Delta x_2} + \rho \frac{\Delta U_3}{\Delta x_3} = 0 \quad (4.2)$$

o. en su forma diferencial

$$\rho \frac{\partial U_1}{\partial x_1} + \rho \frac{\partial U_2}{\partial x_2} + \rho \frac{\partial U_3}{\partial x_3} = 0 \quad (4.3)$$

Una manera más compacta de escribir la ecuación (4.3) es usando la notación de Einstein:

$$\rho \frac{\partial U_i}{\partial x_i} = 0 \quad (4.4)$$

Con esta notación, siempre que los índices repetidos ocurren en un término, se asume que hay una suma sobre todos los índices. Aquí, y en cualquier sección, U_i , es el i -ésimo componente de la velocidad del fluido, y las derivadas parciales con respecto a $.x_i$ se asumen como correspondientes a una de las tres direcciones coordenadas. Para los casos más generales, la densidad puede variar en el tiempo y el espacio, y la ecuación de continuidad toma la forma familiar:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho U_i) = 0 \quad (4.5)$$

4.1.2. Cantidad de movimiento

La ecuación de momento es un enunciado de la conservación de la cantidad de movimiento en cada una de las tres componentes direccionales. Las tres ecuaciones de momento son llamadas en conjunto ecuaciones de Navier-Stokes. En adición al transporte de la cantidad de movimiento por convección y difusión, varias fuentes de momento también están involucradas.

$$\frac{\partial(\rho U_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho U_i U_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{2}{3} \frac{\partial U_k}{\partial x_k} \delta_{ij} \right) \right] + \rho g_i + F_i \quad (4.6)$$

En la ecuación 4.6. los términos de convección están a la izquierda. Los términos del lado derecho son gradiente de presión, un término fuente; la divergencia del tensor de esfuerzo, el cual es responsable por la difusión de la cantidad de movimiento; la fuerza gravitacional, otro término fuente; y otras fuerzas generalizadas (términos fuente), respectivamente.

4.1.3. Turbulencia

Un número de parámetros adimensionales han sido desarrollados para el estudio de dinámica de fluidos que son usados para categorizar diferentes regímenes de flujo. Estos parámetros, o números, son usados para clasificar fluidos así como características de fluidos. Uno de los más comunes de estos es el número de Reynolds, definido como el ratio de fuerzas inerciales, o aquéllas que tienden a detener el fluido. En dominios similares geoméricamente, dos fluidos con el mismo número de Reynolds deben comportarse en la misma forma. Para un flujo simple en un tubo, el número de Reynolds se define como

$$Re = \frac{\rho U d}{\mu} \quad (4.7)$$

donde ρ es la densidad del fluido, U es la velocidad axial en el tubo, d es el diámetro del tubo, y μ es la viscosidad molecular o dinámica del fluido. Para tanques mezcladores, una definición modificada que se usa normalmente es:

$$Re = \frac{N D^2 \rho}{\mu} \quad (4.8)$$

donde N es la velocidad del propulsor, en revoluciones/seg, y D es el diámetro del propulsor. Basado en el valor del número de Reynolds, el fluido cae dentro del régimen laminar, con pequeños números de Reynolds, o en el régimen turbulento, con grandes números de Reynolds. La transición entre regímenes laminar y turbulento ocurre a través de un rango de números de Reynolds, antes que en un solo valor. Para el flujo en el tubo, la transición ocurre en la vecindad de $Re = 2000$, mientras que en tanques de mezclado, es usualmente diferente, ocurriendo en alguna parte entre $Re = 50$ y 5000 . En el régimen turbulento, ocurren las fluctuaciones en la velocidad media y otras variables, y sus efectos necesitan ser incorporados dentro del modelo CFD para que el modelo sea consistente y proporcione resultados significativos. Esto se logra usando un modelo de turbulencia.

Varios métodos están disponibles para incluir turbulencia en las ecuaciones de Navier-Stokes. La mayoría de estos incluyen un proceso de promedio en el tiempo de las ecuaciones de conservación. Cuando la turbulencia es incluida, la cantidad de transporte, digamos velocidad, se asume que será la suma de una componente de equilibrio y fluctuación, $U_i + u'_i$. Después del promedio en el tiempo en muchos ciclos de la fluctuación, los términos que contienen los factores de la componente de fluctuación se promedian a cero. El único término que permanece positivo claramente es el que contiene el producto de dos términos de fluctuación. Los términos residuales son idénticos a los de la ecuación 4.6. De esta forma el así llamado Promedio de Reynolds o ecuación RANS (Reynolds-Averaged Navier-Stokes) para cantidad de movimiento es:

$$\frac{\partial(\rho U_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho U_i U_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\mu \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{2}{3} \frac{\partial U_k}{\partial x_k} \delta_{ij} \right) \right] + \frac{\partial}{\partial x_j} (-\rho \overline{u'_i u'_j}) + \rho g_i + F_i \quad (4.9)$$

Los términos nuevos que envuelven $\overline{u'_i u'_j}$ son llamados esfuerzos de Reynolds. La barra superior indica que estos términos representan valores promediados en el tiempo. Los esfuerzos de Reynolds agregan nuevas incógnitas a las ecuaciones RANS, y necesitan ser relacionadas con otras variables. Una variedad de modelos de turbulencia está disponible para este propósito.

4.1.3.1. La hipótesis de Boussinesq

La hipótesis de Boussinesq hace la suposición que el esfuerzo de Reynolds puede ser expresado en términos de los gradientes de la velocidad media. El enunciado de la hipótesis, muestra la introducción de una nueva constante que es equivalente dimensionalmente a la viscosidad:

$$\overline{\rho u'_i u'_j} = \frac{2}{3} \rho k \delta_{ij} + \left(\mu_t \left[\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right] \right) \quad (4.10)$$

La nueva constante, μ_t es la turbulencia, o viscosidad por remolinos. Se puede ver que cuando la ecuación 4.10 es sustituida en la ecuación 4.9, los términos que contienen las derivadas parciales pueden ser combinados y una nueva cantidad, la viscosidad efectiva, puede ser introducida:

$$\mu_{eff} = \mu + \mu_t \quad (4.11)$$

La hipótesis también introduce otro término que envuelve una nueva variable, k , la energía cinética de turbulencia. Esta cantidad es definida en términos de las velocidades de fluctuación u' , v' , y w' en cada una de las tres direcciones coordenadas:

$$k = \frac{1}{2} \left(\overline{u'^2} + \overline{v'^2} + \overline{w'^2} \right) \quad (4.12)$$

El trabajo del modelo de turbulencia es calcular el esfuerzo de Reynolds sustituyendo en la ecuación 4.9. En algunos casos, esto se hace computando los parámetros k y μ_t (o k y μ_{eff}), sustituyendo en la ecuación 4.10 y al final en la ecuación 4.9. Todos los modelos de turbulencia usan algún nivel de aproximación para cumplir esta meta, y es la naturaleza de las condiciones del fluido en cada aplicación específica la que determina cual conjunto de aproximaciones es aceptable para ser usado. Un breve resumen de algunos de los modelos de turbulencia populares usados hoy en día en aplicaciones industriales se da a continuación.

4.1.3.2. El modelo $k - \varepsilon$

El modelo $k - \varepsilon$ pertenece a la familia de modelos de dos ecuaciones, para, los cuales dos ecuaciones de transporte adicionales deben ser resueltas para computar el esfuerzo

de Reynolds. Es un modelo robusto, lo que significa que es estable computacionalmente, aún en presencia de otra, física más compleja. Se aplica a una gran variedad de flujos turbulentos, y ha servido a la comunidad de modelado de fluidos por muchos años. Es un tanto empírico, basado en gran parte en observaciones de los fluidos con número de Reynolds mas grandes. Las dos ecuaciones de transporte que necesitan ser resueltas para este modelo son la energía cinética de turbulencia, k , y la tasa de disipación de turbulencia, ε :

$$\frac{\partial(\rho k)}{\partial t} + \frac{\partial}{\partial x_i}(\rho U_i k) = \frac{\partial}{\partial x_i} \left(\mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_i} + G_k - \rho \varepsilon \quad (4.13)$$

$$\frac{\partial(\rho \varepsilon)}{\partial t} + \frac{\partial}{\partial x_i}(\rho U_i \varepsilon) = \frac{\partial}{\partial x_i} \left(\mu + \frac{\mu_t}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_i} + C_1 \frac{\varepsilon}{k} G_k - C_2 \rho \frac{\varepsilon^2}{k} \quad (4.14)$$

Las cantidades C_1 , C_2 , σ_k y σ_ε son constantes empíricas. La cantidad G_k que aparece en ambas ecuaciones es un término generado para turbulencia. Este contiene productos de gradientes de velocidad, y también depende de la viscosidad de turbulencia:

$$G_k = \mu_t \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \frac{\partial U_j}{\partial x_i} \quad (4.15)$$

Otros términos fuente pueden ser añadido a las ecuaciones 4.13 y 4.14 para incluir otros efectos físicos tales como remolinos, flotabilidad o compresibilidad, por ejemplo. La turbulencia por viscosidad se deriva de ambas k y ε , e involucra una constante tomada de datos experimentales, C_μ , la cual tiene un valor de 0.09:

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \quad (4.16)$$

Para resumir el proceso de solución para el modelo $k-\varepsilon$, las ecuaciones de transporte se resuelven para la energía de turbulencia cinética y la tasa de disipación. Las soluciones para k y ε son usadas para computar la viscosidad de turbulencia, μ_t . Usando los resultados para μ_t y k , el esfuerzo de Reynolds puede ser computado por substitución en las ecuaciones de cantidad de movimiento. Una vez que las ecuaciones de cantidad de movimiento han sido resueltas, los nuevos componentes de velocidad se usan para actualizar el término de generación de turbulencia. G_k , y el proceso se repite.

4.1.3.3. El modelo RNG $k - \varepsilon$

EL modelo RNG (Yakhot, 1986) fue desarrollado en respuesta a la naturaleza empírica, del modelo $k - \varepsilon$ estándar. Antes que basado en comportamiento de fluido observado, se deriva del uso de métodos estadísticos usados en el campo de la teoría, de grupos de renormalización (RNG). Es similar en su forma al modelo $k - \varepsilon$ estándar, pero contiene modificaciones en las ecuaciones de disipación para describir mejor los flujos con regiones de alto esfuerzo, tales como fluidos en un codo o recirculados siguiendo una zona de recirculación. En adición, una ecuación diferencial es resuelta para la turbulencia por viscosidad. Cuando la solución de esta ecuación diferencial es evaluada en el límite de un número de Reynolds grande, la ecuación 4.16 es regresada con un coeficiente, C_μ , de 0.0845, dentro del 7% del valor empírico 0.09. Mientras el modelo RNG trabaja bien para, flujos con número de Reynolds grande, también trabaja bien para flujos de transición, donde el número de Reynolds está en el rango de baja turbulencia.

