



UNIVERSIDAD DE ANTIOQUIA

Análisis Estadístico de Fractura Mecánica y Optimización de Inspecciones en Aeronaves Pequeñas

Gustavo Alberto Giraldo Echeverri

Trabajo de grado presentado para optar al título de Ingeniero Mecánico

Asesor externo

PhD. Juan David Ocampo de los Ríos

Saint Mary's University, San Antonio, TX, EEUU

Asesor interno

PhD. (c) Liliana Marcela Bustamante Goez

Universidad de Antioquia, Antioquia, Colombia

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería Mecánica

Medellín, Antioquia, Colombia

2022

Cita	Giraldo Echeverri [1]
Referencia	[1] G.A. Giraldo Echeverri “Análisis estadístico de fractura mecánica y optimización de inspecciones en aeronaves pequeñas”, Trabajo de grado profesional, Ingeniería Mecánica, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2022.
IEEE (2020)	



Centro de Documentación Ingeniería (CENDOI)

Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Chancellor: John Jairo Arboleda Céspedes.

Dean: Jesús Francisco Vargas Bonilla.

Jefe departamento: Pedro León Simancas.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

Dedicatoria

A mi madre Luz Marleny Echeverri Gómez, quien siempre me apoyó con amor incondicional a pesar de todas las adversidades y me ayudo a persistir en este camino. A mi padre en el cielo, Ernesto Giraldo Parrado, quien me inspiró a amar la tecnología e inculcó en mí, los valores que me hacen ser la persona de hoy. A mi hermano, Mario Ernesto Giraldo Echeverri, quien es mi compañero de vida. También a mi abuela Mercedes Gómez, a mis tías Lucia Gómez e Inés Gómez, por siempre estar pendientes de mi proceso, darme ese ejemplo de sabiduría y rectitud y por ayudarme siempre en todo lo que quiero hacer. Los amo a todos.

Agradecimientos

El autor de este trabajo esta agradecido con los profesores Pedro León, Liliana Bustamante, Dr. Juan Ocampo and Dr. Harry Millwater, quienes hicieron que este trabajo fuera posible, me guiaron con paciencia para alcanzar los objetivos trazados y me dieron su concejo para seguir en el campo de la investigación.

Table of Contents

I.	RESUMEN.....	8
II.	ABSTRACT.....	9
III.	INTRODUCTION.....	10
IV.	OBJETIVOS	12
	A. Objetivo General.....	12
	B. Objetivos Específicos.....	12
V.	MARCO TEORICO.....	13
	A. Mantenimiento aeronáutico.....	13
	B. Tipos de Mantenimiento.....	14
	1. Mantenimiento Correctivo	14
	2. Mantenimiento de mejora.....	15
	3. Mantenimiento preventivo	15
	C. Distribuciones de probabilidad.....	17
	1. Distribution Normal	18
	2. Distribución Log-Normal.....	20
	3. Distribución Weibull -Valor extremo.....	21
	4. Distribución Gumbel – Valor extremo.....	21
	D. Selección de la distribución de probabilidad.....	23
	E. Análisis de tolerancia al daño usando probabilística - Probabilistic Damage Tolerance Fundamentals (PDTA).....	24
	1. Esfuerzo residual.....	24
	2. Probabilidad de falla – Probability of Failure (POF)	25
	3. Muestreo de Monte Carlo.....	25
VI.	PLANTEAMIENTO DEL PROBLEMA	27
VII.	METODOLOGIA	30
	A. Análisis del Código base en MATLAB	30
	B. Aporte al código de optimización de inspecciones en MATLAB – Optimización del tiempo de ejecución.....	37
VIII.	RESULTADOS Y DISCUSIÓN.....	39
IX.	CONCLUSIONES	45
X.	REFERENCIAS	46

LISTA DE FIGURAS

Fig.1. Crecimiento de grieta en Boing 747 – Johnson Space Center	11
Fig.2. Mantenimiento Aeronáutico [14]	13
Fig.3. Tipos de Mantenimiento.....	14
Fig.4. Mantenimiento basado en riesgo.....	17
Fig.5. Distribuciones de probabilidad Comunes en la Ingeniería[4]	18
Fig.6. Distribución Normal [4]	19
Fig.7. Comparación entre CDF y PDF [4]	19
Fig.8. Comparación entre distribuciones Log-Normal y Normal [4]	20
Fig.9. Distribución Weibull para diferentes valores de β y α [4]	21
Fig. 10. Función de densidad de probabilidad para valor extremo [4]	22
Fig. 11. Función de distribución acumulada para valor extremo [4]	23
Fig.12. Selección de distribuciones para un conjunto de datos [15]	23
Fig.13. Esquema de PDTA [15]	24
Fig.14. Curva de esfuerzo residual [15].....	25
Fig.15. Comparación del POF con Inspecciones y sin Inspecciones [6]	26
Fig.16. Jet Corporativo [15]	27
Fig.17.Wing Spar [15]	27
Fig.18. Graphical User Interface (GUI) SMART-DT [6].....	28
Fig.19. Grafico generado en SMART-DT, Flights vs Probability of failure [6]	28
Fig.20. Diagrama de Flujo para el Primer Estado Del Código en Matlab [16].....	30
Fig.21. Obteniendo Programa de Inspecciones, Primer Estado del Código [16].....	31
Fig.22. Combinaciones de inspecciones con POF inviables de la tabla 1 [16].....	32
Fig.23. Combinaciones de inspecciones con POF inviables de la tabla 2[16].....	33
Fig.24. Grafica antes de aplicar el algoritmo de salto [16].....	34
Fig.25. Grafica después de aplicar el algoritmo de salto [16].....	35
Fig.26. Diagrama de flujo del código base en MATLAB [16].....	36
Fig.27. Archivo .Dat	37

Fig.28. Primera combinación de inspecciones encontrada.....	41
Fig.29. Segunda combinación de inspecciones encontrada.....	41
Fig.30. Combinación de inspecciones seleccionada.....	43
Fig.31. Resultados obtenidos con el código en MATLAB.....	44
Fig.32. Resultados obtenidos con el código en Octave.....	44

LISTA DE TABLAS

Tabla 1. Numero de Combinaciones de Inspecciones Posibles [16].....	31
Tabla 2. Número de Combinaciones de Inspecciones Posibles [16].....	33
Tabla 3. Datos de salida SMART-DT.....	38
Tabla 4. Datos de entrada, fractura mecánica, Handbook – Código sin optimizar	39
Tabla 5. Datos de entrada tipo de inspecciones – Código sin optimizar.....	39
Tabla 6. Datos de entrada Handbook, nivel de riesgo y posibles tiempos de inspección – Código sin optimizar	39
Tabla 7. Datos de entrada Handbook, nivel de riesgo posibles tiempos de inspecciones – Código optimizado.....	40
Tabla 8. Datos de entrada Handbook, fractura mecánica – Código optimizado.....	41
Tabla 9. Datos de entrada Handbook, inspecciones – Código optimizado.....	41
Tabla 10. Datos de entrada Handbook, nivel de riesgo posibles tiempos de inspecciones – Código optimizado.....	41

I. RESUMEN

En la programación de las operaciones de mantenimiento de una aeronave, se consolida como un proceso clave que tiene como objetivo garantizar la máxima seguridad durante la vida útil de la aeronave y mantener los costos operativos lo más bajos posible. Para establecer un programa de inspección y las reparaciones posteriores, es importante evaluar las variaciones en la carga, el material y la geometría; por lo tanto, es esencial un análisis probabilístico de tolerancia al daño (PDTA, por sus siglas en inglés) eficiente en combinación con métodos de optimización avanzados.

Fue necesario contar con la sugerencia de expertos para llegar a consideraciones reales de la industria aeronáutica sobre el mantenimiento: programar diferentes técnicas de inspección según el tiempo que estas se necesiten y los posibles tiempos en los que se podían aplicar estas técnicas. Además, para encontrar la mejor combinación de inspecciones, fue muy importante trabajar con base a un código que proporcionaba una combinación de inspecciones optimizada, sin embargo, persistía un problema, era computacionalmente ineficiente, debido al largo tiempo de ejecución [16]. Teniendo en cuenta las consideraciones encontradas, el nivel de riesgo y los costos, se desarrolló un algoritmo que fue capaz de reducir el tiempo de ejecución del código y a su vez proporcionar la mejor combinación de inspecciones posible. Esta investigación proporcionó la mejor combinación de inspecciones con un tiempo de ejecución reducido, que mantienen bajo el riesgo de falla del avión mientras el costo de la combinación es el menor posible, esto gracias a todas las consideraciones mencionadas anteriormente. También este código se implementó en un lenguaje de programación de libre (Octave).

***Palabras clave-* Probabilidad de falla, Inspecciones, Riesgo, Mantenimiento Predictivo.**

II. ABSTRACT

Scheduling aircraft maintenance operations is a key process that aims to ensure the highest safety during the lifetime of aircraft while keeping the operational costs as low as possible. To establish an inspections schedule and the subsequent repairs, it is important to assess variations in loading, material, and geometry; therefore, an efficient probabilistic damage tolerance analysis (PDTA) in combination with advanced optimization methods is essential.

Having the suggestions of experts allowed to get real considerations of the aeronautical industry on maintenance, such as: programming different inspection techniques according to the time they are needed and the possible times in which these techniques could be applied. In addition, to find the best combination of inspections, was very important to work based on a code that already gave an optimized combination of inspections, however, that code had a problem, it that was computationally inefficient, due to the long run time [16]. Taking into account the considerations found, the level of risk and the costs, an algorithm was developed to be able to reduce the execution time of the basis code and at the same time keep providing the best possible combination of inspections. This research provided the best combination of inspections with the run time reduced, keeping the risk of aircraft failure low while the cost is the lowest possible, thanks to all the considerations mentioned above and. This code was implemented in a free programming language (Octave).

***Keywords* – Probability of failure (POF), Inspections, risk, Predictive Maintenance**

III. INTRODUCTION

La industria aeronáutica es una de las más grandes del mundo, gracias a esta se logran transportar mercancías y personas en un corto periodo de tiempo en comparación con los otros medios de transporte, como el transporte marítimo y terrestre, sin embargo, es una de las más costosas, entre varias razones por que se debe garantizar la confiabilidad de cada una de las aeronaves durante su vida útil, siendo este un proceso riguroso en el que el mantenimiento es la mayor fuente de costos luego de la compra de las aeronaves; por esto el mantenimiento es clave para reducir los costos operativos y alargar la vida útil de las aeronaves.

Actualmente esta industria trabaja con base en probabilística para predecir el riesgo de falla de los sistemas. Se sabe que a medida que la aeronave acumula horas de vuelo, va sufriendo de fatiga estructural al experimentar los ciclos de carga que normalmente se experimentan en una operación habitual, debido a esto, conforme pasa el tiempo de vida útil de la aeronave, el riesgo de falla por fractura mecánica aumenta, por lo tanto, se busca reducir este riesgo a través de inspecciones y sus posteriores reparaciones, no obstante, entre más inspecciones se hagan más costoso va a ser el mantenimiento. Es por esta razón que es necesario utilizar métodos de optimización para el programa de inspecciones, lo cual facilitara la determinación de cuándo es necesario realizar inspecciones obligatorias con el fin de reducir costos y así disminuir la probabilidad de falla de la aeronave.

Para realizar un análisis probabilístico se necesita de muchos datos con el fin de obtener distribuciones estadísticas más precisas y que modelen de mejor manera el comportamiento de las estructuras analizadas, por esto es que este análisis es susceptible a la cantidad de datos que se puedan obtener de una aeronave, sabiendo esto, en este trabajo se analizó un Jet corporativo del cual se tiene una base de datos de una de sus partes estructurales (Wing spar) la cual es la más crítica por su probabilidad de falla, se saben sus características geométricas, así como las propiedades del material y el crecimiento de grieta según sus cargas ya que se han documentado varios fallos en esta parte en específico.

Una vez que se tienen los datos estadísticos de la aeronave, se procede a aplicar el análisis probabilístico de tolerancia al daño - PDTA (Por sus siglas en inglés - Probabilistic damage tolerance analysis) con el fin de encontrar la probabilidad de falla (POF – Probability Of Failure) para esta pieza estructural y de este modo tener una certeza de cuando se debe programar una inspección antes de que el riesgo sobrepase el límite definido por la FAA (Federal Aviation Administration) el cual está definido en 10^{-7} , para realizar este análisis fue fundamental el uso del programa SMART-DT.

SMART-DT (Small Aircraft Risk Technology – Damage Tolerance) es un programa patrocinado por la FAA de libre acceso, con el cual se pueden obtener las probabilidades de falla (POF) de un elemento estructural analizado según sus valores de entrada, tales como el tamaño de grieta inicial, tamaño de grieta reparada, tenacidad a la fractura entre otros [6]. La

POF puede cambiar si se adicionan inspecciones, tal cómo se mencionó anteriormente. Luego de obtener esta probabilidad de falla se debe generar un algoritmo capaz de programar inspecciones para poder disminuir el nivel de riesgo manteniéndolo en un nivel seguro y además que haga un filtro de costos para buscar la combinación más económica.

Teniendo en cuenta estas dos importantes consideraciones, nivel de riesgo y bajo costo, el algoritmo para encontrar el mejor calendario de inspecciones está programado en el código base de MATLAB, en donde por medio de la creación de un archivo .dat con toda la información del avión, la cual es la información de entrada al programa SMART-DT, se ejecuta el programa desde MATLAB tantas veces como combinaciones de inspecciones se tengan [16]. La cantidad de combinaciones depende de cuantos posibles tiempos de inspección se tienen en la vida útil de la aeronave y cuantas técnicas de inspección se van a evaluar en estos tiempos, también y de mucha importancia, son las restricciones encontradas en la literatura y recomendaciones de expertos para acotar la cantidad de combinaciones de inspecciones y así tener un código más eficiente y que no sea tan costoso computacionalmente.

Se puede observar en la figura 1, que hay evidencia de crecimiento de grieta alrededor del remache señalado en la zona del tren de aterrizaje. Luego de la inspección es necesario realizar una reparación para de esta forma disminuir la probabilidad de falla de la aeronave.



Fig. 1. Crecimiento de grieta en Boeing 747 – Johnson Space Center

IV. OBJETIVOS

A. Objetivo General

Desarrollar un algoritmo para optimizar el tiempo de ejecución de un código base en MATLAB, el cual es capaz de generar programas de inspección optimizados basándose en métodos probabilísticos los cuales están enfocados en el análisis de riesgo de las estructuras aeroespaciales. Se consideran los siguientes factores: cargas experimentadas, probabilidad de falla (POF), propiedades de los materiales, geometría y costos.

