UNIVERSIDADE
BEIRA INTERIOR

Estudo de desenvolvimento de um algoritmo de determinação de fiabilidade da aeronave em fase de projeto na lógica da falha catastrófica

Diego Gabriel Rodriguez Côrte-Real

Dissertação para obtenção do Grau de Mestre em
Engenharia Aeronáutica
(Mestrado integrado)

Orientador: Prof. Doutor José Manuel Lourenço da Saúde

Covilhã, 2025

Folha em branco

Declaração de Integridade

Eu, Diego Gabriel Rodriguez Côrte-Real, que abaixo assino, estudante com o número de inscrição 43300 de/o Mestrado Integrado em Engenharia Aeronáutica, da Faculdade de Engenharias, declaro ter desenvolvido o presente trabalho e elaborado o presente texto em total consonância com o **Código de Integridades da Universidade da Beira Interior**.

Mais concretamente afirmo não ter incorrido em qualquer das variedades de Fraude Académica, e que aqui declaro conhecer, que em particular atendi à exigida referenciação de frases, extratos, imagens e outras formas de trabalho intelectual, e assumindo assim na íntegra as responsabilidades da autoria.

Universidade da Beira Interior, Covilhã 5 /03 /2025

Assinado por: **Diego Gabriel Rodriguez Corte Real**
Num. de Identificação: 15436614
Data: 2025.03.05 14:33:53+00'00'

Folha em branco

Agradecimentos

Em primeiro lugar, gostaria de expressar a minha profunda gratidão ao Professor Doutor José Manuel Mota Lourenço da Saúde, cujo apoio, orientação e disponibilidade foram fundamentais para o desenvolvimento deste trabalho.

Aos meus pais, Hilda Morales e Paulo Côrte-Real, não existem palavras que façam justiça ao quanto vos estou grato. Pelo apoio incondicional, pelos sacrifícios feitos ao longo da minha vida e, especialmente, pelo suporte inestimável durante estes últimos anos. Sem vocês, nada disto teria sido possível. Eternamente grato por tudo o que fizeram por mim.

À minha irmã, Andreia Côrte-Real, o meu mais sincero obrigado. Pelos momentos de apoio e por todas as memórias partilhadas que tornaram este percurso mais leve e especial.

Aos meus amigos e família, que me acompanharam nesta jornada, um profundo agradecimento. Pelo encorajamento constante, pelas conversas que aliviaram a pressão, pelas celebrações das pequenas vitórias e, sobretudo, por estarem sempre presentes. O vosso apoio foi essencial e tornou este caminho muito mais significativo.

Folha em branco

Resumo

Num setor onde a segurança é a principal prioridade, a fiabilidade e a manutenção desempenham um papel essencial na prevenção de acidentes, garantindo a integridade dos sistemas e otimizando os processos de manufatura. A consideração destes fatores não só contribui para a minimização dos custos ao longo do ciclo de vida da aeronave, como também prolonga a sua vida útil, resultando numa operação mais eficiente e numa maior satisfação dos operadores e utilizadores.

Com a evolução constante da indústria aeronáutica, a necessidade de aumentar a disponibilidade das aeronaves e reduzir os custos operacionais tem impulsionado a adoção de programas de fiabilidade, que se têm revelado ferramentas fundamentais para a gestão e otimização da manutenção.

Neste contexto, esta dissertação propõe e avalia a viabilidade da implementação de um programa de determinação de fiabilidade para a aeronave LUS222, atualmente em desenvolvimento em parceria com o CEiiA, no enquadramento dado pela probabilidade de falha catastrófica.

Palavras-chave

Programa de fiabilidade; Análise de fiabilidade; Fiabilidade de projeto; Árvore de falhas; Gestão da fiabilidade; MTBF

Folha em branco

Abstract

In a sector where safety is the number one priority, reliability and maintenance play a key role in accident prevention by ensuring the safety of systems as well as optimizing manufacturing processes. The inclusion of these requirements helps minimize life cycle costs and increase the lifespan of aircraft, resulting in greater customer satisfaction.

With the continuous development of the aviation industry, there is a growing demand for increased aircraft availability and reduced maintenance costs, making it necessary to implement reliability programs due to the success they have brought to the industry.

In this context, this master's thesis proposes and evaluates the feasibility of implementing a reliability determination program for the LUS222 aircraft, currently under development in partnership with CEiiA, within the framework provided by the probability of catastrophic failure.

Keywords

Reliability Programme; Reliability Analysis; Design for Reliability; Fault Tree Analysis; Reliability Management; MTBF

Folha em branco

Índice

Folha em branco

Lista de figuras

Folha em branco

Lista de tabelas

Folha em branco

Lista de siglas e de acrónimos

AAN	Autoridade Aeronáutica Nacional
AMC	Acceptable Means of Compliance
ANAC	Autoridade Nacional de Aviação Civil
CM	Condition Monitoring
CTI	Circular Técnica de Informação
CVE	Compliance Verification Engineer
DfR	Design for Reliability
EASA	European Aviation Safety Agency
ESS	Environment Stress Screening
FC	Falha Catastrófica
FDP	Função Densidade de Probabilidade
FMEA	Failure Mode and Effect Analisys
FMECA	Failure modes, Effects, and Criticality Analysis
FTA	Fault Tree Analisys
GM	Guidance Material
GNSS	Global Navigation Satellite System
IATA	International Air Transport Association
IFSDR	In-Flight Shutdown Rate
ILS	Integrated Logistics Support
LCC	Life Cycle Cost
LRU	Line Replaceable Units
MEL	Minimum Equipment List
MMEL	Master Minimum Equipment List
MSG	Maintenance Steering Group
MSI	Maintenance Significant Item
MTBF	Mean Time Between Failure
MTBUR	Mean Time Between Unscheduled Removals
MTOW	Maximum Take-off Weight
MTTR	Mean Time To Repair
PF	Programa de Fiabilidade
PMA	Programa de Manutenção de Aeronaves
RPK	Revenue Passenger Kilometers
VOR	Very High Frequency (VHF) Omnidirectional Range

Folha em branco

Capítulo 1 - Generalidades

1.1 Introdução

Ao entrar no novo milénio, o desenvolvimento económico global, impulsionado pela globalização de negócios e do turismo, gerou um aumento na procura pelo transporte aéreo. As companhias aéreas, por sua vez, tiveram de se adaptar a este crescimento acelerado da indústria, ilustrado na Figura 1.

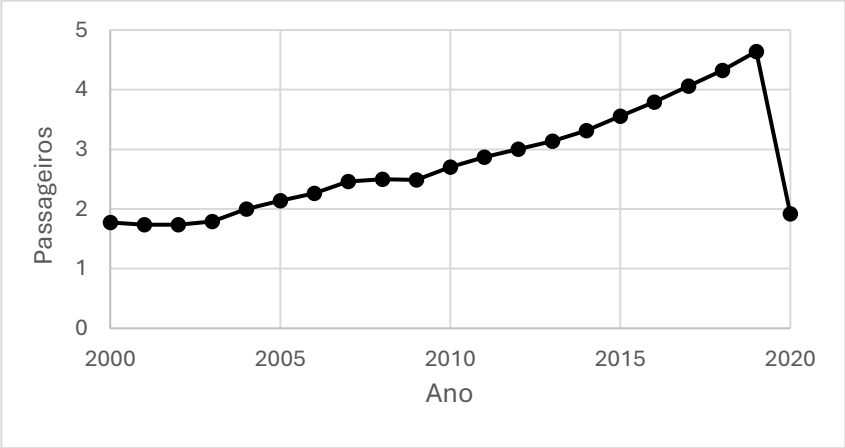


Figura 1 - Passageiros (milhares de milhão) por ano

Fonte: (IEA, 1980-2020)

Esta procura foi, no entanto, interrompida em meados de 2019 devido à pandemia global COVID-19, que colocou as operações aéreas em *stand-by*.

De acordo com a IATA (International Air Transport Association), a partir do início de 2020, a procura pelo transporte aéreo retomou, alcançando patamares pré-pandemia e apresentando uma tendência de crescimento constante, ilustrado na Figura 2.