4.1.3.4. El modelo $k - \varepsilon$ factible

El modelo $k - \varepsilon$ factible (Shih, 1995) es una adición a la familia de modelos de dos ecuaciones considerablemente reciente. Difiere del modelo $k - \varepsilon$ estándar en dos formas. Primero, la viscosidad por turbulencia es computada de diferente manera, haciendo uso de la ecuación 4.16, pero usando una variable para la cantidad C_μ . Esto es motivado por el hecho que en el límite de flujos con alto esfuerzo, algunos de los esfuerzos normales de Reynolds $\overline{u_i^2}$, pueden llegar a ser negativos en la formulación $k - \varepsilon$. lo cual es antifísico, o irrealizable. La forma variable de la constante C_μ es una función de la tasa de esfuerzo local y la rotación del fluido, y está diseñado para prevenir valores antifísicos de los esfuerzos normales desarrollados.

La segunda diferencia es que el modelo $k - \varepsilon$ factible usa diferentes términos de aumento o disminución en la ecuación de transporte para disipación de remolineo. La ecuación resultante es considerablemente diferente de la usada para ambos los modelos $k - \varepsilon$ estándar y RNG A: $k - \varepsilon$. La predicción modificada de ε , junto con el cálculo

modificado para μ_t , hace que este modelo de turbulencia sea superior que otros modelos $k-\varepsilon$ para un buen número de aplicaciones. En particular el modelo es mejor para predecir la tasa de esparcimiento de turbinas rotatorias, tales como la emitida por un aspa propulsora, rotatoria.

4.1.3.5. El modelo RSM

El modelo de esfuerzo de Reynolds (RSM) no usa la hipótesis de Boussinesq. Antes que asumir que la viscosidad por turbulencia es isotrópica, teniendo un valor, como se hace en el modelo $k-\varepsilon$, el modelo de esfuerzo de Reynolds computa los esfuerzos, $\overline{u'_i u'_j}$ individualmente. Para modelos 2D, esto significa cuatro ecuaciones de transporte adicionales. Para modelos 3D, seis ecuaciones de transporte adicionales son requeridas. Junto con la ecuación de transporte para ε , que también debe ser resuelta en el modelo RSM, el efecto completo de la turbulencia puede ser representado en las ecuaciones de la cantidad de movimiento con mayor precisión que la que puede ser obtenida de modelos $k-\varepsilon$. Los flujos para los cuales la suposición de viscosidad de turbulencia isotrópica falla incluye aquellos con remolineo elevado, cambios rápidos en tasa de esfuerzo, o curvatura con líneas aerodinámicas significativas. Debido a que el poder computacional y la velocidad se han incrementado con el paso de varios años, el uso del modelo de turbulencia de esfuerzo de Reynolds ha llegado a ser más difundido, dando un incremento en la precisión por encima de otros modelos de turbulencia basados en RANS cuando los resultados experimentales son comparados para muchas aplicaciones, tales como el flujo en recipientes de batido no regulares.

4.2. El modelo de corte rotativo

El modelo de corte rotativo resuelve las ecuaciones de momento para el dominio completo del corte rotativo. La fuerza de coriolis es incluida en el proceso. Los problemas resueltos en un corte rotativo típicamente usan la velocidad angular de la componente rotativa primaria, Ω , como la velocidad angular del corte. En tanques de

mezclado, los propulsores sirven a este propósito, así se asume que el corte gira con el propulsor. Así el propulsor esta en reposo en el corte rotativo. El tanque, sin embargo, gira en la dirección opuesta, así debe tener una condición de frontera de $-\Omega$. Si existen deflectores, ellos deben rotar dentro del fluido con la misma velocidad angular, $-\Omega$. Desafortunadamente, este simple modelo de estado estable no esta equipado para tratar el movimiento de elementos tales como deflectores dentro o a través del fluido. La aproximación es por tanto útil solamente para tanques no deflectores con paredes alisadas que son equivalentes geoméricamente a una perfecta superficie de revolución. Así un tanque cilindrico no deflector con fondo con eje de simetría y sin dependencia angular interna puede ser simulado de esta manera. Recipientes con deflectores, tubos inmersos, o puertos de entrada/salida de flujo no pueden ser simulados.

4.3. El algoritmo SIMPLE

Para simulaciones 3D, las tres ecuaciones de movimiento 4.6 y la ecuación de continuidad 4.5 se combinan para formar cuatro ecuaciones para las cuatro incógnitas que son la presión y la tres componentes de velocidad. A causa de que no hay ecuación explícita para, la presión, técnicas especiales han sido desarrolladas para extraerla en una forma alternativa. La mejor conocida de estas técnicas es el algoritmo SIMPLE, o Método Semi-Implícito para Ecuaciones Ligadas a Presión (Patankar, 1980 [13, p. 126-130]). En verdad, una familia de algoritmos ha sido derivada de este básico, cada uno de estos tiene una pequeña modificación que lo hace apto para una aplicación u otra.

La esencia del algoritmo es como sigue. Un supuesto campo de presión es usado en la solución de las ecuaciones de momento. (Para todas excepto la primera iteración, el supuesto campo de presión es simplemente la última actualización). Las velocidades nuevas son computadas, pero estas en general, no podrán satisfacer la ecuación de continuidad, así las correcciones a las velocidades están determinadas. Basados en las correcciones de velocidad, una corrección en la presión es computada con lo cual, cuando

es añadida a la presión supuesta originalmente, resulta en una presión actualizada. Siguiendo la solución de las variables restantes del problema, la iteración es completada y el proceso entero repetido.

4.4. Resumen del capítulo

En este capítulo hemos hablado acerca de las ecuaciones conservación que incluyen ecuaciones de continuidad y ecuaciones de cantidad de movimiento, además se describió los principales modelos de turbulencia que utilizan la hipótesis de Boussinesq y el modelo RSM que no la utiliza. Se finalizó con la descripción del algoritmo de corte rotativo y el algoritmo SIMPLE. Esta sección cobra fuerza al aplicarse los modelos aquí descritos dentro de *OpenFOAM*, el cuál será descrito en el siguiente capítulo.

Capítulo 5

Modelado de un ventilador de flujo tubo-axial en *OpenFOAM*

En este capítulo se describe el simulador de fluidos *OpenFOAM* y la forma como fue empleado para modelar un aspa de ventilador de flujo tubo-axial. Se describirá el tipo de geometría empleada y la composición de celdas participantes en la construcción de la forma del aspa. Se hablará de la codificación del aspa a partir de los puntos de control. Se dará una representación matemática entre los puntos del aspa y la figura tridimensional resultante. Se finalizará el capítulo con la forma de visualizar resultados obtenidos en *OpenFOAM*. Puede decirse que la mitad del trabajo recae en esta sección, ya que el mallado y representación geométrica constituyen por si mismos un problema a resolver por separado.

5.1. Descripción de *OpenFOAM*

OpenFOAM es un software capaz de simular casi cualquier cosa (que pueda ser representada como ecuación diferencial) desde fluidos complejos que envuelvan reacciones químicas, turbulencia y transferencia de calor, hasta la dinámica de sólidos, electromagnetismo y la fijación de precios de opciones financieras. *OpenFOAM* es producido por *OpenCFD Ltd*, se encuentra disponible libremente y es *open source* con licencia bajo *GNU General Public Licence*.

La estructura general de *OpenFOAM* es un conjunto de módulos C++ flexibles. Éstos se utilizan para poder simular problemas específicos de la ingeniería mecánica y

para llevar a cabo pre y post-procesamiento de tareas que van desde la simple manipulación de datos hasta la visualización y procesamiento de malla. También es posible crear bibliotecas de modelos físicos, por ejemplo. *OpenFOAM* viene provisto con numerosos métodos numéricos específicos o *solvers* preconfigurados, utilerías y librerías y puede ser usado como cualquier paquete de simulación. Sin embargo, es libre, no sólo en términos del código fuente, sino también en su estructura y diseño, así que sus *solvers*, utilerías y librerías son completamente extensibles [2].

OpenFOAM trabaja sobre la plataforma *linux*. En nuestro caso hemos utilizado *debian lenny*, ya que es el sistema operativo con el que estamos más familiarizados. Todo lo que en adelante se realice puede ser reproducido, ya que la adquisición tanto de *OpenFOAM* como de *debian lenny* solo se limita a tener una buena conexión a internet.

5.2. Justificación del uso de *OpenFOAM*

Existen diversos paquetes que pueden ser empleados para lograr la optimización de diseños, nos centraremos en particular en dos de ellos anteriormente empleados por Waldo Arcos en su tesis de maestría, estos paquetes son: *CFdesign* y *Fluent*. Los siguientes párrafos describen la perspectiva de Waldo respecto al uso de estos paquetes computacionales:

“*CFdesign* es un programa integrado en un paquete *CAD*, lo cual evita problemas de importación de geometría en el analizador *CFdesign* ya que tiene la ventaja de eliminar el uso de un paquete intermedio de mallado entre estos 2. En *Fluent* la desventaja que existe es utilizar un paquete intermedio de mallado, que en este caso fue *Gambit*, lo cual ocasiona problemas al momento de importar la geometría” [3, p. 58].

“Una vez que se logró importar la geometría dentro de *Gambit* y se revisó que esta no tiene discontinuidades, el lograr mallar la geometría resulta en otra complicación adicional. Se recomienda utilizar un valor de tamaño de malla pequeño si la geometría cuenta con muchos redondeos o perfiles esquinados. En *CFdesign* el mallado lo hace

Pro/ENGINEER, sólo se selecciona el tamaño de malla. Una desventaja que se presenta en *CFdesign* es el no permitir seleccionar el tipo de malla (elementos tipo triangular, cuadriláteros, hexaedros, etc.) que se desea para el análisis. Lo cual resulta importante para obtener resultados exactos sin gasto de tiempo de procesamiento e interpretación de datos” [3, p. 58].

De los párrafos anteriores nos damos cuenta que existe el problema de no tener un control de la malla generada. Por lo que en un primer intento de lograr un mejor control del mallado recurriremos al paquete *CFD* conocido como *OpenFoam*, el cual se ha descrito con anterioridad.

5.3. La geometría básica de OpenFoam

OpenFOAM está diseñado como código para espacio tridimensional y define todas sus mallas en esta forma. Sin embargo, problemas de una y dos dimensiones y eje simétrico pueden ser simulados en *OpenFOAM* generando una malla en tres direcciones y aplicando condiciones especiales de frontera en cada parche en el plano normal a la o las direcciones de interés [1]. La geometría básica de *OpenFOAM* consta de puntos (*points*), aristas(*edges*) y caras(*faces*). Hay diversas configuraciones que pueden ser consultadas en el manual de usuario, pero solo necesitaremos la de tipo *hex* que se encuentra definida en la figura 5.1.