B. Objetivos Específicos

- Encontrar consideraciones reales aplicables al mantenimiento aeronáutico basado en referencias de expertos en el tema.
- Optimizar el tiempo de ejecución del código por medio de las consideraciones encontradas y a través de un algoritmo.
- Implementar la metodología en MATLAB.
- Validar y verificar la metodología implementada, comparando los resultados y tiempos de ejecución obtenidos al implementar el algoritmo de optimización de tiempo de ejecución en el código base en MATLAB, con los resultados obtenidos antes de la implementación del algoritmo.
- Implementar la metodología en un lenguaje de programación libre (Octave), para su fácil reproducción y uso.

V. MARCO TEORICO

A. Mantenimiento aeronáutico

El mantenimiento es un conjunto de acciones que tiene como objetivo mantener (preservar) en óptimas condiciones un artículo o máquina, también se puede implementar con el objetivo de regresar ese artículo a sus condiciones originales. En la industria en general (así como en la aeronáutica) se utiliza para poder garantizar la seguridad en las operaciones que realizan las máquinas (aeronaves), así como para prolongar su vida útil y evitar su desgaste prematuro. Es de gran importancia entender la necesidad de cada uno de los sistemas analizados entorno al mantenimiento, pues esto puede ahorrar costos y de esta forma se puede garantizar la viabilidad del proceso de mantenimiento que se está implementando en el sistema analizado.

A modo de ejemplo, en la industria minera se tienen sistemas que a pesar de realizar en ellos un mantenimiento riguroso, su vida útil no se ve significativamente afectada, por lo que realizar estas operaciones solo generan costos innecesarios y un pequeño aumento en su confiabilidad que puede ser despreciable. En estos casos simplemente se podría realizar su reparación en caso de que este falle lo que sería mucho más económico para la empresa que necesita este sistema y requiere economizar gastos. Sin embargo, no se puede generalizar este ejemplo a todas las industrias, por eso es de vital importancia un correcto análisis de cada sistema, con el objetivo de implementar las operaciones cuando sean necesarias, en este punto es donde se aplican los datos estadísticos que se puedan obtener del sistema analizado, entre más datos se tengan mucho mejor y más aún en la industria aeronáutica en la cual realizar un buen proceso de mantenimiento afecta significativamente la vida útil de los sistemas y a su vez su la seguridad de su operación, sin embargo también aumenta significativamente los gastos en los que incurre la empresa que opera las aeronaves o los equipos aeronáuticos.[14].



Fig.2. Aeronautical maintenance reference [14]

B. Tipos de Mantenimiento

En la industria debido a la necesidad de implementar un plan de mantenimiento adecuado a cada caso en el que un sistema lo requiere, han surgido diferentes tipos de mantenimiento, cada uno con un mismo propósito que es el de mantener en correcto y óptimo funcionamiento un sistema, pero con un enfoque diferente, a continuación, se muestra un mapa en la figura .3. con una simplificación de los tipos de mantenimiento, en verde se puede apreciar el tipo de mantenimiento que es en el que se enfocó este trabajo, más adelante se dará la explicación por la cual este es un tipo de mantenimiento apropiado para este fin.

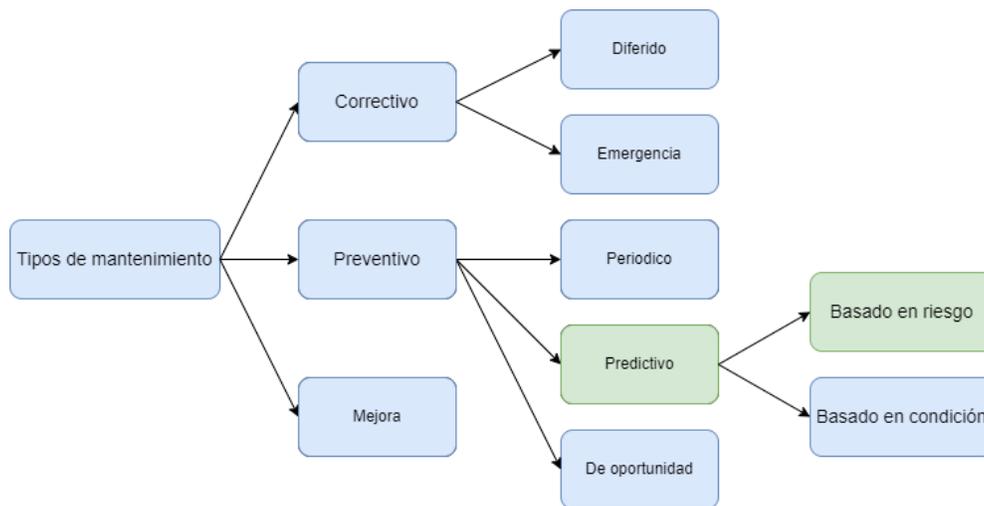


Fig.3. Tipos de Mantenimiento

1. Mantenimiento Correctivo

Es básicamente un conjunto de tareas que se implementan en un sistema o equipo en el cual es necesario intervenir para que este pueda seguir con su tarea, se puede hacer ya sea una reparación o un reemplazo de una pieza.

En este tipo de mantenimiento no es necesario seguir un calendario o plan de mantenimiento. Uno de los mayores problemas de este tipo de mantenimiento es la alta probabilidad de que al momento en que un equipo falle no se cuente con la pieza para su reparación, lo que trae altos costos en la productividad de una empresa.

Se tienen muchos casos en los que el mantenimiento correctivo es necesario e inevitable, pero aplicar solo este tipo de mantenimiento hace que se genere un mayor impacto financiero en las empresas y más aún si esta máquina es altamente requerida para seguir con la producción de un sistema. Lo importante para resaltar aquí es que una gran cantidad de

fallos se pueden evitar si se hace un correcto análisis del sistema una posterior aplicación de un mantenimiento predictivo.

El mantenimiento correctivo se puede subdividir en dos categorías, la primera es mantenimiento correctivo diferido, este agrupa las acciones de mantenimiento preventivo y correctivo que no se hicieron en su debido tiempo y por ende se hacen posteriormente. En muchas ocasiones no se debe necesariamente a negligencia del personal de mantenimiento, esta puede ser una estrategia de costos, que es muy utilizada cuando los recursos designados al mantenimiento son limitados. Es necesario aclarar que no es una estrategia recomendada, ya que a largo o mediano plazo puede aumentar los gastos de la empresa, adicionalmente genera una mayor probabilidad de falla de los equipos sometidos a este tipo de mantenimiento y por último daña la reputación de la empresa.

Por otro lado, se tiene el mantenimiento correctivo de emergencia o no planificado. Este se da cuando se genera el fallo de un equipo o sistema y este requiere ser corregido con urgencia, es decir se deben tomar decisiones en el menor tiempo posible. Este tipo de mantenimiento se aplica en casos muy específicos en lo que la probabilidad de falla del equipo es muy baja y su falla no implica una pérdida significativa de la productividad de la empresa, también se da el caso en el que se aplica por bajos recursos y falta de conocimiento en mantenimiento preventivo.

2. Mantenimiento de mejora

Este tipo de mantenimiento se enfoca en analizar el sistema o equipo con el objetivo de mejorar su mantenibilidad. La mantenibilidad de un equipo se puede entender como aquellas características del sistema analizado que permiten una fácil intervención de este mismo para su reparación o mantenimiento.

A modo de ejemplo se pueden implementar algunas tareas tales como: Mejorar la disponibilidad de los sistemas, con el objetivo de que si falla un equipo de inmediato se pueda poner en funcionamiento el siguiente. También se puede aumentar la fiabilidad de los equipos, optimizar los gastos con los que se realizan las operaciones anteriores.

3. Mantenimiento preventivo

El mantenimiento preventivo es el tipo de mantenimiento más ampliamente utilizado en el campo industrial y su aplicación tiene grandes beneficios para la vida útil de los equipos y sistemas, también puede mejorar la productividad y muy importante puede reducir costos de mantenimiento. Para implementar este mantenimiento generalmente se hace el uso de datos históricos de los equipos a los cuales se les va a hacer el mantenimiento o contando con el factor de oportunidad que tengan las empresas en los que

se aplique, por esto el mantenimiento preventivo se subdivide en tres categorías las cuales se explican a continuación.

En primer lugar, se tiene el mantenimiento preventivo con intervalos constantes de tiempo o periodos en los que se van a hacer las acciones requeridas, un buen ejemplo de este tipo de mantenimiento la revisión técnico mecánica obligatoria que se le deba dar a un carro para que este pueda circular.

Luego se tiene el mantenimiento preventivo de oportunidad, este hace uso de los datos de producción de la empresa para programar las acciones requeridas cuando esta no esté en producción, a modo de ejemplo muchas empresas frenan su producción un domingo de cada mes o por ejemplo el Metro de Medellín, Antioquia Colombia frena su actividad durante las noches, oportunidad que se aprovecha para programar el mantenimiento e implementarlo.

Por último, se tiene el mantenimiento preventivo predictivo, este utiliza técnicas más avanzadas de mantenimiento en la que el uso de la estadística es fundamental para una correcta aplicación. Este en una forma simplificada se subdivide en dos categorías, mantenimiento preventivo predictivo basado en riesgo y mantenimiento preventivo predictivo basado en condición. El basado en la condición genera un programa de mantenimiento sobre la condición actual y futura del equipo analizado.

Y el más importante para este trabajo es el mantenimiento predictivo basado en riesgo, el cual por medio de una gran base de datos que se genera a lo largo del uso de un equipo o sistema, genera distribuciones de probabilidad, las cuales con un correcto análisis de los factores involucrados en el sistema (por ejemplo un avión y sus cargas aerodinámicas, estáticas y también sus horas de vuelo) se puede generar una distribución con la probabilidad de falla, el nivel de riesgo de este equipo o su contraparte se puede determinar la confiabilidad de este equipo. Con estos datos y también los datos estadísticos que entrega cada inspección y su debida reparación, que básicamente lo que hacen es disminuir la probabilidad de falla de la aeronave en el tiempo que se aplica; con estos datos se puede simular la vida útil del equipo y programar un óptimo plan de mantenimiento para este.

A continuación, se explicarán los fundamentos estadísticos básicos para realizar este análisis de mantenimiento preventivo predictivo basado en el riesgo y se muestra en la figura 4. Un breve diagrama de cómo se analizan los datos para generar el análisis estadístico.

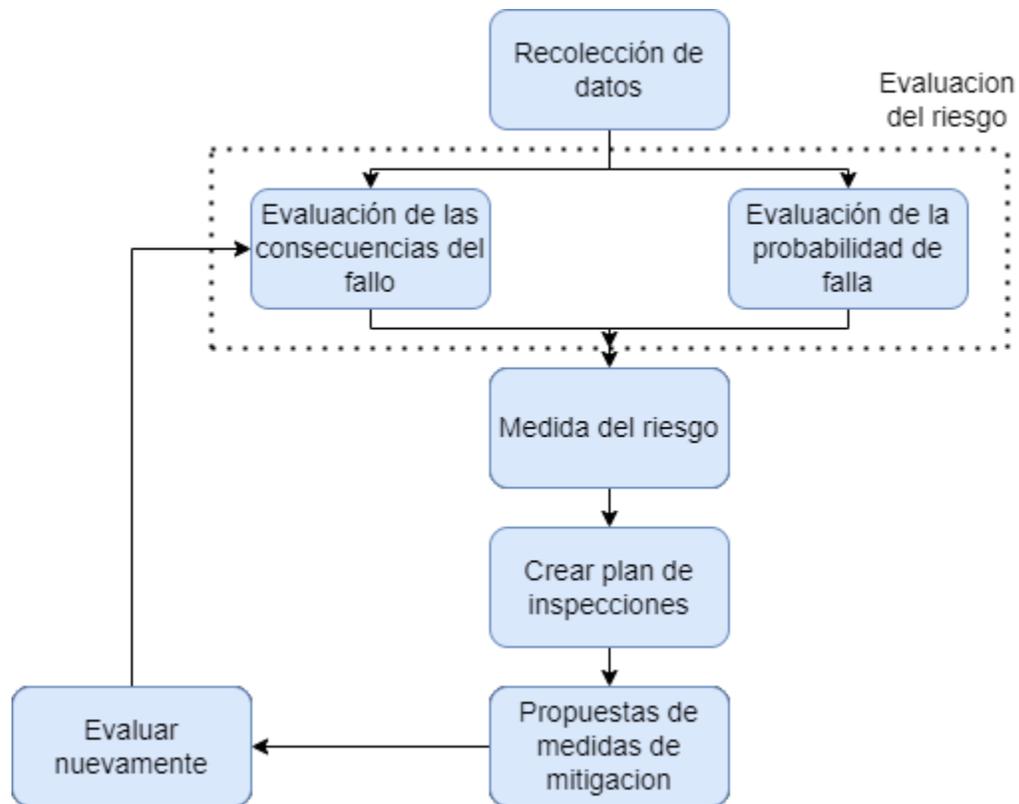


Fig.4. Mantenimiento basado en riesgo

C. Distribuciones de probabilidad

Las distribuciones de probabilidad son una gran herramienta que permite visualizar la probabilidad de que se repitan ciertos eventos de un experimento y con estas se pueden modelar escenarios de eventos futuros considerando factores actuales de estos fenómenos.

En la ingeniería por lo general se utilizan cinco distribuciones de probabilidad por sus diferentes características que permiten modelar determinadas situaciones, al igual que en este trabajo, estas son: distribución Normal, distribución Log-Normal, distribución Weibull y distribución Gumbel mostrados en la figura 5. A continuación, se explicará cada una de estas distribuciones y sus características que las hacen óptimas para análisis de ingeniería.