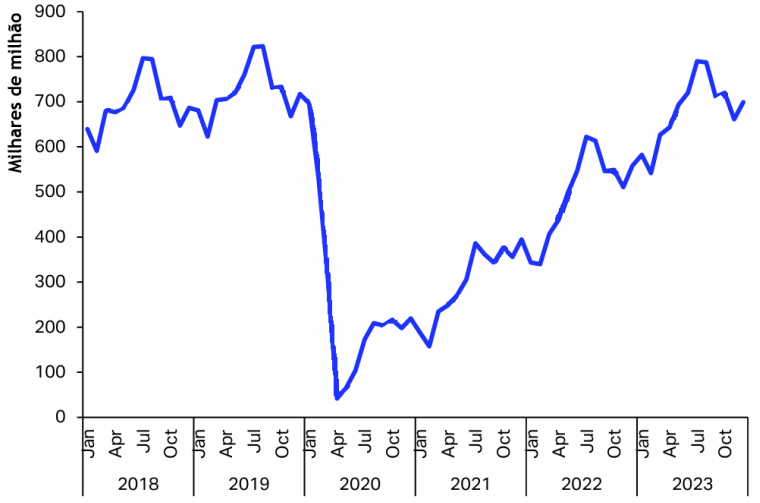


Figura 2 - RPK (milhares de milhão) por mês

Fonte: adaptado de IATA (2023, p. 2)

O gráfico mostra uma evolução contínua dos valores de RPK (Revenue Passenger Kilometers), que é um indicador do número de quilómetros percorridos por passageiros (Aeroflot Russian Airlines, 2022, p. 1).

Para responder a esta nova e crescente procura pela indústria, é essencial que as companhias aéreas desenvolvam métodos que possam sustentar operações de forma a garantir uma exploração ideal[1].

"Um dos principais desafios enfrentados pela indústria aeronáutica consiste em reduzir o tempo do ciclo de design e de produção de uma aeronave ao máximo possível, tendo em conta três aspetos importantes: rapidez na entrega aos clientes, qualidade e segurança, e custos." (Vieira, Rebaiaia, & Chain, 2016, p. 968).

Com base na referência anterior, estes meios a ser desenvolvidos devem ter sempre em conta três fatores importantes:

- Rapidez na entrega aos clientes (*time to market*) de novas aeronaves- Reduzir o tempo para projetar, fabricar e entregar uma aeronave aos clientes é crucial para responder à procura do mercado, manter a competitividade e aumentar a satisfação do cliente.

- Qualidade e segurança – A indústria prioriza qualidade (produção, aeronáutica, percebida) e segurança (operações de voo, procedimentos em terra) como pilares interdependentes. O cumprimento de ambos é essencial para a fiabilidade, integridade operacional e sucesso sustentável das companhias aéreas.

- Custos – O acompanhamento constante dos custos é essencial para garantir a competitividade e sustentabilidade das empresas na indústria. Estes custos podem ser recorrentes, ocorrendo regularmente ao longo do tempo e das missões, como manutenções programadas, mão de obra, entre outros; ou não recorrentes, surgindo de forma esporádica e fora do orçamento operacional regular, como em casos de acidentes, mudanças na legislação, entre outros.

O foco desta dissertação incide sobre a qualidade e segurança abordado na ótica da fiabilidade.

[1] Procura pela utilização eficiente dos recursos disponíveis, maximizando a eficiência operacional e garantindo serviços de qualidade e elevada fiabilidade aos passageiros.

Ao reconhecer a importância da fiabilidade, as companhias aéreas, além de utilizarem programas de manutenção de aeronaves (PMA), outros programas como os de fiabilidade (PF) têm vindo a provar-se cada vez mais importantes e eficazes no aumento de fiabilidade dos componentes que constituem a aeronave e, como resultado, observa-se uma otimização significativa, tanto na redução dos custos operacionais quanto na extensão dos intervalos de manutenção.

Um programa de fiabilidade pode ser dividido em duas fases principais: a fase de projeto da aeronave e a fase operacional. Durante a fase de projeto, o foco está em estabelecer as bases para a fiabilidade da aeronave antes mesmo de ela entrar em serviço, através de processos que variam desde a definição dos objetivos de segurança e de fiabilidade até às análises de condições de falha para todos os componentes que constituem a aeronave.

Neste contexto, surge a necessidade de desenvolver ferramentas que permitam avaliar a fiabilidade de uma aeronave já na fase de projeto, assegurando que os sistemas críticos cumprem os requisitos estabelecidos. O presente trabalho surge como resposta a essa necessidade, através da implementação de um programa de avaliação da fiabilidade para a aeronave LUS222, que visa estruturar e analisar a fiabilidade dos seus sistemas ainda antes da entrada em operação.

Uma vez que a aeronave termina a ciclo de design e é entregue ao cliente, esta entra na sua fase operacional. Aqui, o programa de fiabilidade foca-se em seguir um conjunto de procedimentos e práticas que gerem e controlam um programa de manutenção, através da obtenção e coleta de dados; análise estatística dos dados coletados; e definição de ações corretivas (Kinnison & Siddiqui, 2012, p. 222).

Além disso, um programa de fiabilidade também tem como objetivo melhorar padrões de operação principalmente a partir de aspetos técnicos, mas também desencadear mudanças nos procedimentos operacionais, bem como mudanças no projeto do sistema da aeronave.

O PF fornece informação sobre a fiabilidade de cada sistema e componente específico da aeronave. Essa mesma informação pode ser comparada com dados de fiabilidade de projeto (*design reliability*)[2]. A partir desta comparação é possível identificar sistemas ou componentes que estão a apresentar desempenho abaixo do esperado e tomar ação através da implementação de medidas corretivas.

Estas podem abranger desde intervenções específicas para aprimorar procedimentos até alterações mais amplas e estruturais no programa fundamental de manutenção. Estas medidas

[2] A fiabilidade de projeto é a que é apresentada pelo fabricante de cada sistema e é a referência que é usada para estabelecer o programa de manutenção.

geralmente resultam em uma ou mais das seguintes ações: modificações de equipamentos; alterações em processos ou práticas de manutenção; substituição de peças defeituosas ou mudança de fornecedores; treino adicional; inclusão de novas tarefas de manutenção; ajustes nos intervalos de manutenção (Kinnison & Siddiqui, 2012, p. 231).

Desta forma, o programa de fiabilidade não apenas identifica problemas, mas também implementa ações para melhorar continuamente o programa de manutenção e a fiabilidade geral da aeronave.

1.2 Objetivo

Definir um modelo de avaliação da fiabilidade relativa à aeronave LUS 222, cujo projeto está em curso no CEIIA, de modo a poder constituir contributo para elaboração do respetivo processo e cumprir as exigências que habilitam à certificação, tendo em vista o cumprimento da fiabilidade prevista para a falha catastrófica.

1.3 Limites do trabalho

Este trabalho encontra-se limitado ao facto da aeronave ainda se encontrar na fase de projeto preliminar pelo que não estão definidos cada um dos sistemas, equipamentos, bem como a respetiva arquitetura.

Isto significa que no desenvolvimento do modelo de definição da fiabilidade os dados que serão usados (relativos aos equipamentos) serão os que resultarem da informação aberta disponível no mercado.

De igual modo, um importante limite diz respeito à base temporal que foi usada para a realização desta dissertação a qual é relativamente pequena face à base temporal de desenvolvimento da aeronave LUS-222.

Por fim, os limites específicos para desenvolvimento do algoritmo que esteve na base da solução preconizada estão definidos no respetivo capítulo.

1.4 Metodologia

A metodologia utilizada para desenvolver esta dissertação consiste em uma pesquisa bibliográfica de artigos, livros e publicações que contêm informações afetas a este tipo de estudo.

Inclui-se ainda o recurso à regulamentação disponível na EASA no domínio da certificação CS 23, sendo usada a CS 25 sempre que aquela for incompleta ou omissa.

O desenvolvimento da metodologia de análise da fiabilidade da aeronave ao longo da fase de projeto envolve vários passos, que estão explícitos em baixo:

- Efetuar o estado da arte do processo de determinação da fiabilidade;
- Recolher dados afetos a este tipo de estudo;

- Desenvolver uma arquitetura base dos sistemas afetos à aeronave e definição dos modos de falha de cada um;
- Atribuir e quantificar a fiabilidade dos sistemas e componentes;
- Realizar uma análise de importância e de sensibilidade, com o objetivo de identificar os componentes críticos;
- Realizar análises temporais dos componentes e sistemas considerados.

1.5 Estrutura do trabalho

O conteúdo desta dissertação está dividido em 6 capítulos.

No presente capítulo procura-se introduzir e definir os objetivos deste estudo enquadrados com o conceito de fiabilidade na indústria aeronáutica.

Ainda no mesmo capítulo é também definida a metodologia usada para a realização do trabalho, bem como a estrutura do mesmo, de forma a proporcionar uma fácil orientação ao leitor.

A caracterização da empresa e da aeronave em estudo compõem o segundo capítulo, ao passo que o terceiro capítulo contém o estado da arte, onde é realizada uma abordagem teórica de forma a dominar os conceitos de fiabilidade dentro da indústria aeronáutica.

O quarto capítulo descreve uma metodologia para a criação e implementação da avaliação da fiabilidade prevista para a aeronave LUS-222 tendo por base critérios de falha catastrófica.