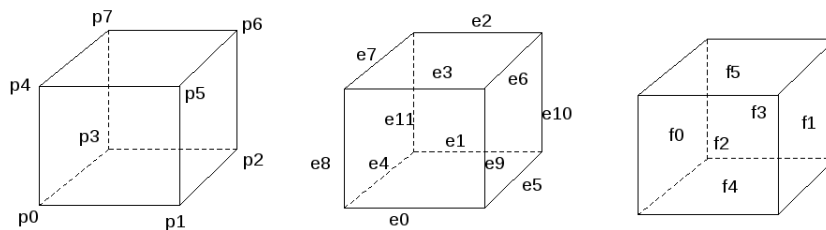


Figura 5.1: Celdas tipo *hex*: Puntos, aristas y caras definidas en *OpenFOAM*.

La orientación definida por *OpenFOAM* debe seguirse a fin de que el proceso de mallado automático sea realizado de manera rápida y eficiente. En nuestro caso se ha logrado que *OpenFOAM* haga las mallas con celdas de tipo *hex* gracias a la forma

de definir las celdas que componen el modelo. Son cuatro las celdas básicas que dan forma al aspa del ventilador. Estas cuatro celdas forman un corte de un octavo del cilindro, es decir, están simulando un ventilador de ocho aspas. El número de divisiones de las celdas se debe escoger de tal forma que sea lo suficientemente pequeño para que se obtenga una buena aproximación y a la vez el tiempo de procesado sea corto. A continuación se describen las cuatro celdas tal como aparecen en la figura 5.2:

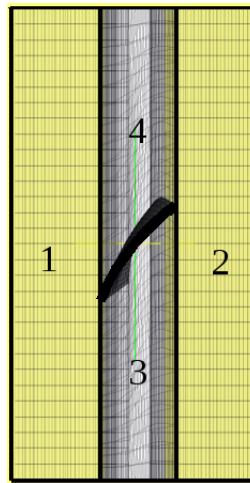


Figura 5.2: Cuatro celdas tipo *hex* definiendo el aspa del ventilador.

Celda 1: Esta celda posee una pared cíclica que simula estar en contacto con la pared cíclica de la celda 2. El número de divisiones a lo alto es de 30, el número de divisiones a lo ancho es de 10 y el número de divisiones de profundidad (hacia dentro de la página) es de 24.

Celda 2: Esta celda posee una pared cíclica que simula estar en contacto con la pared cíclica de la celda 1. El número de divisiones a lo alto es de 30, el número de divisiones a lo ancho es de 10 y el número de divisiones de profundidad es de 24.

Celda 3: Esta celda posee una cara superior que simula una pared del aspa. El número de divisiones a lo ancho es de 20, el número de divisiones a lo largo es de 15 y el número de divisiones de profundidad es de 24.

Celda 4: Esta celda posee una cara inferior que simula una pared del aspa. El número de divisiones a lo ancho es de 20, el número de divisiones a lo largo es de 15 y el número de divisiones de profundidad es de 24.

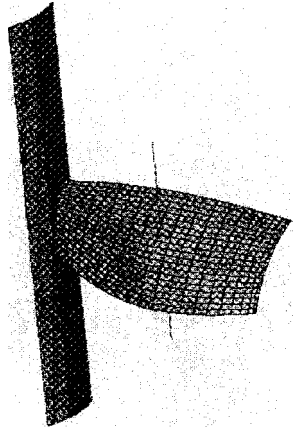


Figura 5.3: Mallado proyectado sobre un aspa del ventilador.

La figura 5.3 puede darnos una mejor idea de como se forma el aspa del ventilador de flujo axial. La vista tridimensional muestra cómo se proyectan las divisiones de las cuatro celdas básicas sobre el aspa del ventilador y sobre la pared interna del cilindro.

5.4. Parámetros del aspa

En esta sección se definirá la relación que existe entre el genotipo de un cromosoma y un aspa de ventilador de flujo axial. El genotipo se encuentra representado por 12 parámetros ordenados como se muestra a continuación.

$$[a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2, a_3, b_3, c_3, d_3]$$

Los valores que estos parámetros pueden tomar son números reales entre 0 y 1, esto es para $1 \leq i \leq 3$:

$$0 \leq a_i \leq 1; 0 \leq b_i \leq 1; 0 \leq c_i \leq 1; 0 \leq d_i \leq 1$$

Dado que se trata de una geometría con simetría cilíndrica, podemos usar las transformaciones de coordenadas cilíndricas a cartesianas y obtener las siguientes relaciones

para $1 \leq i \leq 3$:

$$\begin{aligned}X_i^{ab} &= r_i \cos\left(\frac{\pi}{8}a_i\right) \\Y_i^{ab} &= r_i \operatorname{sen}\left(\frac{\pi}{8}a_i\right) \\Z_i^{ab} &= Hb_i \\X_i^{cd} &= r_i \cos\left(-\frac{\pi}{8}c_i\right) \\Y_i^{cd} &= r_i \operatorname{sen}\left(-\frac{\pi}{8}c_i\right) \\Z_i^{cd} &= -Hd_i\end{aligned}$$

donde

a — y positivo

b = z positivo

c = y negativo

d — z negativo

H = altura media de crecimiento del aspa

Ahora nuestra, aspa quedará determinada por los nueve puntos siguientes (ver figuras 5.4, 5.5 y 5.6):

1. $(X_1^{ab}, Y_1^{ab}, Z_1^{ab})$
2. $(r_1, 0, 0)$
3. $(X_1^{cd}, Y_1^{cd}, Z_1^{cd})$
4. $(X_2^{ab}, Y_2^{ab}, Z_2^{ab})$
5. $(r_2, 0, 0)$
6. $(X_2^{cd}, Y_2^{cd}, Z_2^{cd})$

7. $(X_3^{ab}, Y_3^{ab}, Z_3^{ab})$

8. $(r_3, 0, 0)$

9. $(X_3^{cd}, Y_3^{cd}, Z_3^{cd})$

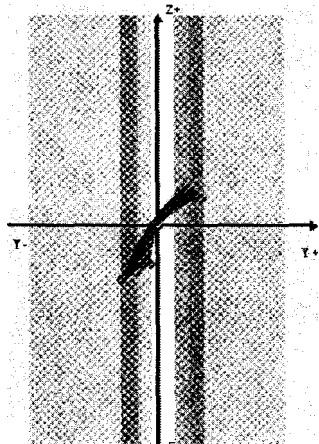


Figura 5.4: Vista del aspa hacia el sentido negativo del eje X

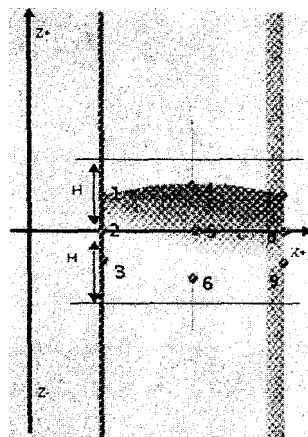


Figura 5.5: Vista del aspa hacia el sentido positivo del eje Y

Para unir estos puntos y suavizar el aspa usaremos las siguientes relaciones:

arcos: Son tres los arcos usados, el primero une los puntos 1, 2 y 3; el segundo une los puntos 4, 5 y 6; el tercero une los puntos 7, 8 y 9.

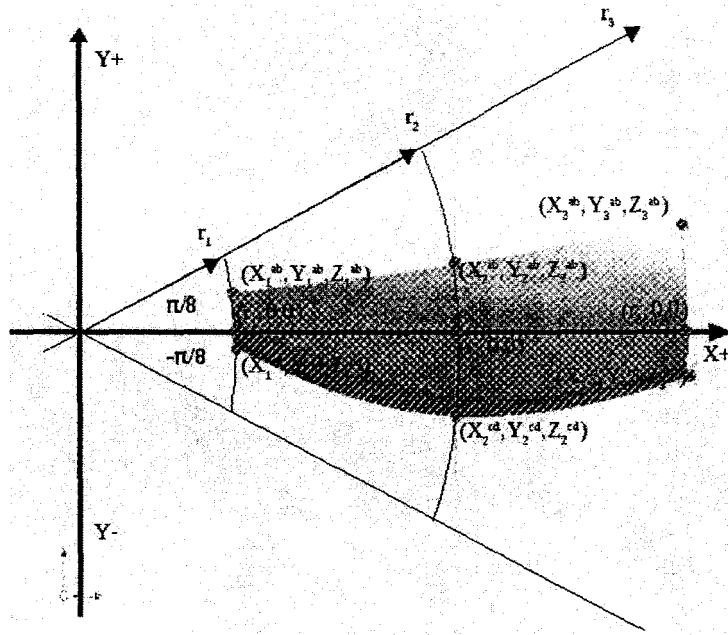


Figura 5.6: Vista del aspa hacia el sentido negativo del eje Z

splines: Son dos los splines usados, el primero une los puntos 1, 4 y 7; el segundo une los puntos 3, 6 y 9.

El toque final para la forma del aspa es darle un grosor a la hoja aquí resultante. En nuestro caso el grosor de nuestra aspa es de 2 milímetros.

Para mayor claridad, mostramos el aspa resultante en las figuras 5.4, 5.5 y 5.6. El aspa corresponde al cromosoma con los valores [0.576, 0.536, 0.229, 0.395, 0.394, 0.693, 0.503, 0.646, 0.402, 0.543, 0.216, 0.439]. De acuerdo a nuestro modelo tenemos que: $a_1 = 0.576$, $b_1 = 0.536$, $c_1 = 0.229$, $d_1 = 0.395$, $a_2 = 0.394$, $b_2 = 0.693$, $c_2 = 0.503$, $d_2 = 0.646$, $a_3 = 0.402$, $b_3 = 0.543$, $c_3 = 0.216$ y $d_3 = 0.439$. Las medidas fijas que hemos usado son las siguientes: $H = 0.02$ m, $r_1 = 0.025$ m, $r_2 = 0.0625$ m y $r_3 = 0.1$ m. Para los valores anteriores se obtienen los siguientes nueve puntos resultantes en coordenadas cartesianas:

1. (0.0243, 0.0056, 0.01072)

2. (0.025, 0, 0)

3. (0.02489, -0.002245, -0.0079)
4. (0.06175, 0.00963, 0.01386)
5. (0.0625, 0, 0)
6. (0.061284, -0.012265, -0.01292)
7. (0.09875, 0.0157, 0.01086)
8. (0.1, 0, 0)
9. (0.09964, -0.00847, -0.00878)

Con estos puntos ya podemos tener una idea de como es el aspa o bien hacer la plantilla o modelo para pruebas de laboratorio.