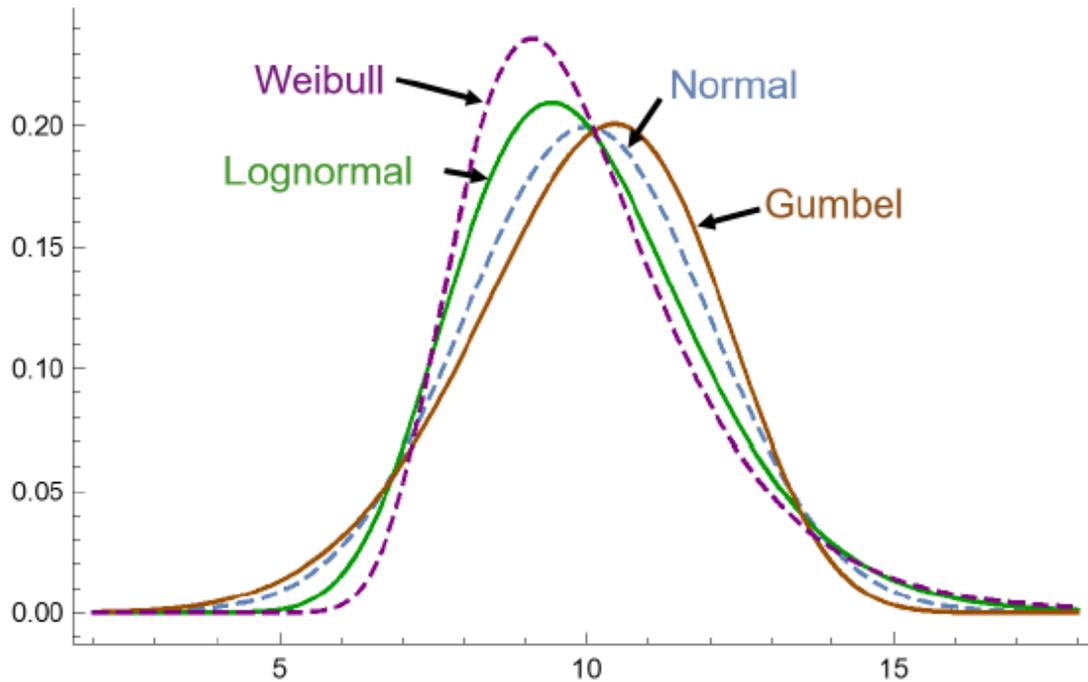


Fig.5. Distribuciones de probabilidad Comunes en la Ingeniería [4]

1. Distribution Normal

La distribución de probabilidad normal o Gaussiana es simétrica y es la distribución más comúnmente utilizada en análisis estadísticos como lo es el caso en análisis de ingeniería. Para este trabajo varios de los parámetros de las propiedades del material son representados con esta distribución tales como: tenacidad a la fractura y resistencia del material.

Esta distribución tiene dos parámetros: Mean $[\mu]$ and standard deviation $[\sigma]$ y esta definida por la siguiente ecuación:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (1)$$

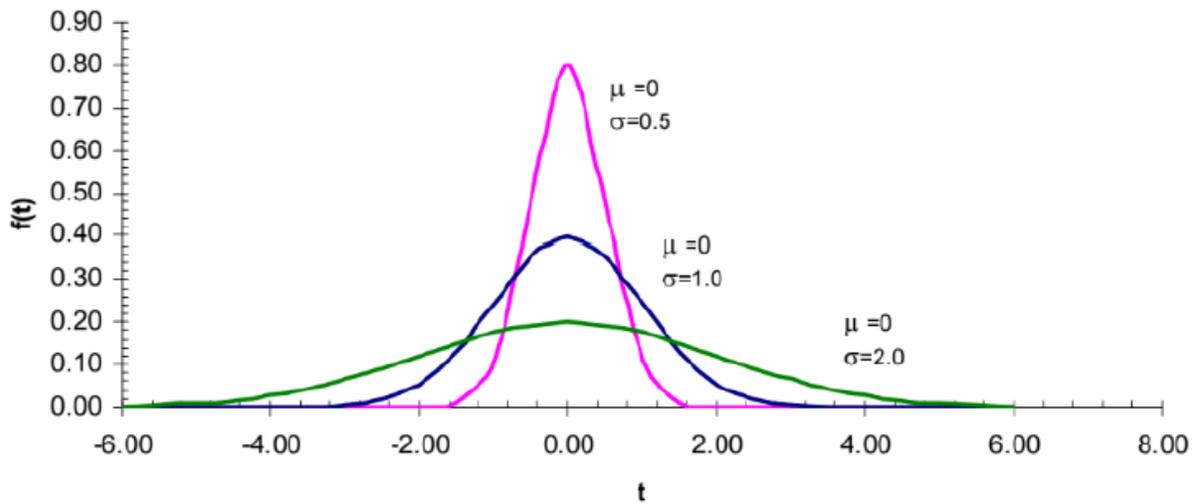


Fig.6. Distribución Normal [4]

La grafica anterior tiene 3 distribuciones normales con diferentes valores para su desviación estándar.

Con esta distribución se introducen dos términos: PDF (por sus siglas en inglés - Probability Density Function) describe la probabilidad relativa de que la variable analizada tome ciertos valores, esta está relacionada con la CDF (por sus siglas en inglés - Cumulative Density Function) donde la integral del área bajo la curva de PDF es el valor del CDF si se integra toda el área este valor sería uno que es una probabilidad de un 100 por ciento, del mismo modo la pendiente de la CDF es igual a PDF.

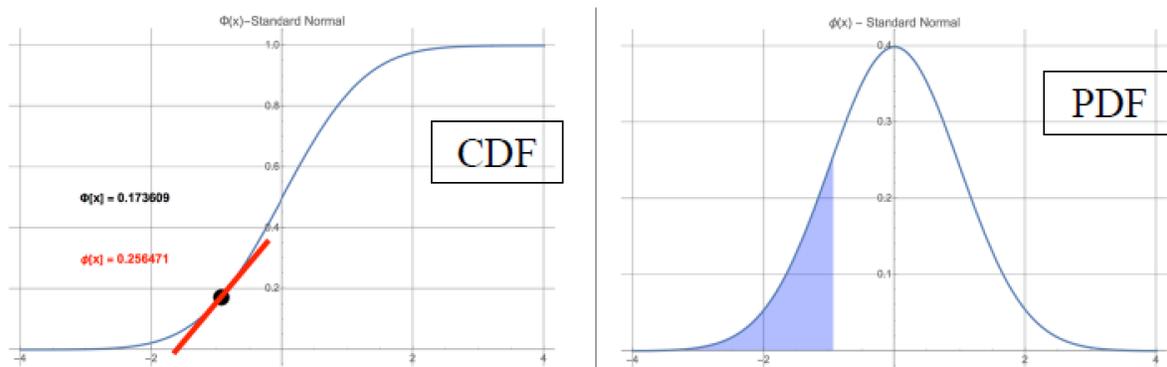


Fig.7. Comparación entre CDF y PDF [4]

2. Distribución Log-Normal

Esta distribución es bastante similar a la distribución Normal, de este modo tenemos que: Si el logaritmo natural de X es normal, se sabe que X es Log-Normal. También se tiene que esta distribución no es simétrica y no puede tomar valores negativos, una de sus mayores características es que la parte final de esta distribución o la cola tiende a ser más prolongada que en la distribución normal, esto se puede apreciar en la figura 8. Con esta distribución se representa el tamaño de grieta y esta esta descrita por la siguiente ecuación:

$$f(x) = \frac{1}{\sqrt{2\pi}(\xi x)} e^{-\frac{1}{2}\left(\frac{\ln(x) - \lambda}{\xi}\right)^2} \quad (2)$$

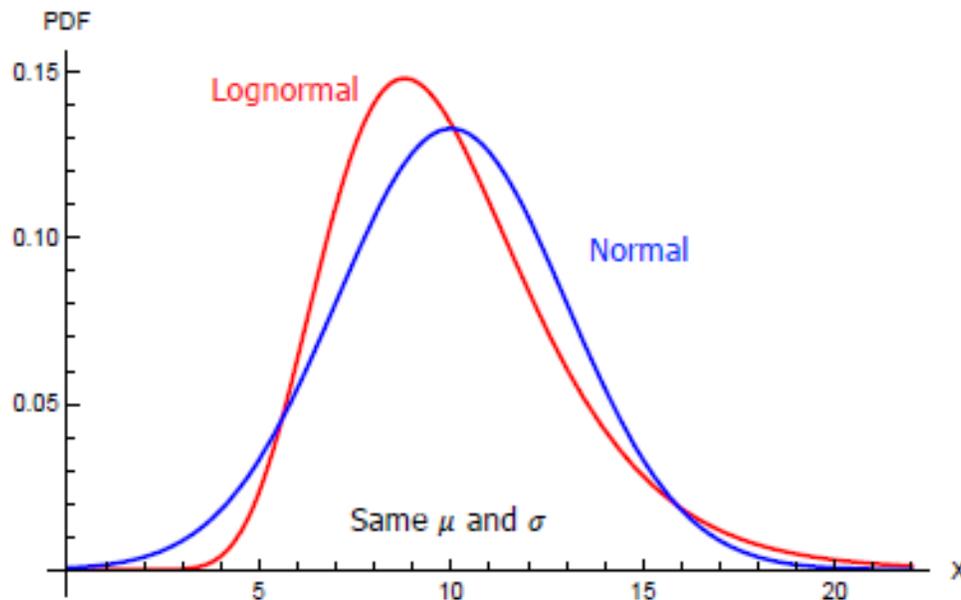


Fig.8. Comparación entre distribuciones Log-Normal y Normal [4]

Esta distribución al igual que la normal tiene dos parámetros: Mean (λ) and desviación estándar (ξ) los cuales tienen una relación con $[\mu]$ y con $[\sigma]$ de la siguiente forma:

$$\xi = \sqrt{1 + \left(\frac{\sigma}{\mu}\right)^2} ; \quad \lambda = \ln(x) - \frac{1}{2} \xi^2 \quad (3)$$

3. Distribución Weibull -Valor extremo

Esta es una distribución bastante flexible, con frecuencia se utiliza en este trabajo para modelar el tamaño inicial de la falla y el tiempo para la falla. Esta esta se define por dos parámetros, El de escala que es $[\beta]$ y el de forma que es $[\alpha]$ y esta descrita por la siguiente ecuación:

$$f(t) = \begin{cases} \frac{\beta}{\alpha} \left(\frac{t}{\alpha}\right)^{\beta-1} e^{-\left(\frac{t}{\alpha}\right)^\beta} & t > 0; \alpha, \beta > 0 \\ 0 & t \leq 0 \end{cases} \quad (4)$$

En el siguiente grafico se muestran algunos ejemplos de la distribución Weibull para diferentes valores de β y α .

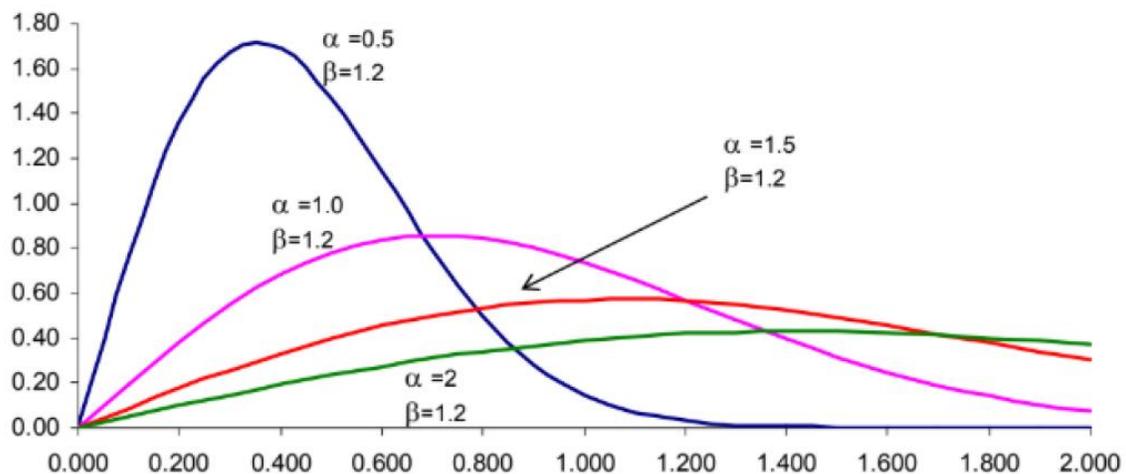


Fig.9. Distribución Weibull para diferentes valores de β y α [4]

4. Distribución Gumbel – Valor extremo

Esta distribución se usa para modelar cargas máximas como distribuciones de valor extremo. Usualmente usa el máximo o el mínimo valor de un conjunto de muestras de diferentes distribuciones. A modo de ejemplo, esta distribución puede ser usada para predecir cuándo va a ocurrir un terremoto y para este trabajo se usa para representar las cargas experimentadas por la aeronave. [4]

Esta distribución está representada por la ecuación (5) y (6). Las gráficas de estas ecuaciones se muestran en las figuras 10 y 11.

$$f(x) = \alpha^{-1} e^{-\frac{(x-\varepsilon)}{\alpha}} e^{-e^{-\frac{(x-\varepsilon)}{\alpha}}} \quad (5)$$

Where:

$\alpha =$ Parametro de escala

$\varepsilon =$ Parametro de locación

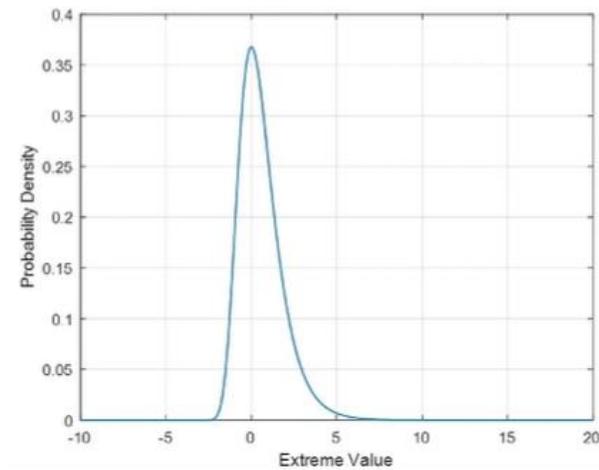


Fig. 10. Función de densidad de probabilidad para valor Extremo [17]

Función de distribución Acumulada para Valor Extremo (CDF):

$$f(x) = e^{-e^{-\frac{(x-\varepsilon)}{\alpha}}} \quad (6)$$

Donde:

$\alpha =$ Parametro de escala

$\varepsilon =$ Parametro de locación

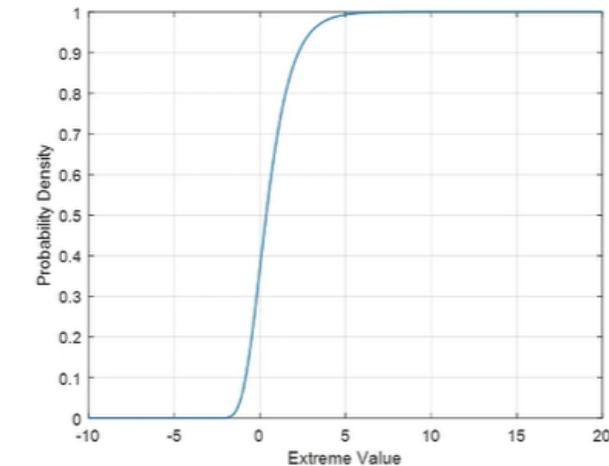


Fig. 11. Función de distribución Acumulada para Valor Extremo [17]

D. Selección de la distribución de probabilidad

Es importante asignar una distribución de probabilidad a los parámetros que se van a utilizar en este trabajo, de esta forma se puede modelar el caso analizado. Por ejemplo, las técnicas de inspecciones son representadas por la distribución que mejor se acomoda a ellas.