No quinto capítulo é realizada uma síntese sobre o processo de viabilidade de adoção da metodologia de determinação da fiabilidade proposta por parte da empresa.

Por fim, no sexto capítulo são enunciadas as conclusões e recomendações para trabalhos futuros.

Capítulo 2 – CeiiA vs projeto LUS-222

2.1 Introdução ao CEiiA

O CEiiA (Centro de Engenharia e Desenvolvimento) é uma empresa portuguesa fundada em 1999 de engenharia e desenvolvimento que concebe, desenvolve e opera produtos de diferentes indústrias, automóvel, aeronáutica, mar e espaço, seguindo sempre a filosofia de tecnologia a partir da sustentabilidade (*sustainability by design*).

Dentro do setor aeronáutico a empresa apresenta um portfolio alargado de diferentes projetos e colaborações com empresas como Embraer, Leonardo Finmeccanica e Daher.

As atividades de desenvolvimento do produto no âmbito da engenharia aeronáutica estão focadas no *design*, análise estrutural, estudos aerodinâmicos e interiores de aeronaves.

Com estas parcerias, o CEiiA participa de programas de reconhecimento à escala global[3], como por exemplo, a aeronave Embraer KC-390, onde a empresa teve um grande impacto no ciclo de desenvolvimento da aeronave, em partes como a fuselagem central, estrutura do leme de profundidade e os *sponsons* da aeronave que, até à data, é a maior aeronave – quanto à superfície alar e peso máximo à descolagem – produzido pela Embraer.

Outro projeto no qual o CEiiA tem parceria é com a companhia Leonardo Finmeccanica, onde participa em diferentes fases de desenvolvimento de seis helicópteros da Leonardo, incluindo atividades de design, análises estruturais e redução de peso.

2.2 LUS 222

Um dos projetos mais recentes no qual o CEiiA se tornou responsável pela engenharia de desenvolvimento foi o projeto da aeronave LUS 222. Segundo o fabricante, o nome "LUS" tem o significado de luz de Portugal, e os três "2" significam dois motores, dois mil quilómetros de alcance e duas toneladas de carga, sendo isto algumas características da aeronave.

Esta aeronave será a primeira desenvolvida e industrializada em Portugal, com todos os componentes fabricados dentro do país, à exceção dos motores e do trem de aterragem, bem como dos sistemas comummente desenvolvidos por terceiros.

[3] Projetos de desenvolvimento que ganham notoriedade e prestígio internacional.

Figura 3 - Protótipo da aeronave LUS 222 em miniatura
Fonte: (Pinto, 2022)

O desenvolvimento da engenharia do projeto está a ser realizado no Parque do Alentejo de Ciência e Tecnologia, em Évora, e a industrialização será realizada no Aeródromo Municipal em Ponte de Sôr, em Portalegre. O objetivo do projeto é ter a primeira aeronave fabricada para certificação até ao final de 2025 ou primeiro trimestre de 2026 e o *maiden flight*[4] até ao final de 2027.

Esta aeronave destina-se, na versão civil, essencialmente aos mercados da América do Sul, Norte de África e Sudeste Asiático, para dar uma alternativa ao trânsito por terra, que é muitas vezes difícil e demorado. Na versão militar, esta será destinada às forças armadas dos países das mesmas regiões, por ser uma aeronave todo-o-terreno, capaz de aterrar em pistas muito curtas e não preparadas (AICEP, 2024).

O LUS-222 tem como objetivo responder a uma procura por este tipo de aeronaves estimada em mais de cinco mil aviões nos países de destino para substituir as aeronaves em fim de vida nos próximos quinze anos (Morgado, 2022).

Na Tabela 1 encontram-se algumas características já conhecidas da aeronave.

[4] A primeira ocasião em que a aeronave descola utilizando os seus próprios sistemas de propulsão, sem auxílio externo, também conhecido por rol-out.

Tabela 1 - Características da aeronave LUS 222

Fonte: (Morgado, 2022)

LUS 222	
Capacidade	19 lugares
Massa em vazio (OEW) [kg]	3.700
Massa máxima à descolagem (MTOM) [kg]	8.200
Carga [kg]	2.000
Alcance [km]	2.000
Comprimento da aeronave [m]	16.5
Altura da aeronave [m]	6.5
Velocidade cruzeiro [km/h]	370
Distância de aterragem* [m]	450
Distância de descolagem* [m]	900

*A aterragem e descolagem podem ser realizadas em pistas curtas e não pavimentadas.

Capítulo 3 – Estado da arte

3.1 Conceitos básicos de fiabilidade

A fiabilidade, através da sua definição mais abrangente e de acordo com o dicionário da língua portuguesa, é algo (pessoa, serviço ou produto) que é digno de confiança (Priberam, 2008-2021).

É a medida do quão confiável é um produto, serviço ou pessoa para alguma tarefa em questão. É um conceito usado no dia-a-dia de todos. Por exemplo, quando doentes, maior parte das pessoas visita um médico e não um advogado, isto porque é depositada mais confiança nos médicos para curar doenças do que nos advogados (Ribeiro de Oliveira, 2015, p. 2).

Confiança, no entanto, não é uma quantidade mensurável. Portanto, olhando para uma definição do ponto de vista da engenharia, a fiabilidade é a probabilidade de um sistema ou componente operar sem falhas, num dado ambiente e em determinadas condições durante um certo período (Kinnison & Siddiqui, 2012, p. 217).

Um termo muito importante ao falar de fiabilidade, é o conceito de fiabilidade inerente. Kinnison e Siddiqui (2012, p. 8) explicam a fiabilidade inerente como sendo a fiabilidade máxima que um sistema pode alcançar. Esta é estabelecida pelas escolhas realizadas na fase de projeto/*design* do mesmo. É reconhecido como um atributo de *design* e nenhuma manutenção irá aumentar este nível inerente do sistema, no entanto é ideal que o operador mantenha este nível de fiabilidade a todos os tempos.

A única maneira de aumentar o nível inerente de fiabilidade de um sistema é através de um *redesign* do mesmo, isto é, modificar escolhas feitas inicialmente na fase de projeto/*design*, como por exemplo, novos materiais ou diferentes processos de manufatura. Desta maneira, o nível de fiabilidade inerente do sistema poderá aumentar tal como ilustrado na Figura 4.

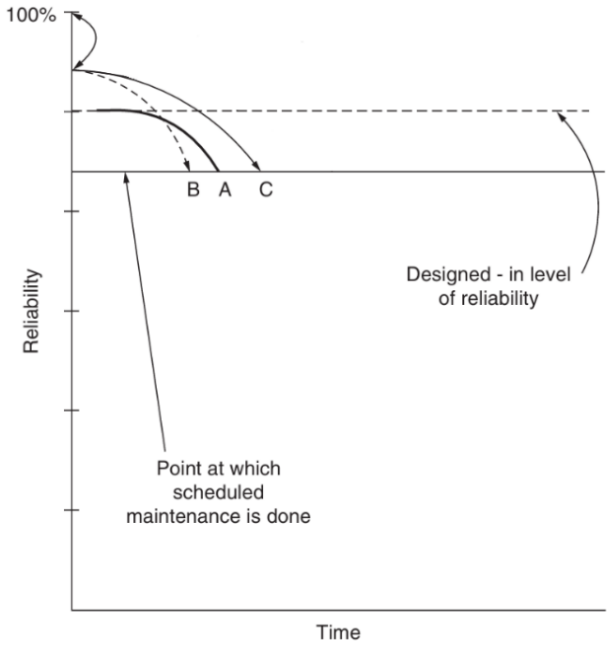


Figura 4 - Efeitos de redesign na fiabilidade de sistemas
Fonte: adaptado de Kinnison e Siddiqui (2012, p. 8)

No gráfico acima, a curva A representa um sistema inicial, o qual teve as suas escolhas de *design* e assim as manteve, não sofrendo *redesign*. As curvas B e C representam sistemas que sofreram *redesign*, o que causou um aumento do nível de fiabilidade inerente, ao reduzir a entropia natural do sistema. No entanto, apesar de se ter diminuído a entropia do sistema, é inevitável que o mesmo deteriorar. É muito provável que a taxa de deterioração varie do sistema inicial, dependendo de inúmeros fatores, o que se traduz nos diferentes declives das curvas dos diferentes sistemas (Kinnison & Siddiqui, 2012, p. 9).

Uma deterioração mais rápida, (sistema B) atinge o ponto onde será necessária manutenção programada mais cedo, e os subsequentes intervalos de manutenção serão curtos o que indica que manutenção será necessária mais frequentemente. Neste caso, apesar da fiabilidade inerente aumentar, mais manutenção será necessária para manter esse mesmo nível, e, portanto, a escolha de *redesign* do sistema pode não ser aceite.