5.5. Visualización de resultados



Figura 5.7: Flujo de aire con velocidad absoluta

Antes de poner en marcha nuestros modelos, es una buena idea ver la malla para ver cualquier error. La malla puede ser vista en *paraView*, la herramienta de post-procesamiento suplida con *OpenFoam*. En la práctica fue muy útil el convertir al formato *vtk*, lo cual se logra con la instrucción *foamToVTK*. La ventaja del formato *vtk* es que

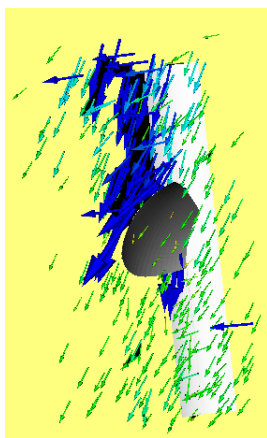


Figura 5.8: Flujo de aire con velocidad relativa

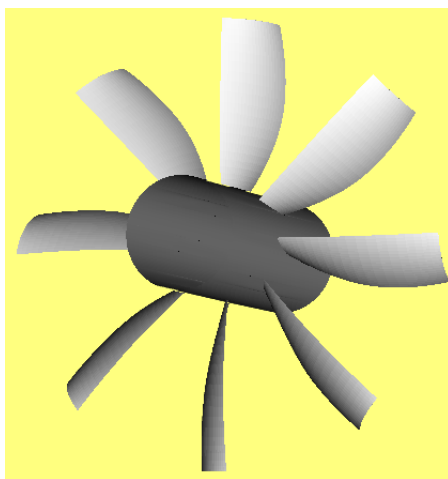


Figura 5.9: Ventilador de flujo axial con ocho aspas

no sólo se puede ver los resultado con la interfaz gráfica de *paraView*, sino que también es posible utilizar instrucciones propias de *vtk* en *scripts* de Python o C++ y grabar los resultados en imágenes de extensión *png*. Con las imágenes de extensión *png* es posible hacer animaciones de formato *avi*. Para las animaciones en particular, se usó el programa *avidemux* ya que permite hacer las animaciones de manera fácil y rápida.

La figura 5.7 fue creada mediante un *script* de Python, en ella se muestra un aspa de ventilador con las flechas apuntando en la dirección del flujo de aire. Este flujo representa la velocidad absoluta generada por el aspa. Para visualizar lo que sucede, imaginemos que vemos pasar el aspa a toda velocidad de izquierda a derecha, mientras

permanecemos inmóviles.

La figura 5.8 también fue creada mediante un *script* de Python, en ella se muestra la velocidad relativa generada por el aspa. Para visualizar lo que sucede, imaginemos que nos movemos junto con el aspa de izquierda a derecha, así veremos el aspa quieta mientras el aire queda atrás. Finalmente el ventilador modelado en nuestro trabajo de investigación se muestra completo en la figura 5.9. El sentido de giro de este ventilador es contrario al de las manecillas del reloj.

5.6. Resumen del capítulo

En este capítulo se ha descrito *OpenFOAM* y la forma como fue empleado para modelar un aspa de ventilador de flujo tubo-axial. Se describió el tipo de geometría empleada y la composición de celdas participantes en la formación de la forma del aspa. Se dio una representación matemática entre los puntos del aspa y la figura tridimensional resultante y se finalizó el capítulo con la forma de visualizar resultados obtenidos en *OpenFOAM*.

Capítulo 6

Algoritmos empleados en proceso de optimización

En este capítulo se describirá la variación realizada al algoritmo *Genitor*. Esta variación consiste en utilizar ranqueo cuadrático en lugar de ranqueo lineal. Se dará un algoritmo complementario que es utilizado en dos pasos del algoritmo genitor, este algoritmo hace la evaluación y envía un correo para que vía internet se puedan ver las evaluaciones obtenidas. Se finaliza el capítulo con el algoritmo que hace la evaluación del cromosoma del aspa descrita en el capítulo anterior. Esta separación de algoritmos fue necesaria para que funcionaran con independencia, es decir, en un futuro con pequeñas variaciones se pueden usar por separado las partes del sistema de optimización que ahora se compone de los tres algoritmos.

6.1. Algoritmo de optimización

El siguiente pseudocódigo muestra los pasos que hemos seguido para optimizar mediante la variación al algoritmo genitor:

ALGORITMO DE OPTIMIZACIÓN

1. Genera población inicial
2. Evalúa población
3. Repite desde 1 hasta número de ciclos
 - a) Selecciona 2 individuos de la población

- b) Cruza los dos padres seleccionados y obtén un hijo único
- c) Muta el hijo resultante
- d) Adhiere el hijo a la población
- e) Evalúa población
- f) Elimina el peor individuo de la población
- g) Salva el mejor individuo del ciclo

La figura 6.1 muestra un diagrama con los pasos de la variación al algoritmo Genitor.

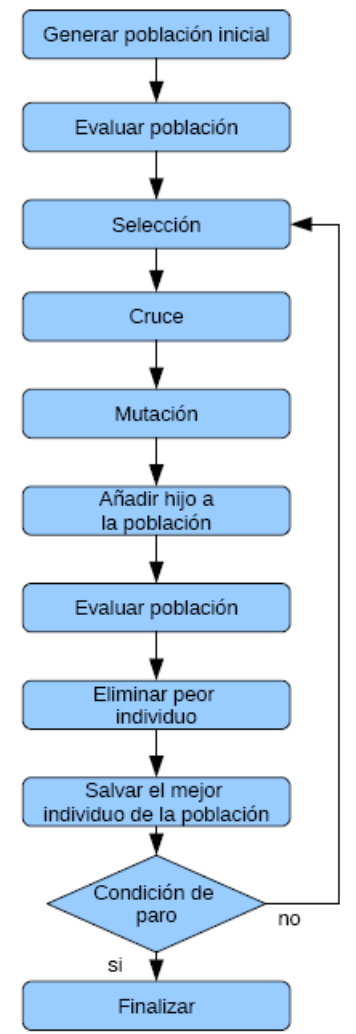


Figura 6.1: Algoritmo de optimización

Los pasos dados en la variación del algoritmo genitor serán descritos aquí para una mejor comprensión:

Genera población inicial: Este paso consiste en generar un archivo con una población inicial. En nuestro caso fue de 30 individuos con un genotipo formado 12 parámetros aleatorios con valores reales entre 0.3 y 0.7, un fenotipo inicial de 100.0 m/s y un parámetro de evaluación igual a 0. Este parámetro cambia a 1 cuando el individuo en cuestión recibe una evaluación en *OpenFOAM*.

Evalúa población: Por claridad se dedicará la siguiente sección a la descripción de este paso, ya que también es un algoritmo y debe dársele un trato especial.

Repite desde 1 hasta número de ciclos: Indica los ciclos que ocurren después de que se generó la población inicial o bien la población resultante a lo largo de la evolución del algoritmo.

Selecciona 2 individuos de la población: En este paso se toman dos individuos de la población, los cuales son seleccionados por ranqueo. Tradicionalmente el ranqueo utilizado es de tipo lineal [14, p. 82], pero decidimos utilizar ranqueo cuadrático para acelerar la convergencia ya que el tiempo de que disponíamos era muy limitado. En nuestro caso el ranqueo se forma utilizando una serie de cuadrados. La suma total de cuadrados de m naturales es obtenida con la fórmula:

$$1^2 + 2^2 + \dots + m^2 = \frac{m(m+1)(2m+1)}{6}$$

Si la población está formado por n individuos ordenados del peor al mejor, se tiene que la siguiente fórmula indica la probabilidad de que el k -ésimo individuo sea seleccionado:

$$P(k) = \frac{6 k^2}{n(n+1)(2n+1)}$$

donde $1 \leq k \leq n$. Para una población de 30 individuos se tiene que, al peor individuo se le da 1^2 partes de 9455. al segundo 2^2 partes de 9455, al tercero 3^2 partes de 9455, ..., al mejor individuo se le dan 30^2 partes de 9455.

Cruza los dos padres seleccionados y obtén un hijo único: En este paso de manera aleatoria se obtiene a que cumpla con $-0.1 \leq a \leq 1.1$ y se obtiene b de la siguiente forma $b = 1 - a$. Llamemos a al cromosoma padre y β al cromosoma madre, llamemos también γ al cromosoma hijo resultante. Ahora para cada uno de los 12 parámetros del cromosoma hijo resultante tenemos:

$$\gamma_i = a\alpha_i + b\beta_i$$

donde $1 \leq i \leq 12$

Muta el hijo resultante: Aquí la mutación se refiere al acto de agregar al hijo resultante un vector aleatorio de valores entre -0.1 y 0.1 de tal forma que el individuo resultante sea muy parecido al original, con ligeras perturbaciones. La probabilidad de mutación utilizada fue de 0.1 . Como puede observarse, siempre que se tenga que realizar una mutación, esta, se realizará en los 12 parámetros del cromosoma resultante. Llamemos δ al cromosoma resultante de una mutación de γ , llamemos también p al vector de perturbación o mutación. Para cada uno de los 12 parámetros del cromosoma mutado tenemos:

$$\delta_i = \gamma_i + p_i$$

donde $-0.1 \leq p_i \leq 0.1$

Adhiere el hijo a la población: Una vez que tenemos el nuevo individuo, este se añade a la población con una evaluación de 100.0 m/s y una marca de no evaluado (eval 0). para que sea reconocido como candidato a, evaluar por la función de evaluación. La forma de añadirse es directamente al final del archivo de población.

Elimina el peor individuo de la población: Una vez que el último individuo ha sido evaluado, se ordenan todos los individuos de la población del mejor al peor de acuerdo a su fenotipo. Ahora se elimina el último renglón del archivo de población por ser el peor y de esa forma el número de individuos en la población permanece constante. En nuestro caso son 30 individuos.

Salva el mejor individuo del ciclo: Ahora que ya tenemos todos los individuos evaluados podemos salvar el mejor del ciclo que ha quedado en el primer renglón del archivo de población. El lugar donde se guarda es un archivo con los mejores individuos de cada ciclo. Esta información puede servir para graficar el comportamiento de nuestro algoritmo.

Como puede verse en esta sección se tienen archivos donde se va guardando y tomando datos, de tal forma que en caso de una falla usaremos las copias para continuar el algoritmo desde donde se había quedado. Esto nos permite recuperar tiempo de evaluación que pudiera perderse a causa digamos de un apagón. Otro beneficio es que nos permite averiguar cómo se ha desempeñado el algoritmo y decidir si vale la pena hacer más evaluaciones de las contempladas o parar de evaluar.

Como se indicó en los pasos (2) y (e) queda pendiente el algoritmo que evalúa la población y que será explicado en la siguiente sección.