En cada uno de los parámetros utilizados se tiene una serie de datos estadísticos, que se utilizan para seleccionar su distribución. El método usado para asignar la distribución se describe a continuación.

1. Se organiza toda la información de los parámetros y se generan graficas con distribuciones empíricas.
2. A las gráficas se les varia la escala de los ejes X y Y.
3. Con las variaciones se busca la gráfica que se acerque más a una línea recta, lo cual indica que la distribución se acomoda bien a la serie de datos.

La siguiente grafica muestra un ejemplo de la selección para las distribuciones estadísticas mencionadas anteriormente.

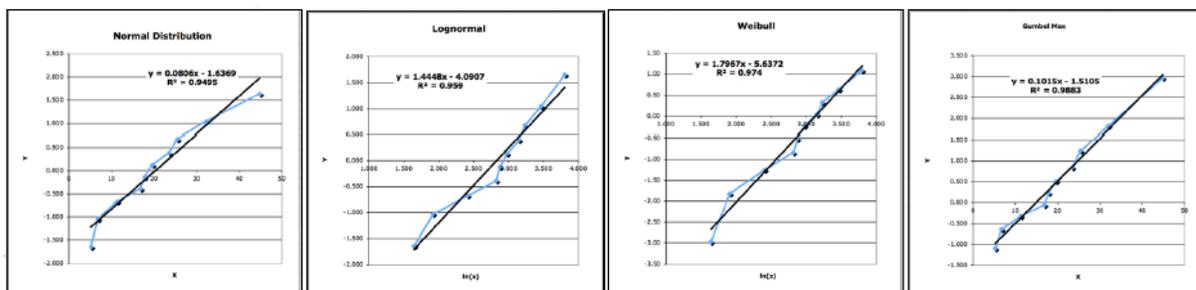


Fig.12. Selección de distribuciones para un conjunto de datos [15]

E. Análisis de tolerancia al daño usando probabilística - Probabilistic Damage Tolerance Fundamentals (PDTA)

El Análisis Probabilístico de Tolerancia al Daño (PDTA - Probabilistic Damage Tolerance Analysis) está enfocado en la probabilidad de falla de una aeronave en cualquier momento de su operación, considera las variaciones en: Tamaño inicial de la grieta, propiedades del material, geometría de la parte estructural analizada, tipo de uso, las cargas que experimenta la aeronave, la confiabilidad de la inspección, la probabilidad de detección y las reparaciones realizadas en la aeronave. PDTA se basa en la física de la mecánica de fractura. La figura 13 muestra un resumen de las consideraciones para aplicar el análisis probabilístico de tolerancia al daño.

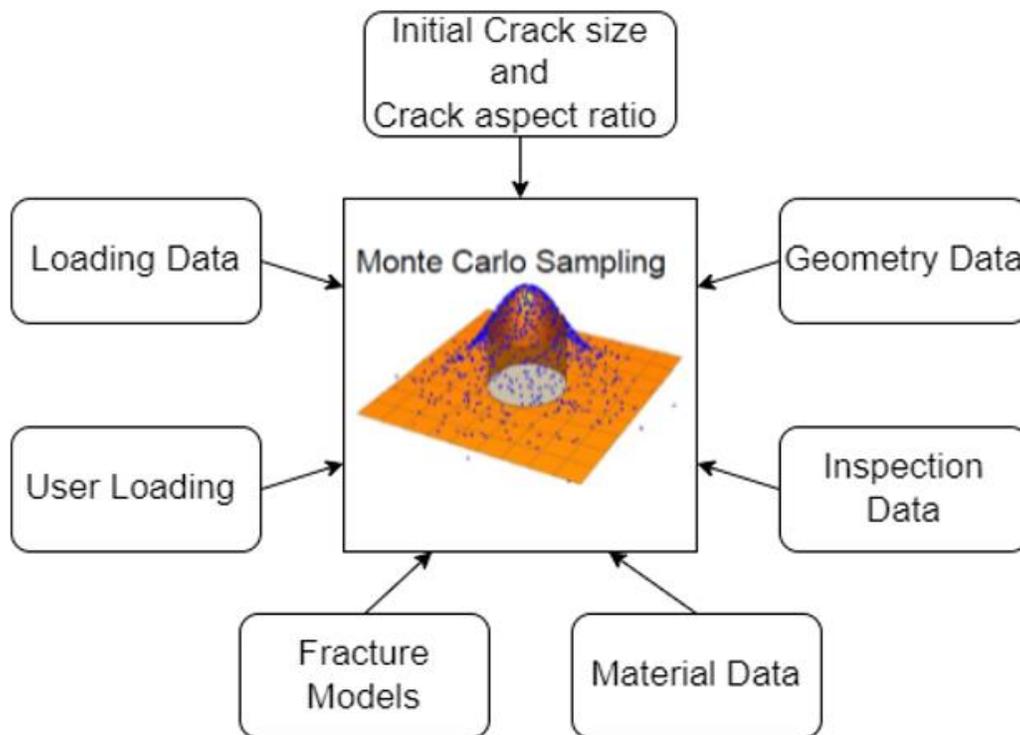


Fig.13. Esquema de PDTA [15]

Para poder trabajar con PDTA es necesario dejar claro algunos conceptos para hacer un análisis correcto de la situación.

1. Esfuerzo residual

Se puede definir como la capacidad de un cuerpo que posee un defecto (grieta) de soportar cargas o fuerzas mecánicas sin que se produzca una falla. El esfuerzo residual considera variaciones en: Tenacidad del material, tamaño de fractura y geometría analizada. Se basa en la mecánica de fractura y se sabe que a medida que pasa el tiempo (Flights) o los ciclos del sistema analizado avanzan, el esfuerzo residual va disminuyendo (Fig.14).

2. Probabilidad de falla – Probability of Failure (POF)

En el campo aeronáutico, la probabilidad de falla está definida como: La probabilidad de que durante un vuelo pueda ocurrir una falla estructural, esto es, el esfuerzo residual del componente del avión σ_{RS} será excedido debido al máximo valor del esfuerzo aplicado $\sigma_{MaxFlightStress}$. Ecuación (7)

$$POF = P(\sigma_{MaxFlightStress} > \sigma_{RS}) \quad (7)$$

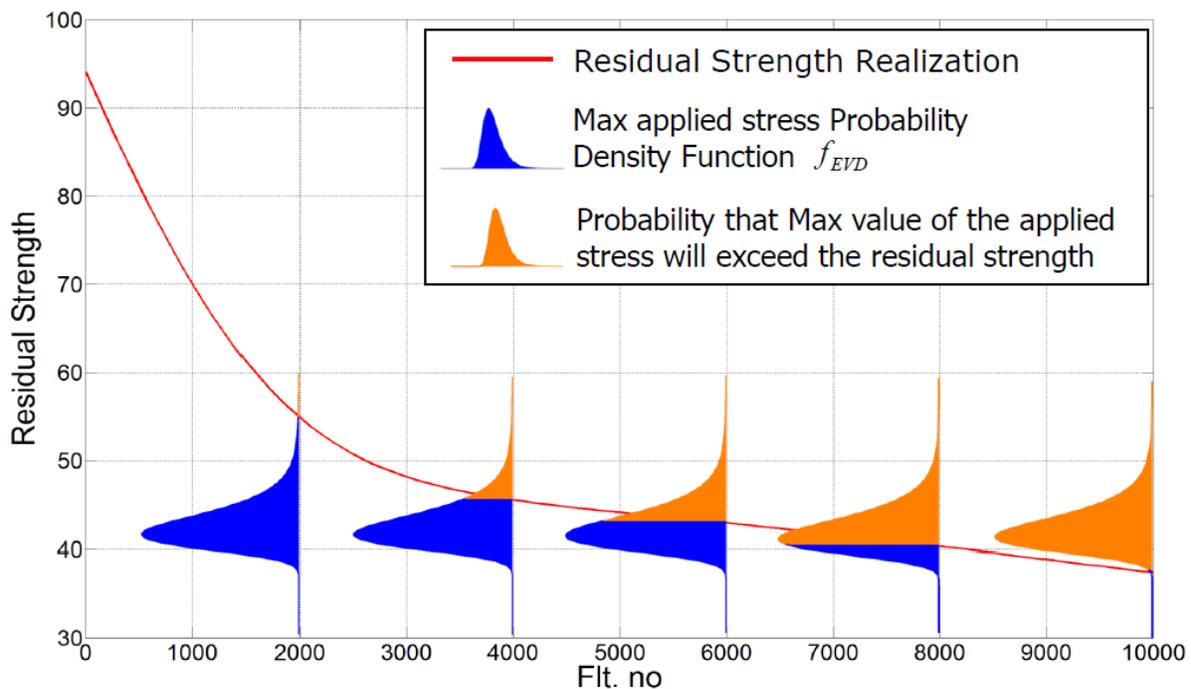


Fig.14. Curva de esfuerzo residual [15]

En la figura 14. se puede apreciar la relación que existe entre el esfuerzo residual y la probabilidad de falla de la aeronave. A medida que el avión acumula vuelos, aumenta la probabilidad de que el máximo valor de carga sobrepase el esfuerzo residual de la pieza de la aeronave, por ende, aumenta la probabilidad de que el esfuerzo máximo aplicado a la parte estructural sea mayor que el esfuerzo residual.

3. Muestreo de Monte Carlo

Se tiene que el muestreo de Monte Carlo es un método no determinístico, el cual posibilita la ejecución de experimentos con muestreo aleatorio para obtener la aproximación de expresiones matemáticas complejas, las cuales son difíciles de obtener normalmente [4]. Algo que tiene mucha importancia en este método es la capacidad de mejorar la aproximación de las expresiones matemáticas según la cantidad de muestreo que se haga, en otras palabras, se puede decir que su error absoluto decrece a medida que la cantidad de muestras aumenta, ecuación (8).

$$POF = \frac{1}{N} \sum POF_i \quad (8)$$

Donde N es el número de muestras y POF_i la probabilidad de falla de cada muestra.

Luego de presentar el marco teórico, cabe mencionar que el programa SMART-DT se basa en lo mencionado anteriormente, y con este programa se puede obtener gráficos como el siguiente mostrado en la figura 15.

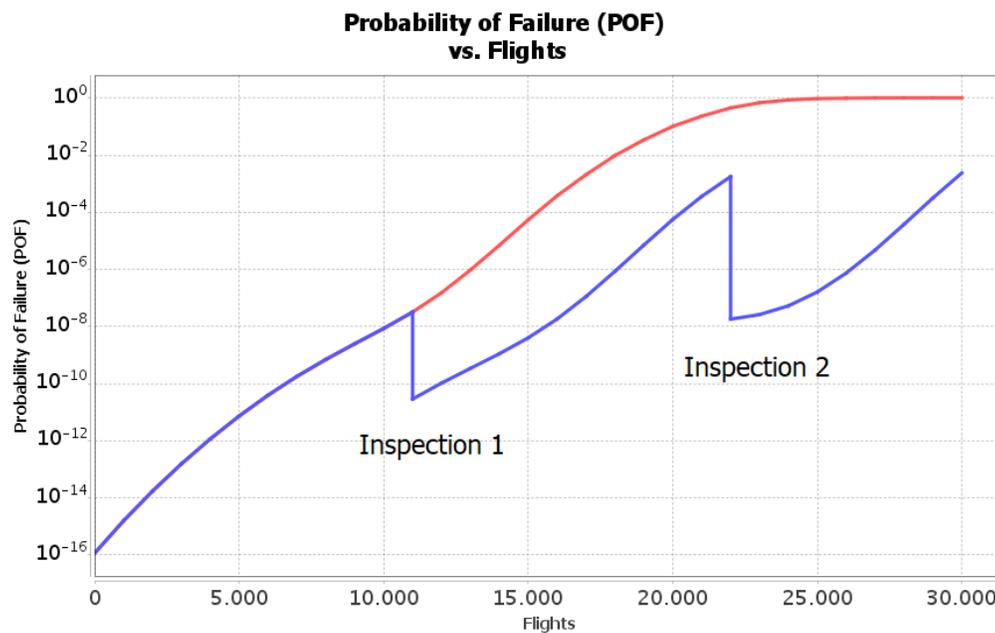


Fig.15. Comparación del POF con Inspecciones y sin Inspecciones [6]

La línea roja del gráfico, en la figura 15, representa la probabilidad de falla de la parte estructural analizada, teniendo en cuenta que hacen inspecciones, luego, la línea azul representa la probabilidad de falla una vez se adicionan inspecciones. Cada inspección y su posterior reparación generan una disminución de la probabilidad de falla de la pieza de la aeronave analizada.

VI. PLANTEAMIENTO DEL PROBLEMA

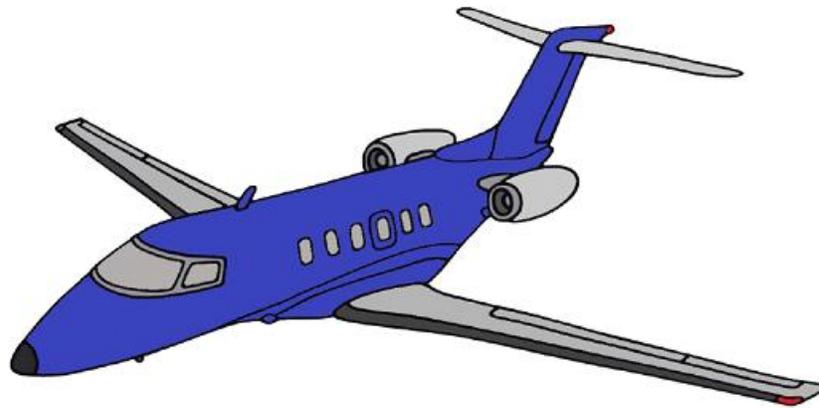


Fig.16. Jet Corporativo [15]

Para este trabajo de grado se realizó el análisis estadístico de un jet corporativo (Fig.16), particularmente el Wing Spar, esta pieza actúa como una viga que ayuda a distribuir las carga que experimentan las alas del avión durante el vuelo y en tierra, hacia la estructura del fuselaje del avión. Es de gran importancia analizar esta pieza ya que es la parte más crítica, con mayor presencia de fatiga y que tiene la mayor probabilidad de falla en la aeronave durante su vida útil, por lo que es necesario hacer un cronograma de inspecciones óptimo para esta pieza. En la figura 17, se puede analizar la geometría de la pieza, así como su simplificación para el análisis y la vista frontal de esta.