Da mesma maneira, quando ocorre uma deterioração lenta do sistema (sistema C), os intervalos de manutenção subsequentes serão maiores e, portanto, será necessário menos manutenção para manter o nível de fiabilidade inerente do sistema. Neste caso, a escolha de *redesign* poderá ser aceite dependendo se a redução de manutenção justifica o custo do *redesign*.

3.1.1 Função fiabilidade

A abordagem da fiabilidade de um sistema está associada a um intervalo de tempo de operação efetivo, que é habitualmente medido em horas de operação. No entanto, outras unidades também são utilizadas como por exemplo, número de ciclos de operação ou distância percorrida.

No seguimento desta secção, descrevem-se alguns termos utilizados para medir a fiabilidade, tais como a função fiabilidade, função de taxa de falha, entre outros. Importa realçar que, numa população de um determinado produto, assume-se que os produtos falham em tempos diferentes, mesmo que funcionem sob as mesmas condições. Deste modo, o fenómeno da falha deve ser tratado estatisticamente. É por esta razão que a definição dos conceitos básicos de fiabilidade se baseia na teoria da probabilidade.

Através da definição, já mencionada, a fiabilidade de um sistema representa a probabilidade do mesmo funcionar sem falhas durante um período.

Neste caso, apenas dois eventos poderão acontecer, o sistema funcionar, ou não. Estes dois eventos chamam-se eventos complementares (porque se um ocorre, ou outro não pode existir) e a expressão matemática é dada por (Weibull.com, 2001, p. 3):

$$R(t) + F(t) = 1 \tag{1}$$

Sendo $R(t)$ a fiabilidade do sistema, isto é, a probabilidade que o mesmo funcione durante um certo período t, e $F(t)$ é a probabilidade de falha do mesmo.

A fiabilidade de um sistema, por ser uma probabilidade, é governada pela seguinte equação:

$$0 \leq R(t) \leq 1 \tag{2}$$

Como a fiabilidade se fundamenta nos princípios da teoria da probabilidade, a partir daí emergem duas funções essenciais para o desenvolvimento do seu conceito: a função densidade de probabilidade (FDP) e a função de falha acumulada. Estas funções permitem descrever e prever o comportamento de sistemas e componentes em relação às suas falhas, proporcionando uma base para modelar e quantificar a fiabilidade ao longo do tempo.

- Função Densidade de Probabilidade (FDP), esta função descreve uma distribuição normal e indica o quão provável é uma variável aleatória (X) assumir um valor específico dentro de um intervalo. A expressão matemática que descreve a definição da função é a seguinte (Verma, 2016, p. 29):

$$P(a \leq X \leq b) = \int_a^b f(x)dx \tag{3}$$

Onde $a \leq b: a, b \in \mathbb{R}$.

A FDP é denotada por $f(x): x \in \mathbb{R}$, no entanto, dentro do tema da fiabilidade a função é denotada por $f(t): t \in \mathbb{R}_0^+$.

Uma propriedade muito importante da FDP é que, por ser uma curva de densidade, cada instante de tempo equivale a uma probabilidade e, portanto, somando todas as probabilidades existentes, tem-se que:

$$\int_0^\infty f(t)dt = 1 \tag{4}$$

- Função de Falha Acumulada, que descreve a probabilidade de uma variável aleatória (X) assumir um valor menor ou igual a um valor específico.

  A expressão matemática é dada por (Verma, 2016, p. 29):

$$P(X \leq x) = F(x) \tag{5}$$

  Esta função é habitualmente denotada por $F(x)$, no entanto, estando no tema da fiabilidade a função é denotada por $F(t): t \in \mathbb{R}_0^+$. Esta função aparece na expressão $R(t) + F(t) = 1(1)$ e representa a probabilidade de um componente já ter falhado num instante de tempo t.

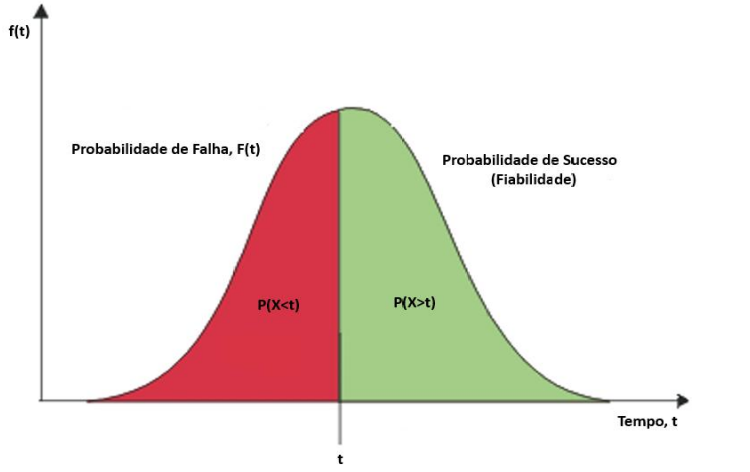


Figura 5 - Função fiabilidade vs função de falha acumulada com a FDP.
Fonte: adaptado de Weibull (2001, p. 3)

No gráfico acima é possível observar a relação entre as funções da expressão $R(t) + F(t) = 1(1)$ com a função densidade de probabilidade. A região colorida a vermelho representa a função $F(t)$, devido à definição matematicamente expressa na equação $P(X \leq x) = F(x)(5)$.

Como os eventos de falha e sucesso são complementares (expressão $R(t) + F(t) = 1(1)$), conclui-se que a região colorida a verde equivale a $1 - F(t) = R(t)$, portanto, a região a verde representa a probabilidade de sucesso do sistema ou a fiabilidade do mesmo.

Relacionando as expressões $R(t) + F(t) = 1(1)$ e $\int_0^\infty f(t)dt = 1\ (4)$ obtém-se a expressão:

$$\int_0^\infty f(t)dt = R(t) + F(t) \tag{6}$$

No gráfico acima, é possível deduzir uma relação entre a função de falha acumulada $F(t)$ e a função densidade de probabilidade $f(t)$ e entre a função de fiabilidade $R(t)$ e a FDP $f(t)$ (Verma, 2016, p. 33):

$$F(t) = \int_0^t f(t)dt \tag{7}$$

$$R(t) = \int_t^\infty f(t)dt \tag{8}$$

Da expressão $R(t) + F(t) = 1(1)$ e $F(t) = \int_0^t f(t)dt\ (7)$ obtém-se a relação inversa da expressão $R(t) = \int_t^\infty f(t)dt\ (8)$ (Verma, 2016, p. 30):

$$F(t) = \int_0^t f(t)dt \Leftrightarrow f(t) = \frac{dF(t)}{dt}$$

$$f(t) = \frac{d[1 - R(t)]}{dt}$$

$$f(t) = -\frac{dR(t)}{dt} \tag{9}$$

Estas fórmulas servem como base para a dedução da expressão de fiabilidade ao longo do tempo, deduzida ao longo dos seguintes capítulos.

3.1.2 Taxas de falha

Define-se por $\lambda(t)$ a taxa de falha instantânea de um componente ou sistema. A taxa de falha representa a probabilidade que um componente não irá sobreviver um tempo adicional *dt* dado que o mesmo sobreviveu até um instante *t* (Ross, 2014, p. 589).

A função $\lambda(t)$ pode ser interpretada como a probabilidade condicional de falha em um pequeno intervalo de tempo $[t, t+dt]$, dado que a unidade ainda se encontra em operação no instante t (Verma, 2016, p. 33).