6.2. Algoritmo de evaluación

En esta sección se describirá el algoritmo de evaluación correspondiente a los pasos (2) y (e) del algoritmo de optimización, el código es el mismo para ambos pasos, así que basta con hacer su descripción una sola vez:

ALGORITMO DE EVALUACIÓN DE LA POBLACIÓN

1. Repite mientras exista un cromosoma sin evaluar
 - a) Procesa cromosoma
 - b) Retorna fenotipo
 - c) Envía un mensaje

La figura 6.2 muestra los pasos (2) y (e) del algoritmo de optimización empleado por nosotros.

La descripción de los pasos aquí seguidos es como se expresa a continuación:

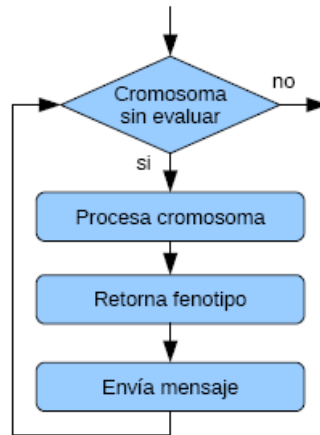


Figura 6.2: Algoritmo de evaluación de la población

Repite mientras exista un cromosoma sin evaluar: Este paso consiste en un ciclo que se repite mientras se encuentren individuos que no hayan tenido aún la evaluación de su genotipo o bien que tengan la marca eval igual a 0. En este paso también se contempla el enviar el individuo a un archivo especializado para su posterior evaluación. Los pasos contemplados en este ciclo se dan a continuación.

Procesa cromosoma: El procesado del genotipo de un individuo también requiere de un trato especial por lo que la siguiente sección está dedicada a este procesado para una mayor claridad.

Retorna fenotipo: Una vez que se ha logrado evaluar al individuo, este se coloca en un archivo especial para que su fenotipo sea escrito posteriormente en la población.

Envía un mensaje: En este paso se envía un archivo a una cuenta de correo electrónico, este archivo tiene la información de los individuos evaluados en el paso (a). El correo enviado en este paso tiene la finalidad de llevar un conteo del número de individuos evaluados y los fenotipos obtenidos. Esto puede servir para darse una idea de la tendencia del algoritmo y además para detectar posibles errores o fallos. Simplemente se abre el correo y dependiendo del último correo enviado se deduce el posible problema. Esto nos libera un poco de tener que estar vigilando

constantemente el desarrollo del proceso de optimización y nos da una seguridad de que las cosas marchan bien al no detectar errores.

La idea de mostrar el procesado del cromosoma en otra sección es porque no se trata simplemente de una sola evaluación del genotipo sino de 7 evaluaciones y lo que se reporta es el promedio, además de otros detalles que necesitan ser explicados por separado para evitar confusiones.

6.3. Algoritmo de procesado de un cromosoma

El siguiente algoritmo representa el proceso seguido para lograr una evaluación del cromosoma que representa al aspa, en el se tratan detalles más específicos que tienen que ver con *OpenFOAM*.

ALGORITMO DEL PROCESO DE EVALUACIÓN DEL CROMOSOMA

1. Crea el diccionario *blockMeshDict*
2. Crea la malla
3. Repite desde 1 hasta N
 - a) Cambia rpm
 - b) Cambia omega
 - c) Ejecuta *simpleSRFFoam*
 - d) Obtén la velocidad promedio
4. Obtén promedio de velocidades
5. Escribe archivo de resultados
6. Escribe archivo de fenotipo

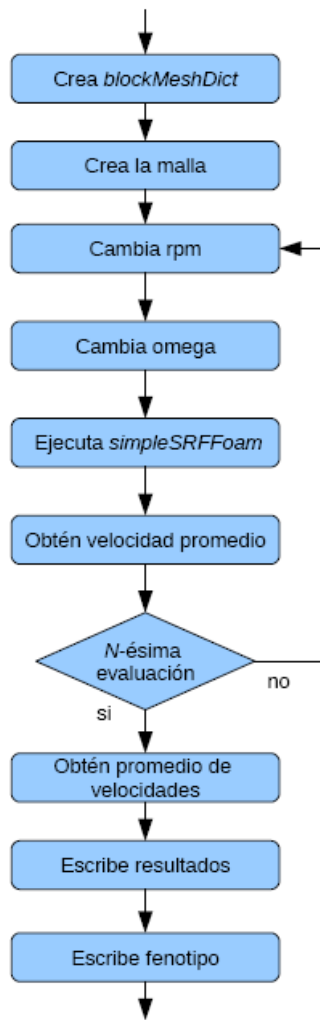


Figura 6.3: Algoritmo de procesado de un cromosoma en *OpenFOAM*

La figura 6.3 muestra los pasos del proceso de evaluación en *OpenFOAM*. Esta evaluación de la velocidad media es reportada al algoritmo de optimización como función objetivo del proceso de optimización.

Aquí se describen los pasos anteriores para una mejor comprensión:

Crea el diccionario *blockMeshDict*: Este paso se refiere a la creación de un archivo llamado *blockMeshDict* utilizado por *OpenFOAM* para un mallado automático. Los doce parámetros del genotipo del cromosoma escogido se usan para definir la forma del aspa. De acuerdo con las reglas de *OpenFOAM* se crean los bloques y se define el número de divisiones que debe tener cada bloque.

Crea la malla: En este paso una vez que se tiene el diccionario *blockMeshDict*, se usa la orden *blockMesh* que de manera automática crea la malla del aspa de ventilador dentro del corte de cilindro, de tal forma que el *solver simpleSRFFoam* pueda simular un fluido a través del sistema.

Repite desde 1 hasta N : Este paso representa un ciclo de N evaluaciones de aspa. En nuestro caso se usó $N = 7$, lo que representa 7 diferentes velocidades. Se empieza por la velocidad de giro mayor y se finaliza con la velocidad de giro menor de tal forma que, si ocurriera un error de mallado, la velocidad de giro menor será tomada y el incidente no acarreará errores al desempeño del algoritmo de optimización.

Cambia rpm: Este paso se refiere al cambio de las revoluciones por minuto. En nuestro caso fue cambiar de 1400 a 2000 para empezar a evaluar, posteriormente de 2000 a 1900, de 1900 a 1800, ..., de 1500 a 1400. Solo uno de estos cambios es realizado en cada paso hasta completar un ciclo. El cambio es realizado a un archivo llamado *SRFProperties* mediante un *script*.

Cambia omega: Este paso se refiere al cambio de la velocidad angular para consistencia con las revoluciones por minuto del paso anterior. El archivo a modificar se llama *omega*.

Ejecuta *simpleSRFFoam*: Este paso se refiere propiamente a la simulación mediante el *solver simpleSRFFoam*. Este *solver* utiliza el algoritmo SIMPLE y una sola referencia en un corte de la sección del cilindro. Como se trata de un solo elemento giratorio, no es necesario utilizar referencias múltiples. El aspa de abanico representa al rotor y la pared del cilindro al estator. El número de iteraciones usadas es de 1000 para una mayor seguridad de convergencia.

Obtén la velocidad promedio: Una vez que el *solver simpleSRFFoam* ha terminado las 1000 iteraciones, usamos un *script* para leer la velocidad media encontrada en la pared de la salida del cilindro.

Obtén promedio de velocidades: Este paso se refiere a la obtención de la velocidad promedio del total de las N velocidades medias encontradas en el ciclo anterior.

Escribe archivo de resultados: Una vez que se han realizado las N evaluaciones de velocidad media y se ha obtenido el promedio total de estas evaluaciones, se procede al reporte de estos resultados en un archivo especial. Este archivo es el que se envía por correo a la persona encargada del proceso de optimización. Este archivo proporciona información inclusive si ocurre un error, pues reporta la evaluación tal y como *simpleSRFFoam* la reporta. Un caso de error claramente detectable es cuando las N velocidades correspondientes a un genotipo resultan idénticas a la velocidad menor del genotipo anterior.

Escribe archivo de fenotipo: Este paso se refiere a la escritura de un archivo con el fenotipo encontrado. Si la velocidad es demasiado grande para representar una velocidad real, se le hace un cambio para que no afecte al desempeño del algoritmo de optimización.

La evaluación del cromosoma parece compleja por tomar N número de velocidades, pero permite dar un rango más amplio de funcionamiento del ventilador y no solo un máximo en una sola velocidad. Esto obedece a la idea de manejar una velocidad más apropiada en cada situación.

6.4. Resumen del capítulo

En este capítulo se vio la variación realizada al algoritmo genitor y los algoritmos complementarios para la realización del proceso de optimización. Se observó que se puede utilizar también el correo electrónico para envío de las evaluaciones resultantes, lo que resulta muy útil a la hora de detectar errores por falla eléctrica principalmente.

Capítulo 7

Resultados

En esta sección hablaremos acerca de los principales resultados obtenidos, los principales puntos que se discuten son: el haber logrado la automatización del proceso de optimización mediante software libre, el haber encontrado la forma de codificar el aspa, el haber encontrado una forma de detectar errores y evitar pérdida de tiempo al monitorear el proceso a distancia, el haber encontrado una aparente relación entre la convergencia del algoritmo genitor y el ranqueo para nuestro modelo, finalmente haber encontrado una aparente relación entre la convergencia de una solución y la velocidad de salida.

7.1. Proceso de optimización automatizado

Después de realizar algunas ejecuciones del algoritmo de optimización en el ambiente *OpenFOAM*, no queda duda que un proceso automatizado es perfectamente factible. La duración de la ejecución las corridas fue variable, ya que algunas corridas solo fueron para prueba y calibración de la malla. Los tiempos oscilaron entre los 3 y los 6 días, arrojando diferentes resultados.