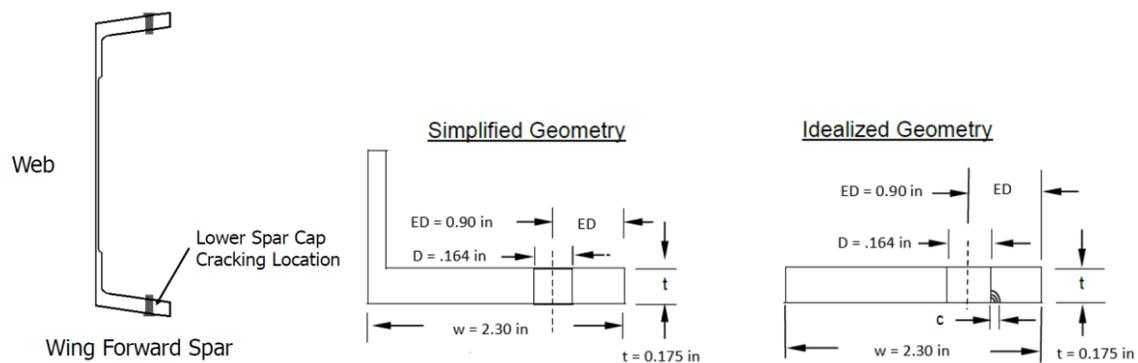


Fig.17.Wing Spar[15]

Se presenta un código en Matlab el cual ejecuta el programa SMART-DT [6]. este programa es de gran importancia para el análisis estadístico ya que utiliza mecánica de fractura de materiales y análisis de tolerancia al daño, es capaz de obtener las distribuciones de probabilidad de falla de un componente aeronáutico [6]. La interfaz de usuario del programa se muestra en la figura 18, donde se puede apreciar que este considera: cargas, propiedades del material, geometría de la pieza y tiempos de inspecciones, entre otros.

Se toma como base un código en MATLAB que es capaz de simular y proveer la probabilidad de falla de una aeronave teniendo en cuenta el calendario de inspecciones que se le programe o incluso sin este, esta programa de inspecciones se hace con el objetivo de mantener el riesgo de falla de la aeronave por debajo de 10^{-7} y es indispensable para la industria elegir la combinación más barata posible, proceso que realiza el código en MATLAB, sin embargo, este código es muy costoso computacionalmente ya que toma mucho tiempo su ejecución para casos aproximados a la realidad (más de 30 horas) y carece de varias consideraciones para hacer una simulación de un caso real.

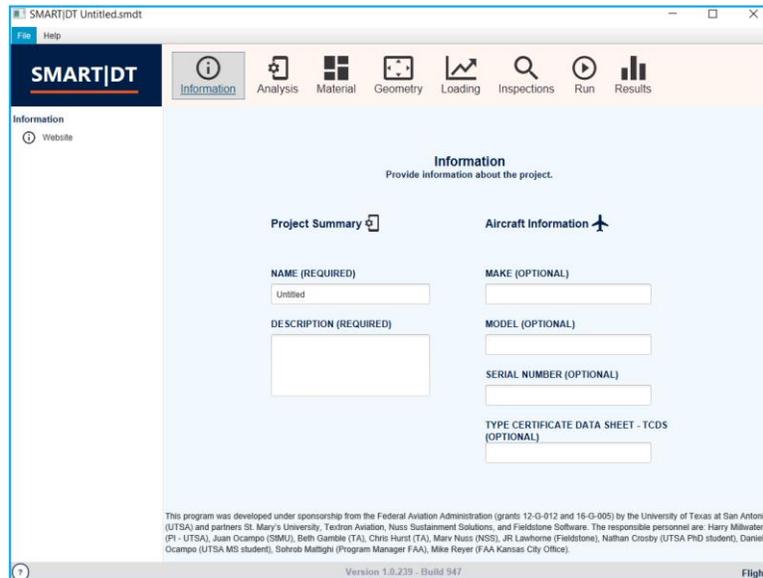


Fig.18. Graphical User Interface (GUI) SMART-DT.

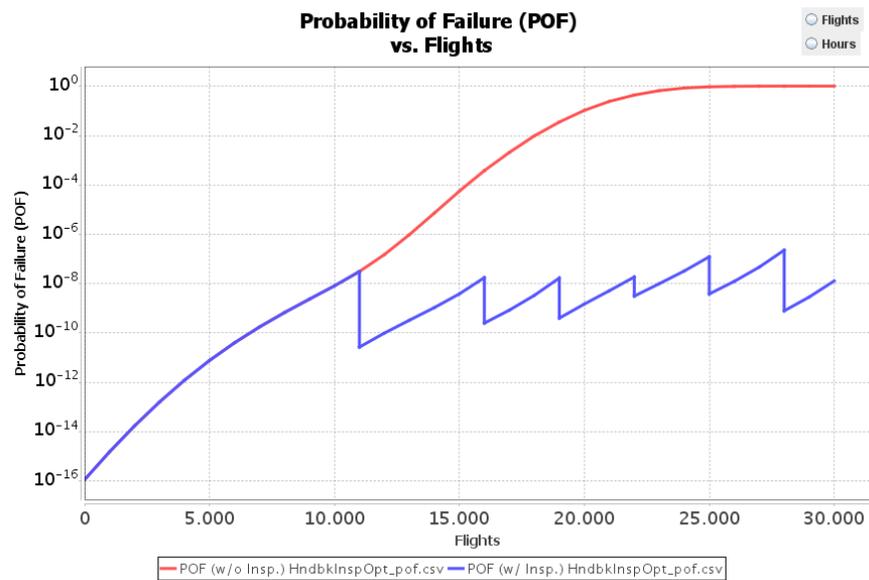


Fig.19. Grafico generado en SMART-DT, Flights vs Probability of failure [6]

El código base de optimización de inspecciones genera un arreglo con todas las posibles combinaciones de inspecciones que se pueden realizar en un avión dadas ciertas técnicas de inspecciones (tales como: corriente Eddy, visual, líquidos penetrantes, etc) y posibles tiempos para programarlas. Desde MATLAB se ejecuta el programa SMART-DT junto con un archivo de entrada (archivo tipo dat.) con los datos estadísticos de la pieza estructural que se está analizando tales como: tamaño inicial de la grieta, tamaño de la grieta reparada y tenacidad a la fractura, también se dan los posibles tiempos de inspección, los costos por cada técnica de inspección que se aplica y las distribuciones de probabilidad para estas técnicas. SMART-DT se ejecuta tantas veces como combinaciones de inspecciones se tengan, de este modo se analiza cada una de las combinaciones, desechando las inspecciones inviables, por que superan el nivel de riesgo permitido y las inspecciones viables, que mantienen el nivel de riesgo por debajo del umbral, las almacena en una matriz en la cual luego serian analizados los costos de cada inspección para elegir la más barata, se muestra en la figura 19. el resultado de una ejecución de SMART-DT.

Se decidió afrontar este problema y optimizar el tiempo de ejecución de este código porque es de gran importancia mantener la viabilidad de la industria aeronáutica a través de ahorrar dinero a las compañías y más importante aún, por que mantiene la seguridad de vuelo en los aviones y a su vez la seguridad de las personas que viajan en este, garantizando así la confiabilidad en cada vuelo.

Como dato de gran relevancia, en este trabajo se buscó asesoría de expertos en el tema de mantenimiento aeronáutico, de esta forma se tuvo en cuenta los datos reales y al investigar información de mantenimiento de las aerolíneas se generó un algoritmo que permite obtener una combinación optimizada, que tiene en cuenta factores reales de la industria y que a su vez disminuyó el costo computacional del código para poder hacer un uso más eficiente de este.

VII. METODOLOGIA

El desarrollo metodológico de este trabajo se realizó en dos partes: La primera parte (parte A) donde se analizó el código base existente y la segunda parte que consistió en la optimización del tiempo de ejecución del código base (parte B). A continuación, se detalla el proceso de cada parte desarrollada.

A. Análisis del Código base en MATLAB

En esta parte, se hizo un análisis riguroso del código base en MATLAB el cual tenía varios algoritmos con los que se encontraba una combinación de inspecciones optimizada. Este análisis se hizo con el objetivo de entender el funcionamiento del código, y poder modificarlo, logrando de esta forma reducir su tiempo de ejecución, también se buscó tener en cuenta criterios de mantenimiento reales y que se aplican actualmente en la industria aeronáutica en torno al plan de mantenimiento realizado en las aeronaves, con el objetivo de generar un algoritmo apropiado para el código.

En su estado más básico el cual sigue el flujo mostrado en la figura 20, el código se inicializaba en MATLAB, luego de esto ejecutaba el programa SMART-DT con el archivo de entrada, el código encontraba el punto donde la POF sobrepasaba el nivel de riesgo, luego de esto, programaba una inspección en el posible tiempo de inspección anterior a ese punto para posteriormente ejecutar SMART-DT de nuevo, tal como se muestra en la figura 21, y así seguía sucesivamente evaluando el punto donde la distribución sobrepasaba el nivel de riesgo hasta tener una combinación de inspecciones que mantuviera la POF de avión por debajo del umbral de riesgo.

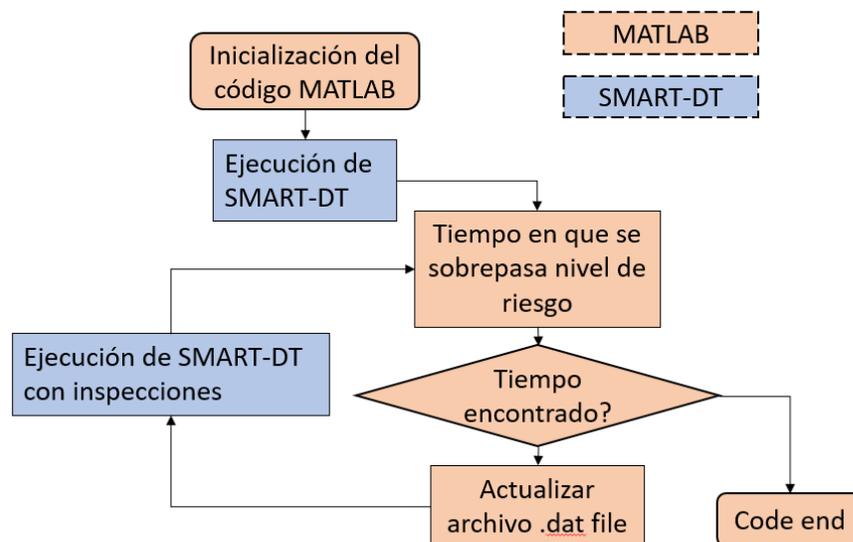


Fig.20. Diagrama de Flujo para el Primer Estado Del Código en Matlab [16]

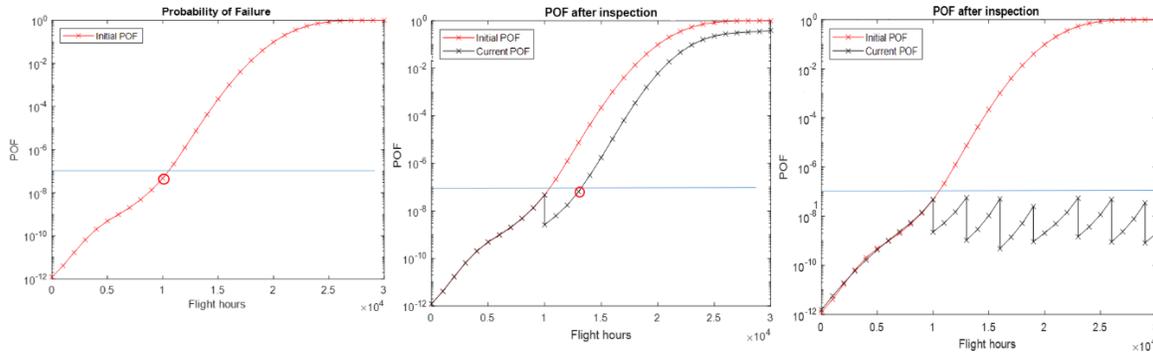


Fig.21. Obteniendo Programa de Inspecciones, Primer Estado del Código [16]

Se implementaron varios algoritmos que se explican a continuación, con el objetivo de mejorar el análisis y hacer una selección optimizada de las inspecciones de la aeronave analizada, teniendo en cuenta todas las posibles opciones de combinación.

En primer lugar, se generó un algoritmo que generaba un arreglo con todas las posibles inspecciones que se podían realizar en la aeronave analizada. Este arreglo se estructuraba con una base binaria, si se trataba de un solo tipo de inspección, donde se define las inspecciones de la siguiente forma: inspección=1; no inspección=0 (Tabla 1); si se tenían dos tipos de inspecciones se estructuraba con una base terciaria donde primer tipo de inspección =1; segundo tipo de inspección=2; no inspección = 0 como se puede ver en la tabla 2 y así sucesivamente se podían adicionar más técnicas de inspecciones.

	Schedule times (flight hours)			
	(2000)	(4000)	(6000)	(8000)
(1)	0	0	0	0
(2)	0	0	0	1
(3)	0	0	1	0
(4)	0	0	1	1
(5)	0	1	0	0
(6)	0	1	0	1
(7)	0	1	1	0
(8)	0	1	1	1
	⋮	⋮	⋮	⋮
(16)	1	1	1	1

Tabla 1. Numero de Combinaciones de Inspecciones Posibles [16]

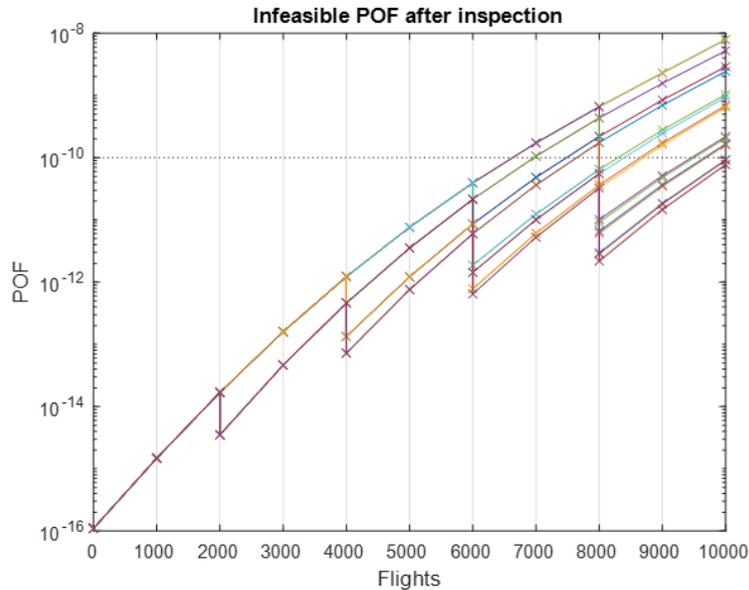


Fig.22. Combinaciones de inspecciones con POF inviables de la tabla 1[16]

Como se muestra en la tabla 1, y a modo de ejemplo, cuando se tiene que los posibles tiempos para implementar una inspección son: 2000, 4000, 6000, 8000, y que solo se tiene una técnica de inspección a evaluar en cada uno de esos posibles tiempos, se genera un arreglo con 4 columnas y 16 filas, el cual contiene todas las posibles combinaciones de inspecciones (16) que se pueden programar en este ejemplo para la aeronave, incluyendo las que sobrepasan el nivel de riesgo.