A expressão matemática da função da taxa de falha é dada por (Verma, 2016, p. 33):

$$\lambda(t) = \lim_{dt \to 0} \frac{P\{X \epsilon (t, t+dt) | X > t\}}{dt} \tag{10}$$

Visto que esta função contém uma probabilidade condicional, é possível utilizar o teorema de Bayes para as probabilidades condicionais:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

De acordo com Sheldon M. Ross (2014, p. 590), tem-se que,

$$P\{X \epsilon (t, t+dt) | X > t\} = \frac{P\{X \epsilon (t, t+dt) \cap X > t\}}{P(X > t)}$$

$$P\{X \epsilon (t, t+dt) | X > t\} = \frac{P(X \epsilon (t, t+dt))}{P(X > t)}$$

$$P\{X \epsilon (t, t+dt) | X > t\} = \frac{P\big(X \epsilon (t, t+dt)\big)}{1 - P(X \leq t)} \tag{11}$$

De forma a deduzir a expressão $P(X \epsilon (t, t+dt))$, é necessário voltar à expressão $P(a \leq X \leq b) = \int_a^b f(x)dx$ (3) de forma a obter:

$$P(X \epsilon (t, t+dt)) = \int_t^{t+dt} f(x)dx$$

Considerando *dt* como uma variação infinitesimal, $f(x)$ será quase constante e igual a $f(t)$ dentro do intervalo $[t, t+dt]$. Desta forma,

$$\int_{t}^{t+dt} f(x)dx \approx f(t)dt$$

E, portanto,

$$P(X\epsilon(t, t+dt)) \approx f(t)dt$$

Substituindo na expressão (11),

$$P\{X\epsilon(t, t+dt)|X > t\} \approx \frac{f(t)dt}{1-F(t)}$$

$$P\{X\epsilon(t, t+dt)|X > t\} \approx \frac{f(t)dt}{R(t)}$$

Regressando à expressão da função da taxa de falha $\lambda(t) = \lim_{dt\to 0} \frac{P\{X\epsilon(t, t+dt)|X > t\}}{dt}$ (10),

$$\lambda(t) = \lim_{dt\to 0} \frac{\frac{f(t)dt}{R(t)}}{dt}$$

$$\lambda(t) = \frac{f(t)}{R(t)} \tag{12}$$

Da expressão $f(t) = -\frac{dR(t)}{dt}$ (9) e $\lambda(t) = \frac{f(t)}{R(t)}$ (12) obtém-se finalmente a expressão da função fiabilidade $R(t)$ (Verma, 2016, p. 34):

$$\lambda(t) = \frac{f(t)}{R(t)} = \frac{-\frac{dR(t)}{dt}}{R(t)} \tag{13}$$

$$\lambda(t)dt = -\frac{dR(t)}{R(t)}$$

Integrando em ambos lados,

$$\int_{0}^{t} \lambda(t)dt = -\int_{0}^{R(t)} \frac{dR(t)}{R(t)} = -\ln R(t)$$

$$R(t) = e^{-\int_{0}^{t} \lambda(t)dt} \tag{14}$$

Desta maneira obtém-se a expressão geral da fiabilidade no instante de tempo t.

Considere-se um sensor de quantidade de combustível da aeronave que segue uma distribuição exponencial, e apresenta uma taxa de falhas constante de $\lambda = 0.0005$ falhas por hora. Se o objetivo da análise for determinar a probabilidade de o sensor funcionar adequadamente ao longo de diferentes períodos de operação, então é necessário realizar os seguintes cálculos:

Considerando um tempo de operação $t = 200$ horas de operação então, a partir da expressão $R(t) = e^{-\int_0^t \lambda(t)dt}$ (14):

$$R(200) = e^{-\int_0^{200} 0.0005dt}$$

Como a taxa de falhas é constante (acontecimento referido nos seguintes capítulos) então:

$$R(200) = e^{-0.0005 \cdot 200} = 0.905$$

Isto significa que existe uma probabilidade de 90.5% de o sensor funcionar corretamente após 200 horas de operação.

Da mesma maneira, através da definição (expressão $P(X \leq x) = F(x)(5)$), também é calculada a probabilidade de falha acumulada do sensor nas primeiras 200h de operação:

$$P(Falha\ para\ t \leq 200) = F(200) = 1 - R(200) = 0.095$$

Portanto, existe uma probabilidade de 9.5% de o sensor falhar nas primeiros 200 horas.

Através da função fiabilidade $R(t) = e^{-\int_0^t \lambda(t)dt}$ (14), é possível observar que a mesma varia consoante a variação da taxa de falha ao longo do tempo.

A partir de testes e análises de fiabilidade realizados no passado, chegou-se a um modelo muito popular que descreve a taxa de falha $\lambda(t)$ de muitas famílias de produtos ou sistemas ao longo do tempo. Este modelo é chamado de Curva da Banheira ou *Bathtub Reliability Curve* (BTRC) devido à sua semelhança com uma banheira (Figura 6) e é composto por três fases diferentes: período de mortalidade infantil (infância), período de vida útil (maturidade) e período de desgaste (fim de vida).

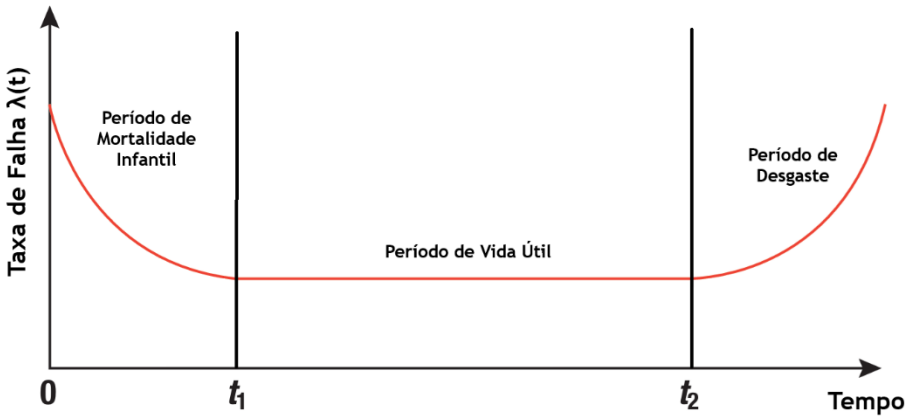


Figura 6 - Curva da Banheira

Fonte: adaptado de Kapur & Pecht (2014, p. 26)

Durante a primeira fase, conhecida como período de mortalidade infantil, a população de produtos apresenta uma taxa de falha elevada que diminui ao longo do tempo devido à eliminação de componentes defeituosos e à estabilização dos processos de fabricação e operação.

Nesta fase, os produtos mais suscetíveis a falhas são identificados e removidos, resultando numa melhoria gradual da fiabilidade do sistema como um todo. Nesta fase, as falhas são frequentemente atribuídas a defeitos de fabricação, controlo de qualidade[5] inadequado ou erros no processo de manufatura.

A taxa de falha eventualmente irá estabilizar num certo instante de tempo t_1, quando os produtos "fracos" já terão falhado (Kapur & Pecht, 2014, p. 26).

Ebeling (2003, p. 32) sugere quatro métodos diferentes para reduzir as falhas dentro deste período, estes métodos são:

1. Testes de *burn-in* ou de *debug*;
2. Environment Stress Screening (ESS);
3. Controlo de qualidade;
4. Testes de validação.

Juran & Godfrey (1998, p. 48.33) vêm os testes de *burn-in* como um tipo de inspeção de 100% da população do produto. Estes consistem em operar todos os produtos de uma população durante um determinado período antes do envio ou instalação, visando identificar potenciais falhas.

[5] Definido como a capacidade de um produto ou serviço atender às expectativas e requisitos do cliente

De forma simples, testes de *burn-in* são procedimentos nos quais os componentes são expostos a condições severas (e.g. elevadas temperaturas, stress) para acelerar falhas potenciais e garantir que os produtos enviados estejam livres de defeitos latentes[6].

Environment Stress Screening (ESS) são testes muito parecidos aos de *burn-in*, onde os componentes ao nível do subsistema são sujeitos a ciclos de temperatura, vibrações mecânicas e regimes de operação intensos para identificar unidades potencialmente defeituosas.

Os testes de *burn-in* inspecionam o produto como um todo, frequentemente em condições constantes de stress e temperatura, enquanto os testes ESS aplicam condições cíclicas e intensas para verificar subsistemas ou componentes em cenários variados. Ambos têm o objetivo de melhorar a fiabilidade, mas com focos e aplicações distintas.

Estes testes são denominados de testes acelerados e, embora eficazes, apresentam limitações: podem introduzir modos de falha que não ocorreriam em condições normais de operação e também são dispendiosos.

Estes testes são atividades de avaliação do produto, mas não substituem melhorias no design e na manufatura. Devem ser aplicados com julgamento e pensamento crítico, no sentido de: ajudam a reduzir defeitos, mas o seu impacto deve ser visto como complementar, o objetivo principal deve ser a melhoria contínua nas fases de design e manufatura. Desta maneira, os defeitos serão minimizados e a necessidade de análise rigorosa é eliminada (Juran & Godfrey, 1998, p. 48.33).

O período de mortalidade infantil é seguido pelo período de vida útil, onde a taxa de falha encontra-se estabilizada e a mesma será mínima e aproximadamente constante, até atingir um instante de tempo t_2. Durante este período, considera-se que as falhas ocorrem devido a fatores aleatórios. Ebeling (2003, p. 32) acrescenta causas como cargas aleatórias e "atos de Deus" ("acts of God[7]") às causas de falha neste período de vida.

Contudo, no contexto aeronáutico, é necessário concentrar a análise de fiabilidade de sistemas e componentes dentro do controlo humano, deixando de fora fatores externos imprevisíveis, como *bird strikes*, raios ou danos causados por detritos (FOD). Portanto, em concordância com este trabalho, eventos denominados de "atos de Deus" não são considerados dentro da análise de fiabilidade dos sistemas da aeronave.