A continuación se describen las condiciones más sobresalientes de las dos mejores corridas:

Primera corrida: Esta corrida duró 150 ciclos, se hizo un total de 180 evaluaciones, estas se dividieron en 30 evaluaciones para la población inicial y 150 para los 150 ciclos siguientes. La variación al algoritmo genitor consistió en tomar dos indivi-

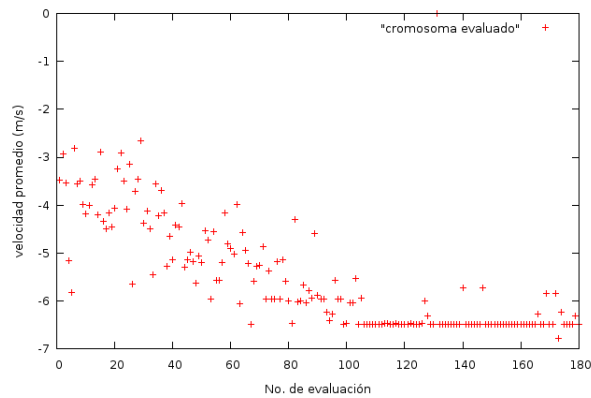


Figura 7.1: Cromosomas evaluados en la primera corrida

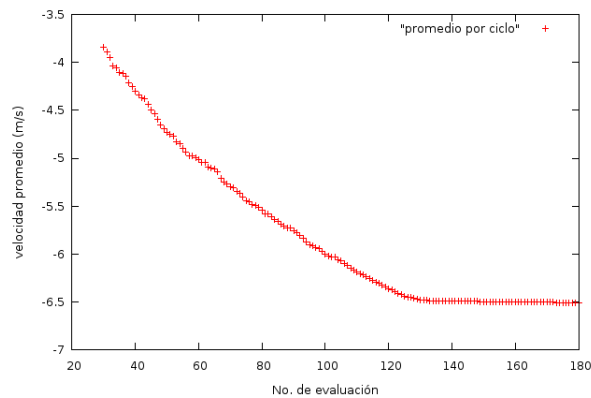


Figura 7.2: Promedio de la población en cada ciclo de la primera corrida

duos de la población mediante ranqueo de tipo cuadrático como se describe en el capítulo 6, enseguida sacamos un solo hijo como descendiente de ellos y se eliminó el peor individuo de la población. En el ciclo 143 aparece un nuevo individuo que se mantiene como el mejor hasta el ciclo 150. Este corrida tuvo una duración de 6 días, fueron 4 días hasta el ciclo 100, pero por una extensión se corrió 50 ciclos más, que fueron los 2 días adicionales. Los resultados fueron muy alentadores, pues las velocidades parecen caer en el rango correcto. Una modificación fue hecha y a partir de esta corrida se empezó con el uso de envío de correo. Esto representó un logro más porque desde fuera de nuestra escuela no se podía acceder a la máquina vía red. Ahora ya era posible saber cómo se desempeñaba el algoritmo, permitiendo tener otras ocupaciones en tanto nada malo ocurriera o

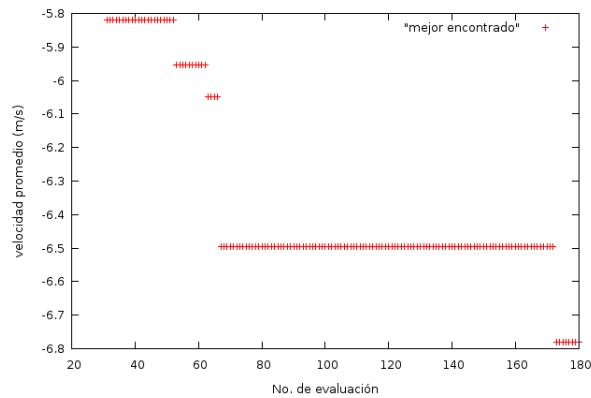


Figura 7.3: Mejores individuos encontrados en la primera corrida

en tanto el algoritmo no finalizara. La velocidad promedio obtenida por la mejor aspa en esta corrida fue de -6.78 m/s. Las figuras 7.1, 7.2 y 7.3 nos dan una idea del comportamiento de nuestro algoritmo de optimización para esta primera corrida.

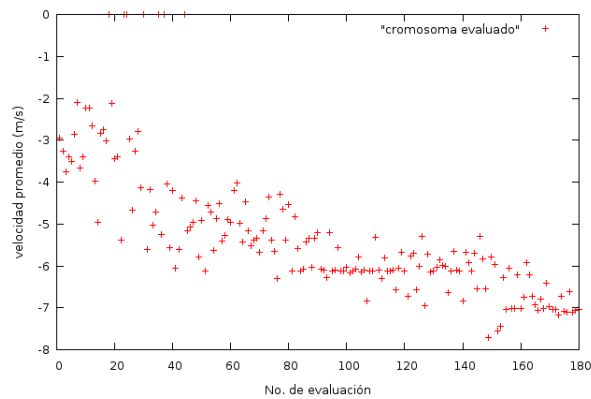


Figura 7.4: Cromosomas evaluados en la segunda corrida

Segunda corrida: Esta corrida duró 150 ciclos, se hizo un total de 180 evaluaciones, estas se dividieron en 30 evaluaciones para la población inicial y 150 para los 150 ciclos siguientes. La variación al algoritmo genitor consistió en tomar dos individuos de la población mediante ranqueo de tipo cuadrático como se describe en el capítulo 6, enseguida sacamos un solo hijo como descendiente de ellos y se eliminó el peor individuo de la población. En el ciclo 119 aparece un nuevo

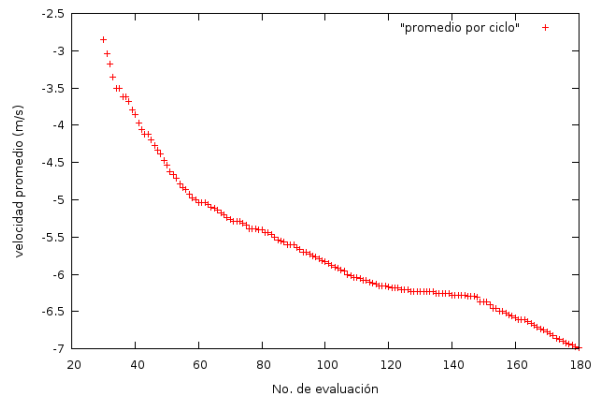


Figura 7.5: Promedio de la población en cada ciclo de la segunda corrida

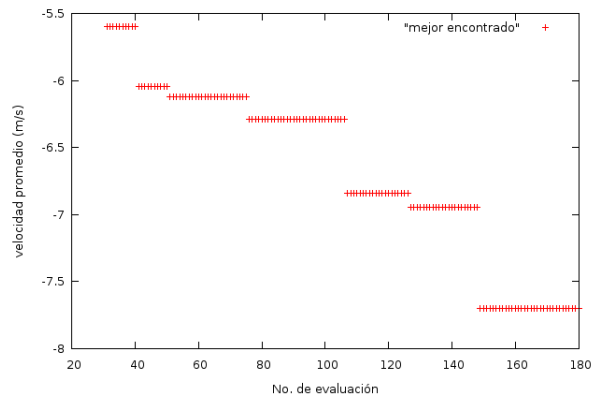


Figura 7.6: Mejores individuos encontrados en la segunda corrida

individuo que se mantiene como el mejor hasta el ciclo 150. Este corrida tuvo una duración de 6 días, fueron 4 días hasta el ciclo 100, pero por una extensión se corrió 50 ciclos más, que fueron los 2 días adicionales. Los resultados fueron los más alentadores, pues la velocidad promedio obtenida por la mejor aspa en esta corrida fue de -7.702 m/s. Las figuras 7.4, 7.5 y 7.6 nos dan una idea del comportamiento de nuestro algoritmo de optimización para esta segunda corrida.

En la figura 7.7 puede compararse el comportamiento promedio de la población en cada ciclo. Como puede observarse en éstas dos corridas hay una diferencia significativa en velocidad, lo que nos lleva a pensar que debemos continuar en la búsqueda de nuevos y mejores resultados, pero por falta de tiempo esto ya no fue posible en el presente trabajo y se dejará para el futuro.

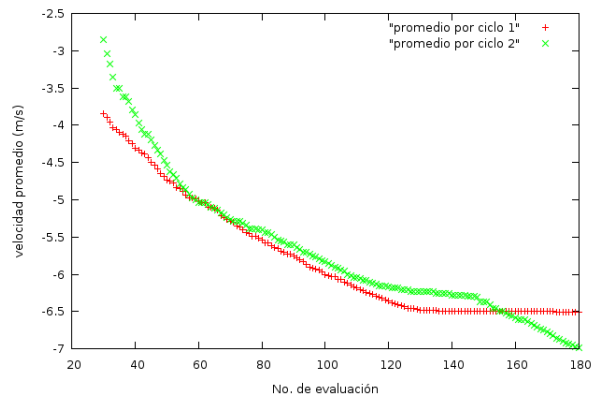


Figura 7.7: Comparación del promedio de velocidad en cada ciclo de las dos mejores corridas

7.2. Codificación del aspa

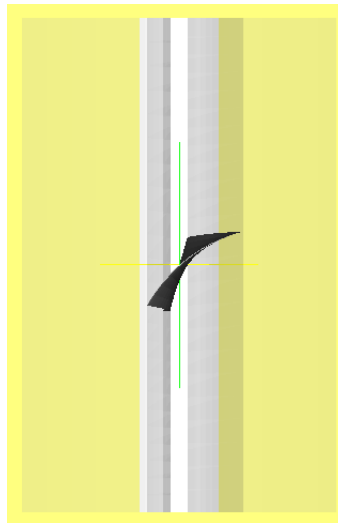


Figura 7.8: Mejor aspa encontrada: Vista hacia el sentido negativo del eje X

Otro logro importante fue la manera en que pudimos codificar el aspa para integrarla con el ambiente *OpenFOAM*. Como se recordará del capítulo número 5, un genotipo está formado por cuartetos de parámetros, esto es a_i, b_i, c_i, d_i , donde $1 \leq i \leq 3$. Una sugerencia como posible extensión es simplemente aumentar el número de puntos de 3 que son actualmente a 5, esto sería $1 \leq i \leq 5$. De igual forma es posible cambiar el conjunto de (a,b,c,d) por uno más extenso, pero puede resultar en formas tan complejas

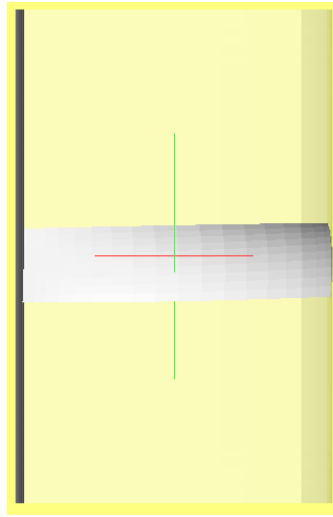


Figura 7.9: Mejor aspa encontrada: Vista hacia el sentido positivo del eje Y

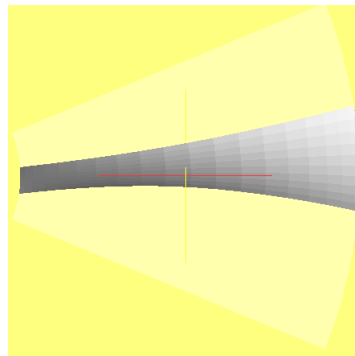


Figura 7.10: mejor aspa encontrada: Vista del aspa hacia el sentido negativo del eje Z

que el tiempo invertido no sea compensado. De cualquier manera, con el modelo aquí empleado se encontró la mejor aspa de nuestra mejor corrida, la mejor aspa encontrada corresponde al siguiente cromosoma:

{ ciclo = 119, evaluación = -7.702 m/s, genotipo = [0.333, 0.389, 0.543, 0.511, 0.418, 0.429, 0.208, 0.499, 0.529, 0.46, 0.288, 0.439] }

Los nueve puntos que definen a la mejor aspa son los siguientes:

1. (0.0247865,0.00325991,0.00778)
2. (0.025,0,0)
3. (0.0244338,-0.00529058,-0.01022)

4. (0.0616599,0.0102133,0.00858)
5. (0.0625,0,0)
6. (0.0622916,-0.00509941,-0.00998)
7. (0.09785,0.0206247,0.0092)
8. (0.1,0,0)
9. (0.0993611,-0.0112856,-0.00878)

En las figuras 7.8, 7.9 y 7.10 se puede apreciar la forma resultante para el cromosoma arriba indicado.