Luego si se quiere aumentar a dos técnicas de inspección para el mismo ejemplo anterior se tendría que generar 4 columnas ya que los posibles tiempos de inspección no cambian, pero en lugar de tener solo 16 combinaciones o filas, se tienen 81 combinaciones o filas mostradas en la tabla 2, lo cual es un aumento bastante significativo. Por este motivo, al realizar un ejemplo real donde los posibles tiempos de inspecciones a evaluar son más de 30 y se tienen más de 3 técnicas de inspección, inevitablemente se generan enormes arreglos que se traducen en capacidad computacional para poder evaluar todos los datos. A continuación, en la ecuación (9) se expresa matemáticamente el número de combinaciones dados los posibles tiempos de inspección y las técnicas de inspección que se van a evaluar.

$$N^{\circ} \text{ of inspection techniques}^{\text{Possible inspection times}} = \text{Combination of inspections} \quad (9)$$

Example 1:

$$2^4 = 16 \leftarrow \text{Combination of inspections to evaluate}$$

Example 2:

$$3^4 = 81 \leftarrow \text{Combination of inspections to evaluate}$$

Schedule times (flight hours)					
	(2000)	(4000)	(6000)	(8000)	
(1)	0	0	0	0	Schedule combinations
(2)	0	0	0	1	
(3)	0	0	0	2	
(4)	0	0	1	0	
(5)	0	0	1	1	
(6)	0	0	1	2	
(7)	0	0	2	0	
(8)	0	0	2	1	
	⋮	⋮	⋮	⋮	
(81)	2	2	2	2	

Number of possible inspections times: number of positions that will be fill with all the numerical combinations in base 3

Tabla 2. Número de Combinaciones de Inspecciones Posibles [16]

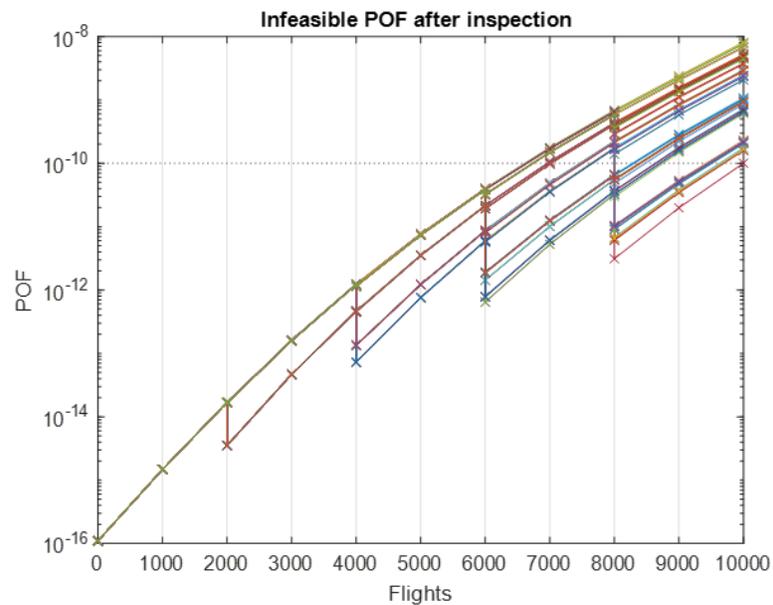


Fig.23. Combinaciones de inspecciones con POF inviables de la tabla 2 [16]

Es importante notar que hasta este punto se están evaluando todas las combinaciones, incluso algunas que no es necesario evaluar, porque se sabe que van a ser inviables, por ejemplo, como se muestra en la gráfica de la figura 24, hay combinaciones en donde se están evaluando inspecciones incluso por encima del límite de riesgo, lo cual no es necesario y hace que el código ejecute más veces SMART-DT. Por eso, otro algoritmo que se implementó es el algoritmo de salto, este algoritmo encuentra el tiempo donde la probabilidad de falla sobrepasa el límite de riesgo, luego programa una inspección en el tiempo inmediatamente anterior, antes que la POF sobrepase el límite, tal como se ve en la figura 25. También hay otro algoritmo el cual se encarga de evitar que el código evalúe más inspecciones en una combinación viable, esto solo genera más costos sabiendo que la combinación ya mantiene la POF por debajo del nivel de riesgo. Estos algoritmos se desarrollaron con el fin de saltar esas combinaciones que evalúan inspecciones por encima del riesgo permitido y que agregan más inspecciones a una combinación viable, dando como resultado un código que ejecutaba muchas menos veces SMART-DT.

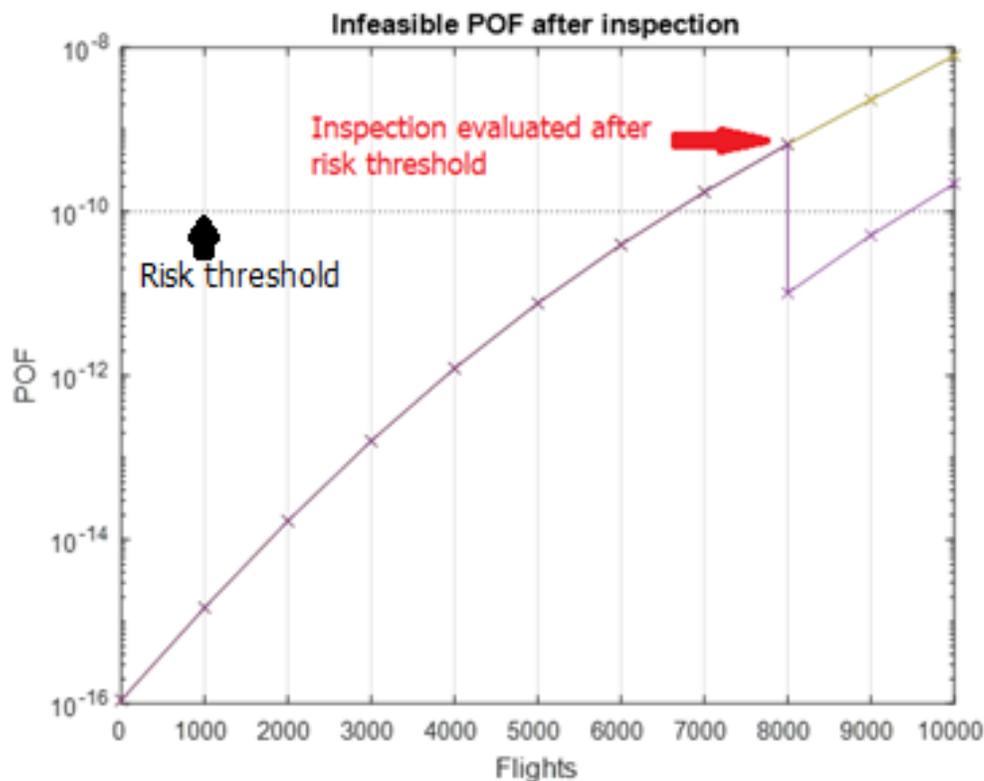


Fig.24. Grafica antes de aplicar el algoritmo de salto [16]

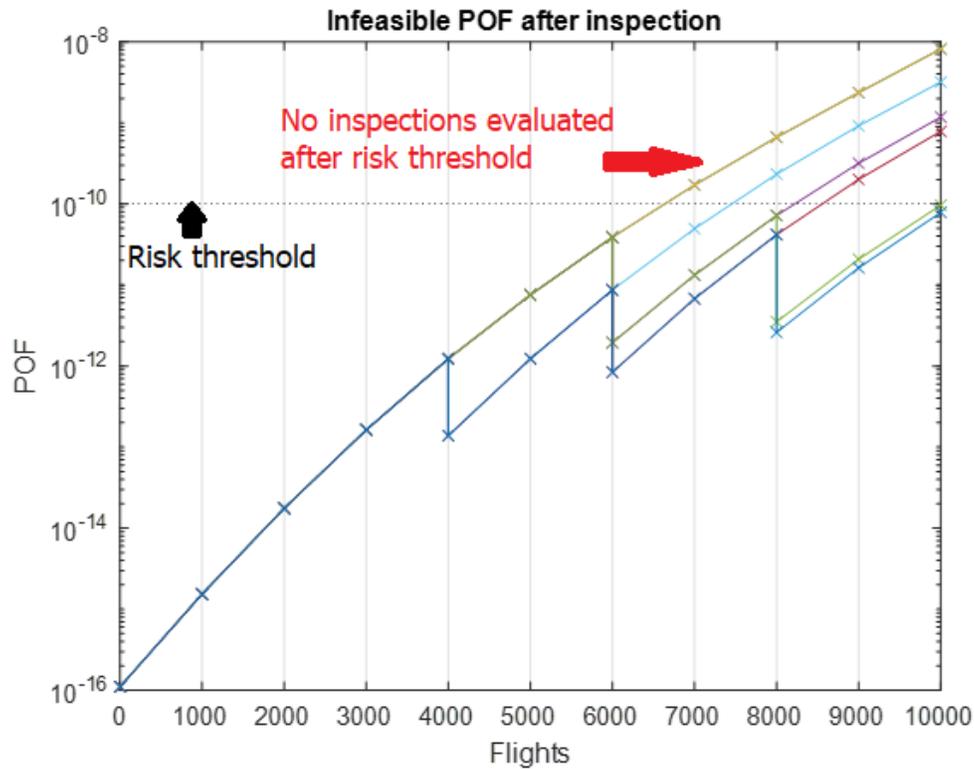


Fig.25. Grafica después de aplicar el algoritmo de salto [16]

La estructura del código de MATLAB que se muestra en la figura 26, usa todos los algoritmos mencionados anteriormente, básicamente se genera un arreglo con todas las posibles combinaciones de inspecciones que se pueden hacer en la aeronave analizada, teniendo en cuenta parámetros de entrada tales como: número de técnicas de inspección y los posibles tiempos en los que se van a evaluar las inspecciones. Posteriormente se evalúa una combinación, con los datos que se proporcionaron, se genera el archivo .dat el cual va a ser la entrada de datos al programa SMART-DT, luego se ejecuta SMART-DT junto con el archivo .dat, como resultado de esto se obtiene la POF que tendría la pieza de la aeronave durante su vida útil, teniendo en cuenta las inspecciones de la respectiva combinación evaluada, luego si el POF está por encima del umbral de riesgo la combinación evaluada se descarta, por el contrario la combinación evaluada está por debajo del nivel de riesgo, esta se guarda en un arreglo con todas las combinaciones viables bajo las condiciones dadas, se repiten estos pasos hasta que se hallan evaluado todas las combinaciones de inspecciones, al final del proceso, el arreglo con todas las combinaciones viables se evalúa para elegir la más barata.

A pesar de que este conjunto de algoritmos daba una combinación optimizada, no era viable su ejecución para casos reales pues este seguía consumiendo una gran cantidad de recursos computacionales, debido a que ejecutar el programa SMART-DT lleva cierta cantidad de tiempo, y al aumentar la cantidad de combinaciones de inspecciones se

aumentaba la cantidad de combinaciones y de veces que SMAT-DT se ejecutaba, se tardaba hasta más de 30 horas entregando un resultado. Este tipo de combinaciones requieren de computadores con buenas prestaciones ya que implican un gran costo computacional, por tal razón se buscó optimizar el código para que su uso en computadores estándar sea viable.

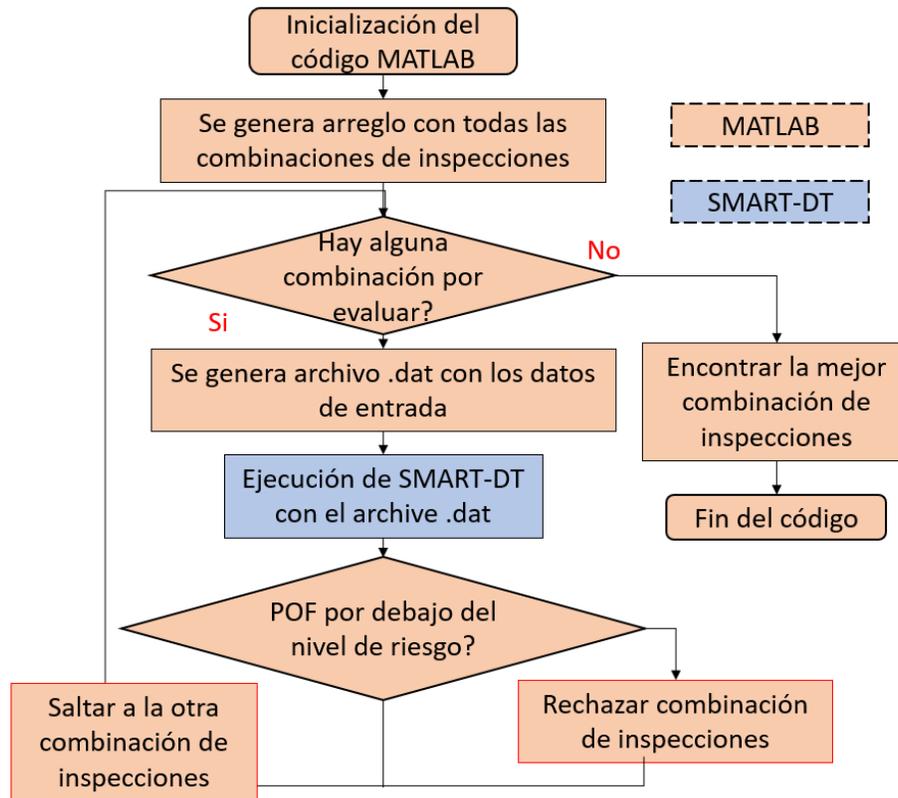


Fig.26. Diagrama de flujo del código base en MATLAB [16]

El código base de optimización de inspecciones tiene una estructura principal desde donde se llaman tres funciones: Skipping algorithm function, Reading POF function e input data function.