[6] Definido falhas pré-existentes em componentes ou sistemas que permanecem ocultas até serem expostas por condições específicas de operação, stress ou envelhecimento.

[7] Atos de Deus referem-se a eventos naturais incontroláveis, como tempestades, raios, *bird strikes*, entre outros.

De modo a eliminar falhas durante este período, Juran & Godfrey (1998, p. 48.33) vêm o bom controlo do produto e procedimentos de manutenção como necessários para atingir estes objetivos.

Ebeling (2003, p. 32), no entanto, sugere o uso de redundância, que apesar de não reduzir a falha individual de cada componente/subsistema, reduz a probabilidade de falha combinada, o que aumenta a fiabilidade do sistema como um todo.

A última fase, conhecida como período de desgaste ou mortalidade senil, é caracterizada por uma taxa de falha crescente à medida que os componentes envelhecem e se desgastam. Durante esta fase, Ebeling (2003, p. 32) sugere a implementação de manutenção preventiva e substituição de componentes desgastados. Entretanto, é importante notar que a substituição de componentes reinicia o ciclo de vida destes itens, movendo-os novamente para o período de mortalidade infantil.

É então importante determinar intervalos ideais para a substituição dos componentes. A determinação é realizada através de análises de dados históricos de falhas e desgaste identificando tendências no comportamento dos componentes e aplicar modelos preditivos de fiabilidade, referidos nos seguintes subcapítulos.

Por exemplo substituir um produto muito cedo pode ser dispendioso, aumentando o número de componentes descartados antes de atingirem sua vida útil total, mas substituir tarde demais aumenta o risco de falhas inesperadas, que podem gerar custos maiores em termos de reparação, tempo de inatividade e até riscos à segurança operacional. Um equilíbrio destes intervalos leva a uma maximização da disponibilidade operacional do produto e a uma minimização dos custos de operação.

3.1.2.1 Modelos de taxa de falha constante

Durante o período de vida útil, da curva da banheira, a taxa de falha é considerada constante (distribuição de falha exponencial), e é então possível chegar a novas expressões mais simplificadas da função fiabilidade $R(t)$ e da FDP de falha $f(t)$ (Verma, 2016, p. 41).

$$R(t) = e^{-\int_0^t \lambda(t)dt} = e^{-\lambda \int_0^t dt} = e^{-\lambda t} \tag{15}$$

$$f(t) = \lambda(t) \cdot R(t) = \lambda \cdot R(t) \tag{16}$$

De modo a obter a taxa de falha λ basta saber o tempo médio entre falhas (MTBF) do componente (Ebeling, 2003, p. 42):

$$MTBF = \frac{1}{\lambda} \tag{17}$$

3.1.2.2 Modelos de taxa de falha dependentes do tempo

Nem todos os componentes apresentam uma taxa de falha constante, alguns contém o fator fadiga, que é uma característica dependente do tempo e irá fazer com que a taxa de falha do componente varie.

Esta distribuição de probabilidade com a falha dependente do tempo é chamada de distribuição de Weibull e pode ser utilizada para modelar taxas de falha crescentes e decrescentes. É dada pela seguinte expressão matemática (Ebeling, 2003, p. 58):

$$\lambda = \frac{\beta}{\theta}\left(\frac{t}{\theta}\right)^{\beta-1} \tag{18}$$

Sendo β o parâmetro de forma ($\beta \geq 0$) e θ o parâmetro de escala ($\theta \geq 0$).

Desta forma, a função fiabilidade $R(t)$ ficará (Ebeling, 2003, p. 59):

$$R(t) = e^{-\left(\frac{t}{\theta}\right)^{\beta}} \tag{19}$$

O valor de MTBF é obtido a partir da seguinte expressão (Ebeling, 2003, p. 59):

$$MTBF = \theta\Gamma\left(1 + \frac{1}{\beta}\right) \tag{20}$$

A função $\Gamma(\mathrm{x})$ é denominada por função gama. Os valores desta função encontram-se tabelados em (Ebeling, 2003, p. 473).

3.1.3 Fiabilidade de sistemas

Ao analisar a fiabilidade de um sistema complexo[8], existem duas abordagens de como analisar a fiabilidade do mesmo.

A primeira consiste em tratar o sistema como um todo, assumindo que todos os componentes falham de acordo com a mesma distribuição estatística, como a distribuição exponencial ou de Weibull, independentemente das suas características individuais. Esta abordagem é mais simples e requer menos esforço, pois não exige a análise detalhada dos dados de falha de cada componente.

[8] Sistema composto por múltiplos subsistemas ou componentes interdependentes, onde a fiabilidade de cada elemento afeta diretamente o sistema geral.

Contudo, a sua simplicidade reduz a precisão, uma vez que desconsiderar as diferenças entre os componentes pode resultar em previsões menos fiáveis para o sistema como um todo.

A segunda abordagem, mais detalhada, consiste em analisar individualmente cada componente, identificando os seus modos de falha específicos, o que permite uma avaliação mais precisa da fiabilidade global do sistema.

3.1.3.1 Configuração em série

Os componentes de um sistema podem estar organizados de duas maneiras: em série ou em paralelo.

Na configuração em série, todos os componentes precisam de funcionar corretamente para o sistema operar normalmente. Se qualquer componente falhar, todo o sistema irá falhar. Nesta configuração, todos os componentes são considerados críticos, pois a falha de um impede o funcionamento do sistema. A relação entre os componentes em um sistema com configuração em série encontra-se na figura seguinte:

Figura 7 - Diagrama em bloco para componentes em série
Fonte: adaptado de Ebeling (2003, p. 84)

A fiabilidade para um sistema em série é calculada através da seguinte expressão (Ebeling, 2003, p. 84):

$$R_s(t) = R_1(t) \times R_2(t) \times \ldots \times R_n(t) \tag{21}$$

É importante notar que, como a fiabilidade do sistema resulta do produto de todos os seus componentes, o resultado nunca será maior que o valor mínimo de fiabilidade dos componentes, portanto, é muito importante que todos os componentes apresentem uma fiabilidade elevada.

Se cada componente apresentar uma taxa de falha constante, λ_i, a fiabilidade do sistema será dada por:

$$R_s(t) = \prod_{i=1}^{n} R_i(t) = \prod_{i=1}^{n} \exp(-\lambda_i t) = \exp\left(-\sum_{i=1}^{n} \lambda_i \times t\right) \tag{22}$$

A Figura **8** ilustra um sistema em série com três componentes com uma hora de funcionamento e com taxas de falha constantes, apresentadas na Tabela 2.

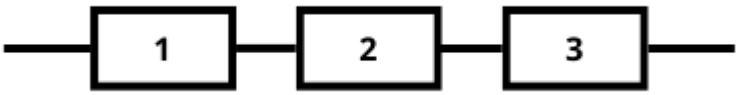


Figura 8 - Diagrama em bloco de um sistema em série
Fonte: (autor, 2025)

Tabela 2 - Taxas de falha de componentes de um sistema em série
Fonte: (autor, 2025)

Componentes	Taxa de Falha
1	$\lambda_1 = 0.002$
2	$\lambda_2 = 0.02$
3	$\lambda_3 = 0.001$

A fiabilidade de cada componente é calculada utilizando a expressão $R(t) = e^{-\int_0^t \lambda(t)dt} = e^{-\lambda \int_0^t dt} = e^{-\lambda t}$ (15). Desta forma, obtém-se os seguintes valores para o tempo $t = 1$ hora:

$$R_1(1) = 0.998,\ \ R_2(1) = 0.980,\ R_3(1) = 0.999$$

A fiabilidade do sistema em série é dada pelo produto das fiabilidades dos componentes:

$$R_s(1) = R_1(1) \cdot R_2(1) \cdot R_3(1) = 0.977$$

Ao analisar o sistema, observa-se que, apesar dos componentes 1 e 3 apresentarem valores elevados de fiabilidade, a fiabilidade total do sistema é significativamente influenciada pelo componente 2, que apresenta uma fiabilidade inferior.

Tal como referido acima, de modo a obter uma fiabilidade elevada num sistema em série é necessário que todos os componentes apresentem fiabilidade elevada. Se o objetivo for melhorar a fiabilidade do sistema, deve-se dar particular atenção à melhoria do desempenho do componente mais crítico, neste caso, o componente 2.