7.3. Comparación entre aspas

Para darnos una idea del desempeño de las aspas hemos comparado el peor con el mejor desempeño logrados en la mejor de nuestras corridas. La figura 7.11 muestra el ventilador cuyas aspas dan el peor desempeño encontrado en la mejor corrida, este ventilador tiene una velocidad promedio de -2.109 m/s. El sentido de giro de este ventilador es contrario al de las manecillas del reloj. La figura 7.12 muestra el ventilador cuyas aspas dan el mejor desempeño encontrado en la mejor corrida, este ventilador es una velocidad promedio de -7.702 m/s. Este ventilador también gira en sentido contrario al de las manecillas del reloj.

Las diferencias más notables son que la mejor aspa es más angosta, con lo cual permite un mayor flujo de aire entrante. La mejor aspa tiene una curva que abre hacia la dirección de giro, en tanto que la peor tiende a abrir hacia atrás, de esa forma la mejor aspa encausa el flujo de una mejor forma. Otra característica de la mejor aspa es una protuberancia central hacia la dirección de giro, creemos que esta forma permite al aire fluir hacia los extremos, de esta forma el aire recibirá un mayor impulso que en el centro. Otra característica que se observa es el tamaño mayor de la parte cortante

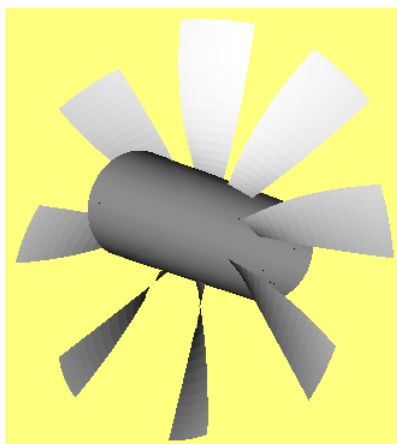


Figura 7.11: Ventilador con peor desempeño

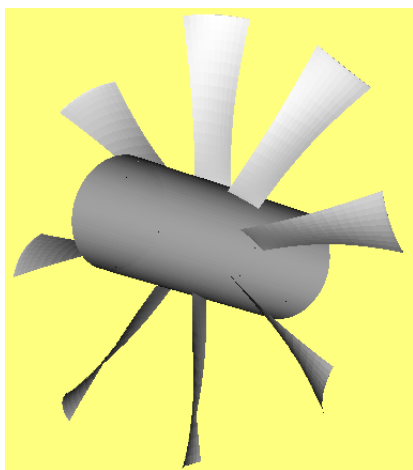


Figura 7.12: Ventilador con mejor desempeño

del aspa con respecto a la parte impulsora, esta característica se puede apreciar mejor en la figura 7.10.

7.4. Detección de errores

La detección de errores a tiempo es fundamental en este tipo de modelos, ya que cada evaluación puede durar hasta 42 minutos. De entre los principales errores que ahora podemos detectar señalaremos los dos más importantes:

Error de convergencia prematura: Este error puede ser detectado simplemente

mirando la población actual, si se observan varios individuos iguales, lo más probable es que empiece la convergencia. El detalle es que si esta convergencia es prematura, se perderá un importante tiempo al permitir que se evalúen individuos completamente idénticos. Más conveniente será entonces parar esa corrida e iniciar una nueva, o bien se pueden crear individuos aleatorios sustituyendo a los peores individuos, de tal forma que nuevamente haya variedad.

Error por ausencia del envío de correo: Puede suceder que en un momento determinado no recibamos el correo en la hora correspondiente, eso significa que un error ocurrió. Es conveniente entonces revisar nuestro algoritmo y ver que está sucediendo. Pudo suceder que ocurrió un apagón o simplemente se perdió la comunicación con nuestra red.

Existen otros errores como por ejemplo un error de mallado o divergencia de la solución, en este caso se ha preparado el orden de evaluación para que no se afecten los resultados. La manera como logramos superar este error fue poniendo las velocidades de revoluciones por minuto de mayor a menor. Así cuando hay error se evalúa al siguiente individuo con una menor velocidad, evitando así un falso mínimo.

7.5. Desempeño del algoritmo de optimización

Tras analizar lo sucedido con las primeras corridas de nuestras variaciones al algoritmo genitor, nos dimos cuenta de que aparentemente hay una relación entre el ranqueo y la generación de convergencia de nuestro modelo, por ejemplo si se desea que la convergencia sea rápida, puede elegirse ranqueo de grado superior al cuadrático que nosotros actualmente usamos. Si por el contrario se desea una convergencia lenta, es conveniente usar un ranqueo de tipo lineal.

Otra de las cosas que hay que mencionar es que debido al gran tiempo empleado para las evaluaciones, es necesario usar algoritmos evolutivos que requieran de pocas evaluaciones, así el escoger genitor quedó plenamente justificado. Cabe mencionar que es posible integrar otros algoritmos evolutivos en nuestro modelo, pero el esquema puede

variar mucho, debido a que el número de las evaluaciones en otros algoritmos puede llegar a ser elevado. De cualquier manera nunca está de más probar nuevas alternativas, sobre todo si se dispone del tiempo necesario para hacer las evaluaciones.

7.6. Relación entre convergencia y velocidad de salida

Tras analizar varios individuos con velocidades promedio elevados, nos dimos cuenta que estos individuos poseían velocidades de salida mayores que la velocidad tangencial correspondiente, nuestra suposición es que las velocidades tangenciales no deben ser superadas. Este hecho nos llevó a poner una condición de no permitir buena evaluación a los individuos cuyas velocidades superaran las velocidades tangenciales correspondientes a cada velocidad angular.

La causa probable de las divergencias se encuentra en el hecho de que cada ventilador posee una curva de trabajo y una región crítica, esta región crítica debe ser evitada a la hora de poner el ventilador en funcionamiento. Si por alguna causa para algunas velocidades se tiene trabajo en esta zona crítica, entonces es de esperarse resultados inesperados como grandes velocidades negativas fuera del rango permitido, o bien dudosas velocidades positivas. El rango de velocidades permitido es el comprendido por velocidades menores o iguales a la velocidad tangencial en la orilla del radio mayor del aspa.

Capítulo 8

Discusión

En esta sección se tratará de explicar el significado de los resultados obtenidos en el capítulo 7, también se hará mención de hasta donde se cumplieron los objetivos y las hipótesis propuestas. Además se hablará del posible trabajo futuro y del giro que puede o debe tomar nuestra investigación.

8.1. Conclusiones

En vista de los resultados obtenidos en el capítulo 7 podemos concluir que:

- Fue posible la optimización de la forma del aspa de un ventilador de flujo tub axial mediante la implementación de algoritmos genéticos haciendo uso de un software CFD libre llamado *OpenFOAM*. Además el proceso pudo ser realizado de forma automática.
- Fue posible codificar la forma de un aspa tridimensional mediante cromosomas con parámetros reales, y los puntos de control definidos pudieron modificar la forma del aspa hasta encontrar un desempeño aceptable.
- Fue posible la detección de una buena parte de errores, lo que nos permitió un uso eficiente del tiempo dedicado al proceso de optimización. Además, se encontró soluciones efectivas a esos errores.
- Fue posible detectar divergencias en la evaluación de la velocidad media del aspa. Tales divergencias pudieron ser eliminadas mediante el uso de restricciones. Estas

restricciones permitieron evitar resultados erróneos y como consecuencia se evitó una pérdida de tiempo infructuosa.

- Es recomendable utilizar algoritmos evolutivos del tipo genitor o similares, sobre todo cuando el tiempo requerido para las evaluaciones es muy grande. Cabe mencionar, que el tiempo de evaluación de un aspa es mucho mayor que el tiempo para realizar los otros pasos de nuestro proceso de optimización.

Por lo visto en los puntos anteriores nos damos cuenta que nuestras hipótesis y objetivos propuestos se han cumplido. Aún así, es necesario llevar a la práctica los conocimientos contenidos en el presente trabajo de investigación. Sería conveniente optimizar un modelo real al que se le hayan practicado pruebas de laboratorio.

8.2. Trabajo futuro

En los siguientes párrafos se exponen algunas ideas que por falta de tiempo no pudieron ser llevadas a la práctica. Estas ideas no están contempladas en el alcance de la presente tesis, pero surgieron al adentrarnos en nuestra investigación.

Como proyecto futuro es necesario trabajar el proceso de mallado de *OpenFOAM*. Entender como *OpenFOAM* ordena sus diferentes archivos referentes a la geometría de un sistema representaría uno de los logros más grandes, ya que permitiría modelar de una manera más rápida y eficiente. Sería conveniente revisar los métodos de mallado actuales con el objeto de implementar el código dentro de *OpenFOAM*. Generalizar los métodos de mallado puede ser muy útil para ahorrar tiempo cuando nos enfrentemos a modelos con geometrías distintas a las conocidas.

OpenFOAM soporta el procesado en paralelo, lo que nos permite pensar en la utilización de un *cluster*. Un *cluster* permitiría usar algoritmos distintos al genitor que también sean eficientes. Podría empezarse probando con un algoritmo genético paralelo maestro-esclavo. Un buen incremento en la velocidad de procesado puede ser logrado con el uso de varias computadoras. Estas computadoras podrían permitir usar

un mallado más fino, ya que el aumento de tiempo de procesado se compartiría entre las máquinas existentes.

La combinación del control de mallado y el empleo del *cluster* podría permitir el aumento de puntos de control. Un buen control de mallado es necesario cuando la forma del aspa se torna compleja, ya que si no se tiene un buen control seguramente se tendrán errores de mallado, lo que conduciría a resultados erróneos. Un aumento de velocidad debido a un *cluster* podría compensar el aumento del espacio de búsqueda, ya que al aumentar los puntos de control también se estaría aumentando el espacio de búsqueda.

Sería conveniente realizar un análisis detallado de las relaciones existentes entre la velocidad de salida y los ángulos de inclinación de las aspas combinados con la anchura de estas, tomando en cuenta diferentes velocidades de giro. Quizá con este análisis se logre encontrar reglas o leyes de ventiladores que ahora no se conocen.

Sobre la detección de errores, consideramos conveniente que se combine la simulación con la práctica, ya que eso daría una mayor seguridad en lo que se haga y permitiría corregir errores que sin comparación real no pueden ser detectados.