En la estructura principal es donde se escribe el archivo .dat mostrado en la figura 27, que contiene los datos de entrada del programa SMART-DT, estos datos son: los datos de la aeronave a analizar, los datos estadísticos del método ya sea que se use Montecarlo o Adaptive importance sampling, también se definen los datos de fractura mecánica del material tales como: crecimiento de grieta, tamaño inicial de la grieta, relación de aspecto de la grieta, tenacidad a la fractura y límite elástico del material. Por último, se definen las características estadísticas de cada una de las técnicas de inspección implementadas, así como su ID, también los tiempos donde se programan.

```

!-----!
! AIRCRAFT INFORMATION                               ! INSPECTIONS
!-----!
TITLE = HndbkInspOpt                                ! INSPECTIONS = 2000 4000 6000
AC_MAKE = Acme                                       ! INSPECTION_TYPE = 1 1 2
AC_MODEL = SkyRunner                                  ! INSPECTION_ID = 1
AC_SERIAL_NUM = A11                                  ! PROB_OF_INSPECTION = DETERMINISTIC 1.0
AC_TCDS =                                             ! POD = LOGNORMAL 1.05964 3.66907
!-----!
! METHOD                                               ! REPAIR_CRACK_SIZE = PERFECT
!-----!
INTEGRATION_METHOD = AIS 1161578612                 ! INSPECTION_ID = 2
AIS_TARGET_COV = 0.1                                ! PROB_OF_INSPECTION = 1.0
AIS_NSAMPLES = 100                                  ! POD = LOGNORMAL 0.0180 0.0109
AIS_MAXITER = 100                                    ! REPAIR_CRACK_SIZE = PERFECT
POF_MAX_INC = 22000 1000                            ! INSPECTION_ID = 3
ANALYSIS_TIME_UNITS = FLIGHTS                       ! PROB_OF_INSPECTION = 1.0
!-----!
! FRACTURE MECHANICS                                 ! POD = LOGNORMAL 0.1413 0.0302
!-----!
CRACK_GROWTH_CODE = MASTERC_USER RiskAssessmentHandbook.avsn ! REPAIR_CRACK_SIZE = PERFECT
INITIAL_CRACK_SIZE = LOGNORMAL 0.00248 0.00129
CRACK_ASPECT_RATIO = DETERMINISTIC 1.0
FRACTURE_TOUGHNESS = NORMAL 26.0 2.0
YIELD_STRENGTH = DETERMINISTIC 80.0
!-----!
! FRACTURE MECHANICS                                 ! LOADING AND EVD PARAMETERS
!-----!
CRACK_GROWTH_CODE = MASTERC_USER RiskAssessmentHandbook.avsn ! EVD_TYPE = USER 14.5 0.8 0.0
INITIAL_CRACK_SIZE = LOGNORMAL 0.00248 0.00129
CRACK_ASPECT_RATIO = DETERMINISTIC 1.0
FRACTURE_TOUGHNESS = NORMAL 26.0 2.0
YIELD_STRENGTH = DETERMINISTIC 80.0
!-----!
! FRACTURE MECHANICS                                 ! DESCRIPTION
!-----!
! Starting inputs for nspction optimization problem based on the USAF Risk Assessment
! Handbook example problem.

```

Fig 27. Archivo. Dat

Por otro lado, el archivo .dat y el programa SMART-DT se ejecutan desde MATLAB en esta misma estructura principal.

La primera función que se llamará es " Input data function " que contiene los datos de entrada definidos por el usuario: nivel de riesgo, cronogramas para posibles inspecciones, costos de cada técnica de inspección definida. Luego de la ejecución de SMART-DT con el archivo .dat, la segunda función que se utiliza es "la función de lectura". Esta función permite que el código lea los datos de salida generados por SMART-DT. Finalmente, se llama a la "función de salto", esta función encuentra las combinaciones factibles y no factibles almacenando las factibles en un arreglo, también salta alguna combinación que no necesita ser evaluada.

B. Aporte al código de optimización de inspecciones en MATLAB – Optimización del tiempo de ejecución

Una vez comprendido los algoritmos utilizados en el código base y realizado los procesos indicados en la parte A, se procedió a organizar algunos detalles, tales como:

- En el código se escribieron varias líneas con el fin de dar tratamiento a los datos de un archivo .csv, en este se almacenan los datos de salida del programa SMART-DT, para la primera iteración del programa, el código debe leer las columnas del archivo .csv que corresponden a los vuelos y la probabilidad de falla de la aeronave sin inspecciones, columnas 1 y 3 respectivamente de la tabla 3.
- Después de esta primera iteración se continúa leyendo los vuelos de la aeronave, pero esta vez con la probabilidad de falla al aplicar las inspecciones de la combinación analizada, columnas 1 y 3 respectivamente de la tabla 3.
- Adicionalmente se definieron varios contadores para encontrar la cantidad de veces que se ejecutaba el programa SMART-DT.

- Como parámetro externo se calculó el tiempo de ejecución del código en su fase inicial, teniendo en cuenta que este varía según la computadora en la cual se ejecute el programa, por tal razón, este análisis se llevó a cabo en la misma computadora para tener los valores de referencia fijos.

		Results without Inspections		Results with Inspections	
Flight Number	Flight Hours	SFPOF	Cumulative	SFPOF	Cumulative
0	0	4.56E-13	4.56E-13	4.56E-13	4.56E-13
1000	1650	4.61E-12	1.80E-09	4.61E-12	1.80E-09
2000	3300	4.38E-11	1.92E-08	4.38E-11	1.92E-08
2000	3300	4.38E-11	1.92E-08	4.38E-11	1.92E-08 <--Insp
3000	4950	3.03E-10	1.53E-07	3.03E-10	1.53E-07
...

Time First iteration All other iterations

Tabla 3. Datos de salida SMART-DT

Entender el funcionamiento del programa SMART-DT fue muy importante, ya que su correcta manipulación permitiría obtener simulaciones de casos reales y analizar los datos de salida del código con una buena fundamentación.

Con el fin de mejorar el código de optimización de inspecciones, se realizó una búsqueda de los factores usualmente empleados en la industria del mantenimiento aeronáutico para incorporarlos al código. En primer lugar, se buscó aplicar algunos de los tipos de inspecciones comúnmente usados en la industria aeronáutica, como es el caso de la inspección visual, Corriente de Eddy, entre otras, también se asignaron los costos asociados a cada una de las técnicas de inspección. De esta búsqueda se encontró que la inspección visual es la técnica que las aerolíneas emplean con mayor frecuencia para generar el plan de mantenimiento, debido a que es la más económica e incluso el piloto la puede realizar, otras técnicas de inspección más avanzadas se usaban con menor frecuencia por sus costos elevados y la complejidad de su ejecución. Adicionalmente para realimentar el código, se llevaron a cabo una serie de reuniones con expertos en mantenimiento aeronáutico con el objetivo de discutir y definir las restricciones para el desarrollo del trabajo.

Finalmente, se procedió a generar un algoritmo capaz de programar cada una de las técnicas de inspección en los diferentes posibles tiempos de inspección. Con esto se buscaba reducir en gran medida la cantidad de veces que se ejecutaba SMART-DT desde MATLAB y consecuentemente el tiempo total de ejecución del código de optimización de inspecciones logrando así una optimización de este mismo.

VIII. RESULTADOS Y DISCUSIÓN

Para validar el nuevo algoritmo implementado en el código, se realizó un ejemplo con el que fin de obtener el mismo resultado antes que el código fuese optimizado

Se tomó un ejemplo del manual para diseño de tolerancia al daño - Handbook for Damage Tolerant Design) debido a sus facilidades para comparar el nuevo código optimizado con resultados previos obtenidos con el código antiguo sin optimizar [1] [16]. Como se explicó en la sección de la metodología, tanto como para el código sin optimizar y el código optimizado se deben suministrar las variables de entrada que el código escribe en el archivo .dat. También proporcionar los datos estadísticos para el análisis de fractura mecánica mostrados en la tabla 4, de la parte estructural que será evaluada y los datos estadísticos para las inspecciones que se utilizan (Tabla 5). Por último, se deben indicar los posibles tiempos donde se puede programar una inspección. En la tabla 6 se encuentran los posibles tiempos de inspección empleados. El umbral de falla fue de 10^{-7} en este caso. Antes de optimizar el código, las tres técnicas de inspección eran evaluadas en cada uno de los posibles tiempos de inspección, lo que significaba que SMART-DT tuviera que ejecutarse muchas más veces que el código optimizado

Tabla 4. Datos de entrada para el análisis de fractura mecánica

Variable	Dist. Type	Mean	St.Dev.	Notes
Initial Crack Size	Lognormal	0.00248 in	0.00129	Reamed fastener hole
Repair Crack Size	Lognormal	0.00248 in	0.0129	Assuming Repair is Replacement of part
Fracture Toughness	Normal	26.0 ksi	2.0	7050- T651 Plate
EVD	Gumbel	14.5 ksi	0.8	

Tabla 5. Datos de entrada tipo de inspecciones – Código sin optimizar

Insp ID	Inspection Tpe	Material	Crack Type	Dist. Type	Mean [in]	St. Dev. [in]	Source	Cost
POD 1 	Automate bolt hole eddy current	Aluminum	T	Lognormal	0.018	0.0109	Aeronautical Applications of Non-destructive	5x
POD 2 	Eddy current sliding probe	Aluminum	Overall	Lognormal	0.0788 +0.0625	0.0302	NDE Capabilities Book	1x
POD 3 	Visual	Aluminum		Lognormal	0.99714 +0.0625	3.66907	NDE Capabilities Book	0.1x

Tabla 6. Posibles tiempos de inspección– Código sin optimizar

Possible inspection times									
2,000	4,000	6,000	8,000	10,000	12,000	14,000	16,000	18,000	20,000

Para las inspecciones visuales y Eddy current sliding probe, la media de su distribución tiene un factor (+0.0625), este factor que se suma a la media se debe al remache que dificulta detectar una grieta, por lo que la media de la distribución se vuelve un poco más grande.

Para el código optimizado, se utilizan los mismos datos de entrada presentados en las tablas 4 y 5. Los datos de entrada son iguales, la diferencia es que a través del algoritmo que permite seleccionar que inspección se pueden elegir las inspecciones adecuadas para cada tiempo. De esta manera, en todos los posibles tiempos de inspección se evalúa la inspección visual, en los tiempos 6k, 12k y 18k se evalúan la inspección visual y Automate bolt hole Eddy current y en el tiempo 12k se evalúan los tres tipos de inspección presentados para este ejemplo, visual, Eddy current sliding probe y Automate bolt hole Eddy current como se indica en la tabla 7.

Tabla 7. Datos de entrada Handbook, nivel de riesgo posibles tiempos de inspecciones – Código optimizado

Possible inspection times									
2,000	4,000	6,000	8,000	10,000	12,000	14,000	16,000	18,000	20,000
		  			  			  	  

Los resultados obtenidos son mostrados en las figuras 28 y 29. A la izquierda se pueden ver las gráficas sin optimizar y a la derecha una vez implementados los algoritmos de optimización. La figura 28 presenta el número de veces que se ejecutó SMART-DT para este ejemplo específico, tanto en el código optimizado como no optimizado, así como el tiempo que necesitó para evaluar todas las combinaciones necesarias para llegar a una solución.

Los resultados obtenidos con el código optimizado y no optimizado fueron los mismos, ambos presentan dos combinaciones de inspecciones, que son las más económicas posibles y mantienen el nivel de riesgo de falla de la aeronave por debajo de 10^{-7} .

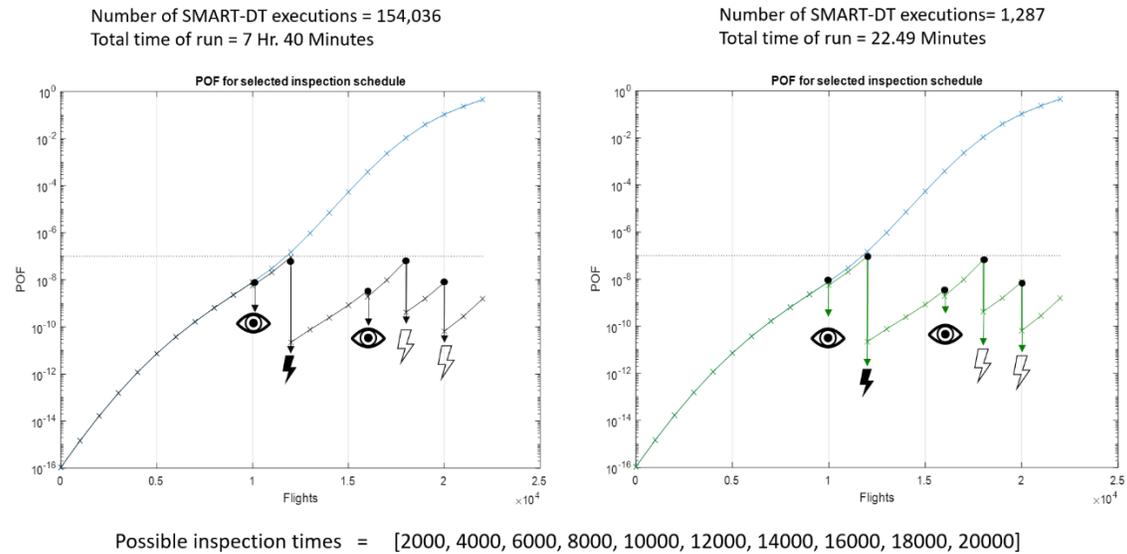


Fig.28. Primera combinación de inspecciones encontrada

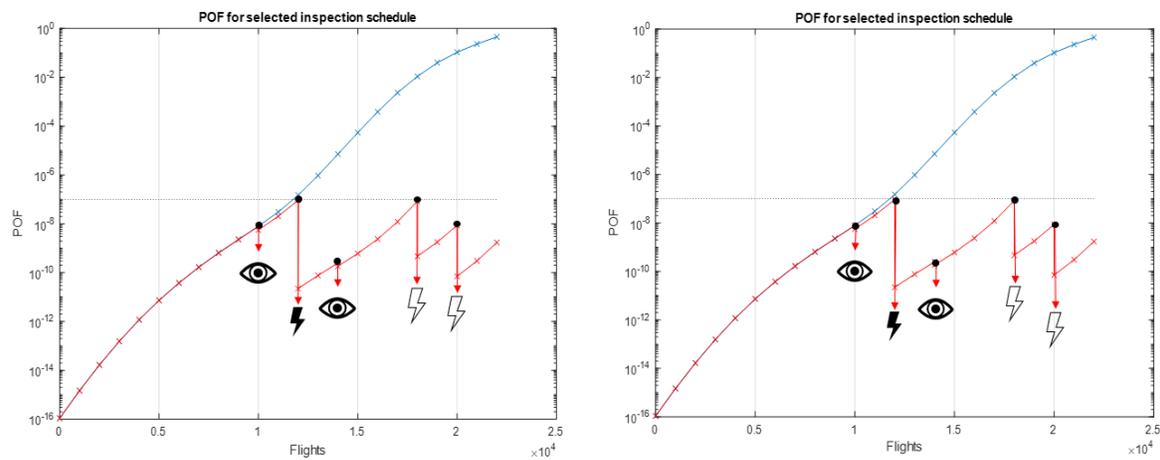


Fig.29. Segunda combinación de inspecciones encontrada

Es importante señalar que los resultados obtenidos para la disminución del tiempo de ejecución del código y la disminución del número de veces que se ejecutó SMART-DT están condicionados a la configuración del ejemplo, ya que con otros ejemplos el número de veces que se ejecuta el algoritmo de salto puede aumentar o disminuir, lo importante es que el gasto computacional asociado al código se redujo significativamente.