3.1.3.2 Configuração em paralelo

Na configuração em paralelo, o sistema irá falhar quando todos os componentes (em paralelo) falharem. O diagrama em bloco que representa um sistema em paralelo está descrito na figura seguinte:

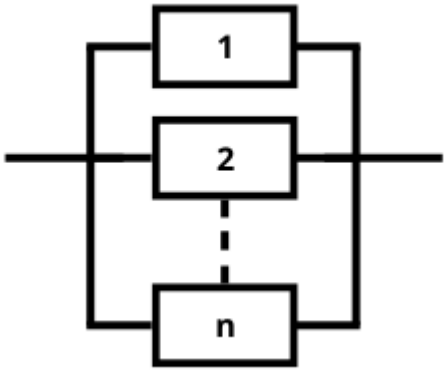


Figura 9 - Diagrama em bloco para componentes em paralelo
Fonte: adaptado de Ebeling (2003, p. 86)

Nesta configuração a fiabilidade do sistema é dada pela seguinte expressão (Ebeling, 2003, p. 86):

$$R_s(t) = 1 - \big(1 - R_1(t)\big) \times \big(1 - R_2(t)\big) \times \ldots \times \big(1 - R_n(t)\big) \tag{23}$$

Generalizando,

$$R_s(t) = 1 - \prod_{i=1}^{n} \big(1 - R_i(t)\big) \tag{24}$$

Através da expressão $R(t) + F(t) = 1(1)$ é possível simplificar ainda mais a expressão obtendo então,

$$R_s(t) = 1 - \prod_{i=1}^{n} \big(F_i(t)\big) \tag{25}$$

Onde $F_i(t)$ representa a probabilidade de falha de cada componente i. Dito isto, $\prod_{i=1}^{n} \big(F_i(t)\big)$ será sempre menor que a probabilidade de falha do componente mais fiável. Portanto, é verdade que a fiabilidade do sistema será sempre maior ou igual ao valor da fiabilidade do componente mais fiável.

Isto é importante entender, pois, ao contrário de um sistema em série, onde todos os componentes devem ter fiabilidades elevadas para que a fiabilidade do sistema seja elevada, num sistema em paralelo, nem todos os componentes têm a obrigatoriedade de apresentar valores de fiabilidade elevados (apesar de na prática ser muito importante todos os componentes apresentarem fiabilidade elevada de modo a maximizar a fiabilidade do sistema).

A Figura 10 ilustra um sistema em paralelo com todos os componentes iguais e tempo de funcionamento igual ao do sistema em série.

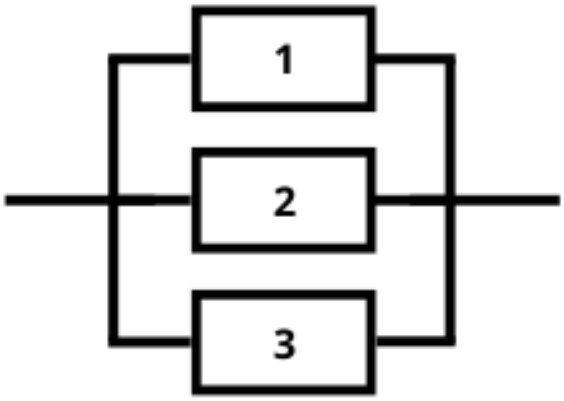


Figura 10 - Diagrama em bloco de um sistema em paralelo
Fonte: (autor, 2025)

Visto que os componentes e o tempo de funcionamento são iguais ao do sistema em série, a fiabilidade dos componentes é dada por:

$$R_1(1) = 0.998,\ \ R_2(1) = 0.980,\ R_3(1) = 0.999$$

A fiabilidade do sistema é dada por:

$$R_s(1) = 1 - (1 - R_1(1)) \cdot (1 - R_2(1)) \cdot (1 - R_3(1)) = 0.9999$$

Ao colocar os componentes em paralelo observa-se um aumento significativo da fiabilidade do sistema comparado ao sistema em série. Isso ocorre porque, neste caso, todos os componentes atuam como redundantes, isto é, mesmo que um ou mais componentes falhem, o sistema pode continuar a operar com os componentes restante, o que aumenta a probabilidade de o sistema permanecer funcional durante mais tempo.

3.2 Gestão da fiabilidade

Em grande medida, a fiabilidade é considerada como um atributo inerente de um sistema, componente ou produto (Ebeling, 2003, p. 145).

Como tal, ao longo de todo o ciclo de vida de uma aeronave, este aspeto deve ser uma consideração essencial para a implementação de um sistema robusto de gestão que abarque desde o projeto inicial até a operação contínua e manutenção.

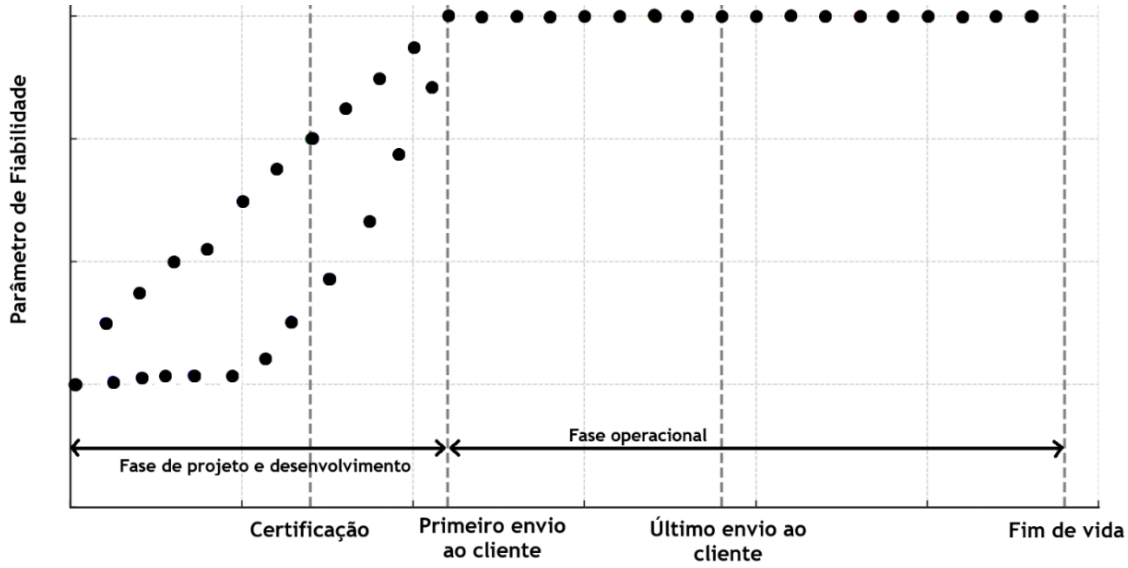


Figura 11 - Crescimento da fiabilidade ao longo de todo o ciclo de vida da aeronave
Fonte: adaptado de NASA (2000, p. 147)

A gestão da fiabilidade envolve a análise da mesma durante a fase de projeto da aeronave através de processos que envolvem definir objetivos de segurança, avaliar métodos de *design*, análise de modos de falha, e implementação de ações corretivas aos métodos implementados com o objetivo de maximizar a fiabilidade até ao fim da fase de projeto da aeronave, conforme ilustrado na Figura 11. Este processo é comummente designado de *design for reliability*, explicado no capítulo 3.3.

Após a entrega ao cliente, a aeronave entra na sua fase operacional. Dentro desta fase, a gestão da fiabilidade torna-se parte de um processo mais amplo conhecido como Integrated Logistics Support (ILS). O ILS na fase operacional vai além da gestão da fiabilidade, englobando um conjunto abrangente de disciplinas e práticas que visam otimizar o desempenho e a disponibilidade da aeronave, enquanto minimiza os custos operacionais e de manutenção. Este processo é explicado com mais detalhe no capítulo 3.4.

3.3 *Design for Reliability* (Fiabilidade de projeto)

O processo de *Design for Reliability (*DfR*)* na indústria aeronáutica é uma abordagem que integra a fiabilidade nos sistemas e componentes de uma aeronave desde as fases iniciais do projeto. Este processo é essencial para assegurar a segurança operacional, a eficiência e a rentabilidade ao longo do ciclo de vida da aeronave. O fluxograma apresentado na figura 12 ilustra as atividades de fiabilidade associadas ao processo de DfR.