Apéndice A

POBLACION FINAL DE LA MEJOR CORRIDA

(Evaluación y puntos que definen cada aspa)

-7.702	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0248	0.025	0.0244	0.0617	0.0625	0.0623	0.0978	0.1	0.0994
Y	0.00326	0	-0.00529	0.0102	0	-0.0051	0.0206	0	-0.0113
Z	0.00778	0	-0.0102	0.00858	0	-0.00998	0.0092	0	-0.00878
-7.547	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0243	0.0618	0.0625	0.0621	0.0984	0.1	0.0995
Y	0.00461	0	-0.00586	0.00929	0	-0.00661	0.0179	0	-0.00996
Z	0.00942	0	-0.0114	0.00994	0	-0.00862	0.01	0	-0.00842
-7.439	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0984	0.1	0.0996
Y	0.00459	0	-0.00561	0.00944	0	-0.00676	0.018	0	-0.00937
Z	0.00886	0	-0.011	0.00922	0	-0.00826	0.01	0	-0.00832
-7.164	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0247	0.025	0.0245	0.0619	0.0625	0.0623	0.0983	0.1	0.0996
Y	0.00416	0	-0.00493	0.00851	0	-0.00549	0.0181	0	-0.00906
Z	0.00794	0	-0.00906	0.0107	0	-0.00908	0.0101	0	-0.00842

-7.107	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0245	0.0618	0.0625	0.0622	0.0983	0.1	0.0996
Y	0.00428	0	-0.00517	0.00902	0	-0.006	0.0182	0	-0.00918
Z	0.00818	0	-0.00984	0.0098	0	-0.00872	0.01	0	-0.00836
-7.099	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0622	0.0983	0.1	0.0996
Y	0.00432	0	-0.00524	0.00912	0	-0.00613	0.0182	0	-0.00925
Z	0.00828	0	-0.01	0.00966	0	-0.00866	0.01	0	-0.00836
-7.09	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0983	0.1	0.0995
Y	0.00447	0	-0.00557	0.00951	0	-0.00661	0.0182	0	-0.00949
Z	0.00874	0	-0.0109	0.0091	0	-0.00838	0.00996	0	-0.00834
-7.058	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00444	0	-0.00551	0.00956	0	-0.00661	0.0182	0	-0.00941
Z	0.0086	0	-0.0108	0.00894	0	-0.00834	0.00994	0	-0.00832
-7.047	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00445	0	-0.00549	0.00956	0	-0.00664	0.0182	0	-0.00933
Z	0.00856	0	-0.0108	0.00888	0	-0.0083	0.00994	0	-0.00832
-7.041	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00444	0	-0.00548	0.00958	0	-0.00664	0.0182	0	-0.00933
Z	0.00854	0	-0.0108	0.00886	0	-0.0083	0.00994	0	-0.0083
-7.036	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00446	0	-0.00547	0.00958	0	-0.00666	0.0182	0	-0.00929
Z	0.00854	0	-0.0108	0.00884	0	-0.00828	0.00994	0	-0.0083

-7.027	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00446	0	-0.00546	0.00958	0	-0.00666	0.0182	0	-0.00925
Z	0.00852	0	-0.0108	0.00882	0	-0.00826	0.00994	0	-0.0083
-7.026	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00448	0	-0.00547	0.00956	0	-0.00669	0.0182	0	-0.00925
Z	0.00852	0	-0.0108	0.00882	0	-0.00822	0.00996	0	-0.00828
-7.023	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00441	0	-0.00545	0.00961	0	-0.00661	0.0183	0	-0.00933
Z	0.00846	0	-0.0108	0.0088	0	-0.0083	0.00992	0	-0.0083
-7.021	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.0045	0	-0.00546	0.00956	0	-0.00671	0.0181	0	-0.00922
Z	0.00852	0	-0.0108	0.0088	0	-0.00818	0.00996	0	-0.00828
-7.02	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00441	0	-0.00545	0.00961	0	-0.00661	0.0183	0	-0.00929
Z	0.00846	0	-0.0107	0.00878	0	-0.0083	0.00992	0	-0.0083
-7.015	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0618	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00449	0	-0.00544	0.00958	0	-0.00671	0.0182	0	-0.00918
Z	0.00848	0	-0.0108	0.00876	0	-0.00818	0.00996	0	-0.00828
-6.967	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0245	0.0617	0.0625	0.0622	0.0983	0.1	0.0996
Y	0.00432	0	-0.00509	0.01	0	-0.00659	0.0185	0	-0.00867
Z	0.0077	0	-0.0103	0.00766	0	-0.008	0.00992	0	-0.00824

-6.945	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0245	0.0617	0.0625	0.0621	0.0982	0.1	0.0996
Y	0.00422	0	-0.00502	0.00992	0	-0.00661	0.0187	0	-0.00867
Z	0.00748	0	-0.0101	0.0076	0	-0.00796	0.00982	0	-0.00818
-6.93	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0247	0.025	0.0244	0.062	0.0625	0.0621	0.0981	0.1	0.0995
Y	0.00375	0	-0.00551	0.00788	0	-0.00705	0.0193	0	-0.00984
Z	0.0076	0	-0.00998	0.0103	0	-0.0082	0.00896	0	-0.00752
-6.838	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0245	0.025	0.0243	0.0618	0.0625	0.0621	0.0985	0.1	0.0995
Y	0.0048	0	-0.00594	0.00917	0	-0.00683	0.0175	0	-0.00976
Z	0.00964	0	-0.0115	0.0101	0	-0.00842	0.0101	0	-0.00838
-6.838	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0245	0.025	0.0243	0.0618	0.0625	0.0621	0.0985	0.1	0.0995
Y	0.0048	0	-0.00594	0.00917	0	-0.00683	0.0175	0	-0.00976
Z	0.00964	0	-0.0115	0.0101	0	-0.00842	0.0101	0	-0.00838
-6.794	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0617	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00441	0	-0.00536	0.00966	0	-0.00666	0.0183	0	-0.0091
Z	0.00826	0	-0.0106	0.0085	0	-0.00816	0.00992	0	-0.00826
-6.756	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0244	0.0617	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00436	0	-0.00528	0.00973	0	-0.00664	0.0184	0	-0.00902
Z	0.00808	0	-0.0105	0.0083	0	-0.00814	0.0099	0	-0.00824
-6.718	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0247	0.025	0.0244	0.0617	0.0625	0.0622	0.098	0.1	0.0994
Y	0.00367	0	-0.00536	0.01	0	-0.00564	0.0198	0	-0.0106
Z	0.00804	0	-0.0104	0.00868	0	-0.0094	0.00946	0	-0.00862

-6.717	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0247	0.025	0.0244	0.0617	0.0625	0.0622	0.098	0.1	0.0994
Y	0.00369	0	-0.00536	0.01	0	-0.00566	0.0198	0	-0.0106
Z	0.00804	0	-0.0104	0.00868	0	-0.00936	0.00948	0	-0.0086
-6.714	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0241	0.0616	0.0625	0.0621	0.0986	0.1	0.0988
Y	0.00426	0	-0.0065	0.0108	0	-0.00744	0.0167	0	-0.0158
Z	0.00926	0	-0.0102	0.0107	0	-0.00958	0.0108	0	-0.00968
-6.639	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0241	0.0616	0.0625	0.062	0.0985	0.1	0.0987
Y	0.00431	0	-0.0065	0.0107	0	-0.00776	0.0172	0	-0.0162
Z	0.00926	0	-0.0103	0.0104	0	-0.0099	0.0104	0	-0.00956
-6.614	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0247	0.025	0.0245	0.0617	0.0625	0.0622	0.0981	0.1	0.0995
Y	0.00398	0	-0.00516	0.0101	0	-0.0061	0.0192	0	-0.00953
Z	0.00772	0	-0.0103	0.00796	0	-0.00864	0.00968	0	-0.00842
-6.567	P1	P2	P3	P4	P5	P6	P7	P8	P9
X	0.0246	0.025	0.0245	0.0617	0.0625	0.0621	0.0983	0.1	0.0996
Y	0.00429	0	-0.00514	0.00983	0	-0.00664	0.0186	0	-0.00879
Z	0.00774	0	-0.0103	0.0079	0	-0.00802	0.00986	0	-0.0082

Bibliografía

- [1] *OpenFOAM The Open Source CFD Toolbox User Guide Version 1.5*, July 2008.
- [2] *Sitio oficial de OpenFOAM*, 2009. <http://www.opencfd.co.uk/openfoam>.
- [3] W. J. Arcos. *Optimización del diseño de un abanico de un motor de corriente alterna para mejorar su eficiencia mediante algoritmos genéticos*. Tesis de maestría, ITESM, 2006.
- [4] R. K. Bock and W. Krischer. *The data analysis briefbook*. Springer, 1998.
- [5] K. DeJong. *An Analysis of the Behavior of a Class of Genetic Adaptive System*. Phd dissertation, Univ. of Michigan, Ann Arbor, 1975.
- [6] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
- [7] Greenheck. *Fundamentos de Ventilación*, agosto 2007.
- [8] J. Holland. *Adaptation In Natural and Artificial Systems*. University of Michigan Press, 1975.
- [9] J. R. Koza et al. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers, 2003.
- [10] L. F. Mantilla and J. A. Cardona. *Gestión energética de los motores eléctricos: mejora de la eficiencia de los accionamientos con el uso de variadores de velocidad*. Departamento de Ingeniería Eléctrica y Energética E.T.S.I.I. y T., Universidad de Cantabria, 2005.

- [11] E. M. Marshall and A. Bakker. *Computational Fluid Mixing*. Fluent Inc., 10 Cavendish Court, Lebanon, NH 03766, USA, 2003. <http://www.fluent.com>.
- [12] S. Orozco. *Algoritmos Genéticos*. Escuela de Ingeniería en Ciencias y Sistemas, Universidad de San Carlos de Guatemala, Guatemala, Febrero 2007.
- [13] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Taylor and Francis, 1980.
- [14] G. Rawlins. *Foundations of Genetic Algorithms*. Morgan Kaufmann, 1991.
- [15] G. Syswerda. Uniform crossover in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 2–9, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.
- [16] U.S. Department of Energy Energy Efficiency and Renewable Energy. *Improving Fan System Performance: A sourcebook for industry*, April 2003. Air Movement and Control Association International.
- [17] C. H. Villarreal. Optimization of the savonius wind turbine using a genetic algorithm. Master’s thesis, ITESM, December 2008.
- [18] D. Whitley. The genitor algorithm and selective pressure. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 116–121, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.
- [19] D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4:65–85, 1994.
- [20] B. Yuan. Deterministic crowding, recombination and self-similarity. In *CEC ’02: Proceedings of the Evolutionary Computation on 2002. CEC ’02. Proceedings of the 2002 Congress*, pages 1516–1521, Washington, DC, USA, 2002. IEEE Computer Society.