Luego de haber evaluado el nuevo código y validado su efectividad, se realizó el ejemplo del jet corporativo Capstone, pero esta vez se amplió el número de posibles tiempos de inspección a 28 con el fin de obtener distribuciones más cercanas a casos reales donde los posibles tiempos de inspección son mayores a 30. También se conservó el riesgo de falla en 10^{-7} . Los datos de entrada para este ejemplo se muestran en las tablas 8, 9, 10 y en la figura 30.

Tabla 8. Datos de entrada análisis fractura mecánica ejemplo Capstone

Variable	Dist. Type	Mean	St.Dev.	Notes
Initial Crack Size	Lognormal	0.009055 in	0.001252	Reamed fastener hole
Repair Crack Size	Lognormal	0.00248 in	0.0129	Assuming Repair is Replacement of part
Fracture Toughness	Normal	37.0 ksi	2.0	7050- T651 Plate
EVD	Gumbel	16.74 ksi	2.08	

Tabla 9. Datos de entrada tipo de inspecciones ejemplo Capstone

Insp ID	Inspection Tpe	Material	Crack Type	Dist. Type	Mean [in]	St. Dev. [in]	Source	Cost
POD 1 	Automate bolt hole eddy current	Aluminum	T	Lognormal	0.0180	0.0109	Aeronautical Applications of Non-destructive	5x
POD 2 	Eddy current sliding probe	Aluminum	Overall	Lognormal	0.0788 +0.0625	0.0302	NDE Capabilities Book	1x
POD 3 	Visual	Aluminum		Lognormal	0.99714 +0.0625	3.66907	NDE Capabilities Book	0.1x

Tabla 10. Datos de entrada posibles tiempos de inspecciones ejemplo Capstone

Possible inspection times													
1k	2k	3k	4k	5k	6k	7k	8k	9k	10k	11k	12k	13k	14k
													
15k	16k	17k	18k	19k	20k	21k	22k	23k	24k	25k	26k	27k	28k
													

Para las condiciones de este ejemplo, el código encontró un total de 89 combinaciones de inspección que tienen el mismo costo y son las más económicas entre todas las combinaciones posibles, entre todas estas se puede medir la probabilidad de falla acumulada, con el fin de elegir una sola de estas combinaciones ya optimizadas. El tiempo de ejecución fue de 6 horas 25 min y se realizaron un total de 23801 ejecuciones en SMART-DT para obtener el resultado. Para este ejemplo, se puede apreciar en la tabla 10 que se evaluó la técnica de inspección visual en todos los tiempos de inspección posibles, cada 3000 vuelos se evaluaron las técnicas de inspección visual y Eddy current sliding probe y finalmente cada 6000 vuelos se evaluaron las tres técnicas de inspección utilizadas: inspección visual, Eddy current sliding probe y Automate bolt hole Eddy current. Los resultados de estas combinaciones son mostrados en la figura 30.

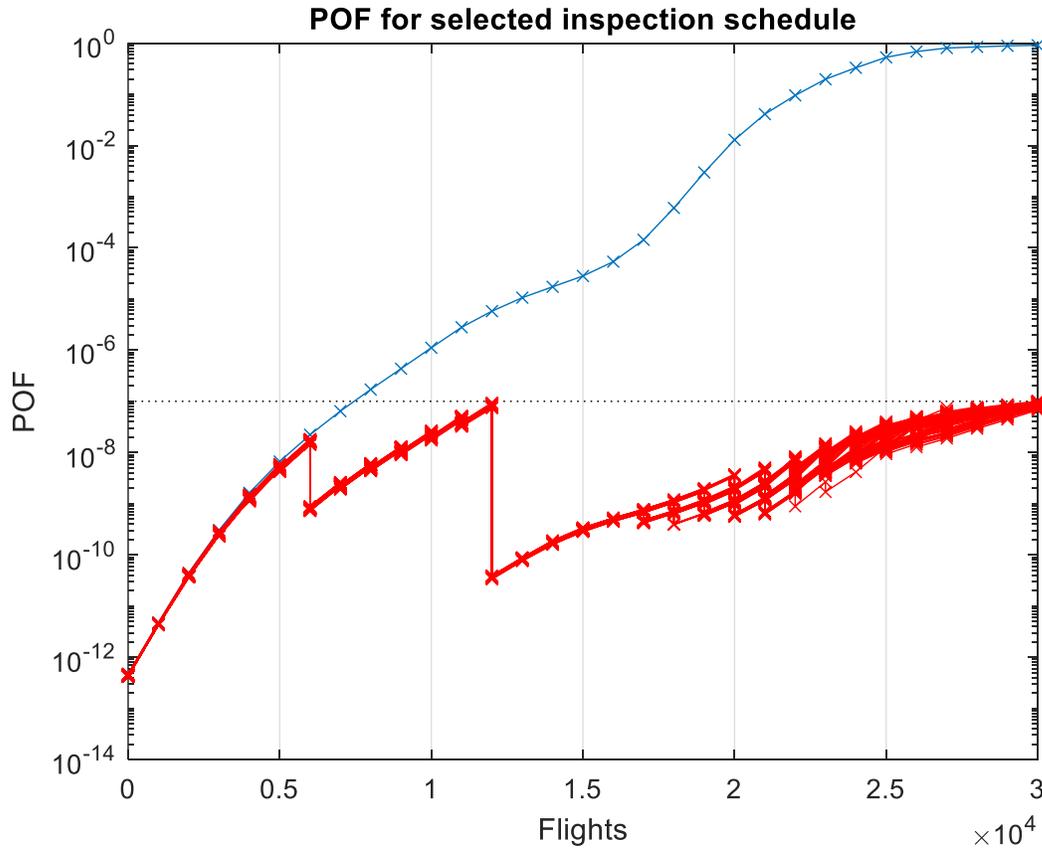


Fig.30. Combinación de inspecciones seleccionada

Por último, se presenta los resultados de un ejemplo realizado en Octave comparado con los resultados obtenidos en MATLAB. Se utilizó el lenguaje de programación Octave porque es un código libre, lo cual facilita el acceso al código a cualquier usuario y reduce los costos ya que no se debe pagar por licencias. Los datos de entrada para realizar esta comparación son los mismos datos usados en el ejemplo del Handbook (tablas 4, 5 y 6; y figuras 28 y 29). En las figuras 31 y 32 se muestran los resultados de esta comparación, se puede ver que los resultados de MATLAB y Octave son los mismos, esto muestra que la implementación del código en Octave fue exitosa y se puede seguir avanzando en cualquiera de los dos lenguajes para reducir aún más el tiempo de ejecución del código.

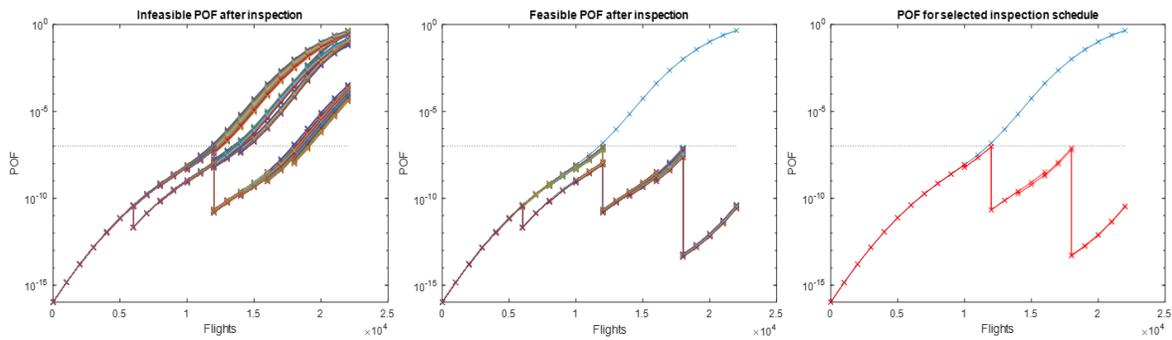


Fig.31. Resultados obtenidos con el código en MATLAB

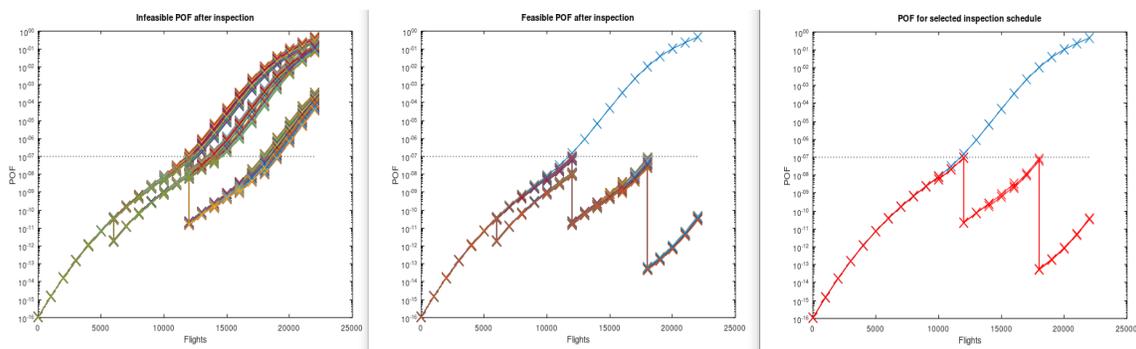


Fig.32. Resultados obtenidos con el código en Octave

IX. CONCLUSIONES

- De acuerdo con los resultados, se puede mejorar significativamente la vida útil de una aeronave al aplicar técnicas de inspecciones basadas en riesgo.
- Es importante tener una buena base de datos para generar un correcto análisis, sin embargo, tener muchos datos que analizar limita la capacidad de hacerlo de forma eficiente lo que hace necesario implementar medidas cada vez más avanzadas de análisis de datos.
- Así como se demostró en este trabajo, saber las condiciones reales a las cuales están sometidos los plenes de mantenimiento de una aeronave sirve como un filtro para manejar los datos y generar algoritmos que se acerquen a las verdaderas condiciones de operación.
- A pesar de haber reducido en gran medida el tiempo de ejecución del código presentado, todavía se necesita optimizar más su tiempo de ejecución, pues en casos no tan idealizados como este, se deben hacer análisis más extensos para encontrar una solución al calendario de inspecciones óptimo.
- Es de gran importancia crear programas (y sus códigos) que sea posible utilizarlos en computadores estándar, ya que esto garantiza su viabilidad en el mercado.

X. REFERENCIAS

- [1] "Section 2.3. Residual Strength Methodology". *Handbook for Damage Tolerant Design. LexTech*. Retrieved 13 March 2017
- [2] J. Ocampo *et al.*, "Development of a probabilistic linear damage methodology for small aircraft," in *Journal of Aircraft*, 2011, vol. 48, no. 6, pp. 2090–2106, doi: 10.2514/1.C031463.
- [3] H. Millwater, J. D. Ocampo, and T. Castaldo, "Probabilistic damage tolerance analysis for general aviation," in *Advanced Materials Research*, 2014, vol. 891–892, doi: 10.4028/www.scientific.net/AMR.891-892.1191.
- [4] Mohamad Modarres, "Reliability engineering and risk analysis".US (2017)
- [5] H. Millwater, J. D. Ocampo, and N. Crosby "Probabilistic methods for risk assessment of airframe digital twin Structures" (2019)
- [6] H. Millwater, J. D. Ocampo, B. Gamble, C. Hurst, M. Nuss, S. Mottaghi and N. Crosby "Probabilistic Damage Tolerance for Aircraft Fleets Using the FAA-Sponsored SMART|DT Software" in *29th ICAF Symposium - Nagoya* (2017)
- [7] H. Millwater, J. D. Ocampo, B. Gamble, C. Hurst, M. Nuss, S. Mottaghi and N. Crosby "An Ultrafast Crack Growth Lifting Model to Support Digital Twin, Virtual Testing, and Probabilistic Damage Tolerance Applications" (2019)
- [8] A.C. Estes and D.M Frangopol "An Optimized Lifetime Reliability-Based Inspection Program for Deteriorating Structures"
- [9] A.A. Kale, R.T. Haftka and B.V. Snkar "Efficient Reliability-Based Design and Inspection of Stiffened Panels Against Fatigue" in *Journal of Aircraft. Vol 45, No.1. (2008)*
- [10] Z. Yang and G. Yang "Optimization of Aircraft Maintenance plan based on Genetic Algorithm" in *International Conference on Medical Physics and Biomedical Engineering* (2012)
- [11] J. Luque and D. Straub "Algorithms for optimal risk-based planning of inspections using influence diagrams" *Engineering Risk Analysis Group, Technische Universität München, Germany*
- [12] C. Han, T. Ma and S. Chen" Asphalt pavement maintenance plans intelligent decision model based on reinforcement learning algorithm"
- [13] Z. He, K.P. Tran, S. Thomassey, X. Zeng, J. Xu and C. Yi "Multi-objective optimization of the textile manufacturing process using deep-Q-network based multi-agent reinforcement learning"

[14] S. Noack, D. Leiby, C. Ritz, N. Moran, and M. Parrish “Aircraft Maintenance” [Online] Disponible en: <https://www.elliottaviation.com/aircraft-maintenance/>

[15] H. Millwater, J. D. Ocampo, B. Gamble, C. Hurst, M. Nuss, and N. Crosby “Small Aircraft Risk Technology- SMART- DT” [Online] Disponible en: <https://smartdtsoftware.wixsite.com/smart/software>

[16] J. Ocampo, D. Posso, and H.R. Millwater “Optimized Risk Based Inspections for Aircraft Fleets” Aircraft Airworthiness & Sustainment Conf., Virtual, August 2020.

[17] E. Weisstein. "Extreme Value Distribution" [Online] Disponible en: <https://mathworld.wolfram.com/ExtremeValueDistribution.html>