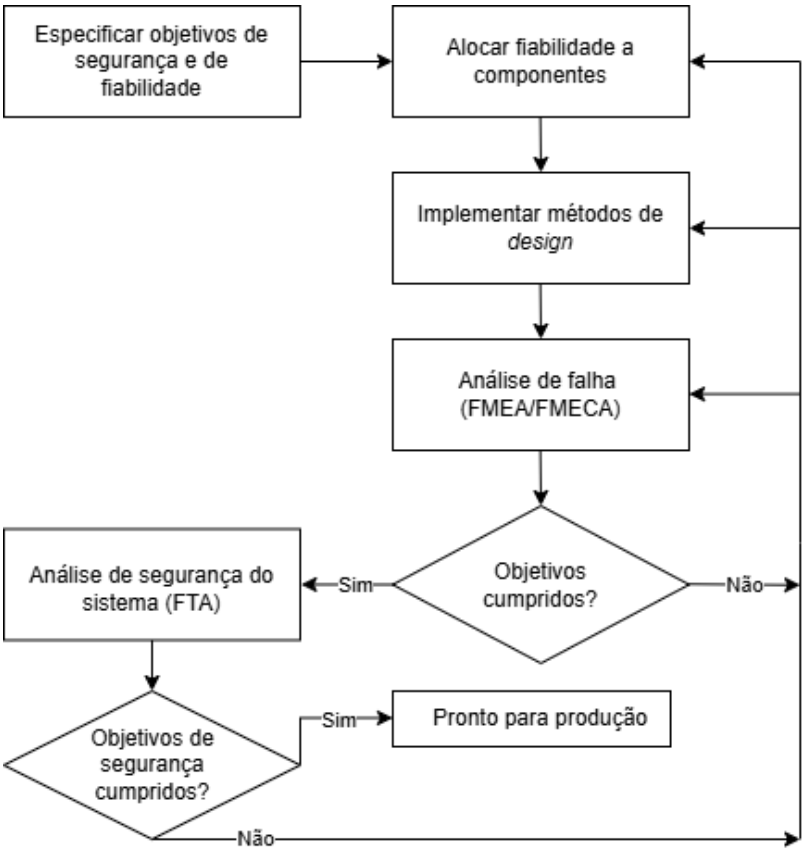


Figura 12 - Processo de Design for Reliability

Fonte: adaptado de Ebeling (2003, p. 146)

A definição dos objetivos de segurança operacional e de fiabilidade técnica constitui um dos passos mais relevantes no processo de *Design for Reliability* (DfR) durante a fase de projeto de uma aeronave.

Este processo assume uma importância fundamental, uma vez que visa assegurar que o sistema cumpre os requisitos regulamentares, responde às expectativas operacionais e garante os níveis de segurança operacionais estabelecidos pelas normas regulamentares da indústria.

Os objetivos de segurança operacional representam os requisitos relacionados com a prevenção de riscos catastróficos e a mitigação de acidentes ou incidentes que possam comprometer a integridade da aeronave, da tripulação e dos passageiros.

Estes objetivos são definidos de acordo com normas internacionais, como os regulamentos da EASA AMC 25.1309. A sua finalidade é assegurar que a probabilidade de ocorrência de falhas críticas é suficientemente baixa para evitar consequências graves, tais como a perda de vidas humanas, a perda da aeronave ou danos severos.

Por outro lado, os objetivos de fiabilidade técnica[9] estão relacionados com a capacidade dos sistemas e componentes da aeronave desempenharem as suas funções conforme projetado, ao longo do ciclo de vida previsto, sem falhas inesperadas.

A finalidade destes objetivos é garantir que os sistemas não cumpram apenas os requisitos de segurança operacional, mas também sejam operacionalmente eficientes e economicamente sustentáveis, minimizando a necessidade de manutenção não programada e os custos associados.

A formulação destes objetivos passa por várias etapas. Inicialmente, é realizada uma análise dos requisitos regulamentares, com base em normas internacionais que definem limites de probabilidade para falhas de sistemas críticos.

Por exemplo, para sistemas cuja falha pode resultar em consequências catastróficas, a probabilidade de falha não deve exceder 10^{-9} (por hora de voo). Este limite, amplamente adotado na indústria aeronáutica, estabelece uma referência para o desenvolvimento de sistemas e componentes. A evolução deste limite encontra-se explicada no capítulo 3.3.1.

O próximo passo do processo envolve a análise funcional e a afetação de objetivos, identificando as funções principais da aeronave e os requisitos operacionais associados. Com base nessa análise, os objetivos de segurança[10] e fiabilidade são distribuídas por cada sistema, considerando a criticidade de cada função e o impacto das falhas na segurança operacional (Ebeling, 2003, p. 151).

Ferramentas como diagramas em blocos ou árvores de falhas são essenciais neste processo. Na análise de árvores de falhas (FTA), o ponto de partida são os requisitos globais de segurança operacional da aeronave, que definem probabilidades máximas aceitáveis para eventos indesejáveis, como falhas catastróficas, cuja probabilidade deve ser inferior a 10^{-9} por hora de voo. A FTA permite desdobrar a probabilidade do evento topo pelos ramos da árvore, distribuindo de forma coerente as metas de fiabilidade pelos sistemas e subsistemas envolvidos.

O foco desta dissertação situa-se dentro deste passo de afetação de fiabilidade, no entanto a árvore de falhas analisada não atribui diretamente as metas de fiabilidade de cima para baixo.

A análise consiste em verificar a fiabilidade de cada evento básico e a partir daí, estudar a contribuição de cada evento para a probabilidade do evento topo. Este método é, na prática, uma análise *bottom-up*, começando pela probabilidade de falha dos componentes individuais e acumulando as probabilidades ao longo da árvore de falhas até chegar ao evento topo.

[9] Estes objetivos centram-se em métricas como MTBF, taxas de falha, disponibilidade operacional, entre outros.

[10] Daqui em diante, o termo segurança ir-se-á sempre referir à segurança operacional no âmbito do termo "safety" evitando confusões com o termo "security".

Embora os objetivos de segurança operacional e de fiabilidade técnica sejam distintos, estes são complementares e interdependentes. A segurança operacional depende diretamente da fiabilidade técnica, uma vez que sistemas mais fiáveis reduzem a probabilidade de falhas críticas que possam comprometer a segurança. Por outro lado, a fiabilidade técnica incorpora a segurança operacional como prioridade, focando-se particularmente em falhas que afetam diretamente a integridade do voo.

Assim, ambos objetivos partilham métricas e metodologias analíticas, mas com perspetivas distintas: enquanto a segurança operacional privilegia a mitigação de riscos catastróficos, a fiabilidade técnica centra-se na sustentabilidade operacional e económica.

A implementação de métodos de design inclui a definição e aplicação de estratégias de redundância, tolerância a falhas e robustez nos sistemas críticos. Durante esta fase são desenvolvidas soluções técnicas e operacionais que atendem às metas de fiabilidade, minimizando o risco de falhas.

Uma vez formalizados os processos de design, é realizada uma análise de modos de falha associada aos métodos implementados. O processo *Failure Mode and Critical Analisys* (FMEA) ou *Failure Mode, Effect, and Criticality* Analisys (FMECA) é um processo iterativo[11] que influencia as escolhas de design ao identificar modos de falha, classificar a severidade dessas condições de falha, a probabilidade de ocorrência das mesmas de forma qualitativa e quantitativa e a contínua implementação de medidas corretivas. Uma explicação mais detalhada deste processo encontra-se no capítulo 3.3.2.

O último processo consiste em realizar uma análise de árvore de falhas que tem como objetivo fornecer uma compreensão clara e detalhada de como diferentes falhas e condições combinadas podem levar a eventos indesejados ou catastróficos. Este processo encontra-se detalhado no capítulo 3.3.4.

3.3.1 Da probabilidade de falha 10^{-9}

A evolução e definição da probabilidade de falha catastrófica de 10^{-9} está diretamente relacionado ao aumento da complexidade das aeronaves e à necessidade de padrões de segurança mais rigorosos.

Durante o período pós-Segunda Guerra Mundial as regras e normas técnicas na indústria da aviação eram predominantemente prescritivas e baseadas em dados empíricos. Contudo,

[11] Este processo é considerado iterativo porque à medida que mais componentes e subsistemas são definidos, novas causas e modos de falha são descobertos e as melhorias e ajustes definidos na iteração anterior devem ser aprimoradas continuamente.

com o surgimento de aeronaves mais modernas e sistemas mais interdependentes, tornou-se evidente que esses métodos tradicionais eram insuficientes para lidar com a complexidade crescente e os riscos associados (Vincent, 2021).

Era necessário um nível mais profundo de análise que considerasse não apenas os riscos individuais dos sistemas, mas também as interações entre eles. Essa transição resultou numa abordagem regulatória baseada no desempenho, onde a avaliação de riscos passou a incluir tanto as falhas de sistemas individuais quanto as consequências das interações entre sistemas.

Um dos marcos dessa evolução foi a introdução do princípio de correlação inversa entre a probabilidade de falha e a gravidade do seu impacto.

"Esta evolução conduziu a um princípio fundamental: deve existir uma correlação inversa entre a probabilidade de ocorrência de uma falha e a gravidade do seu impacto na aeronave e/ou seus ocupantes" (EASA AMC 25.1309, 2020, pp. 2-F-45).

Na análise de viabilidade de um projeto reconheceu-se a importância de estabelecer metas probabilísticas baseadas em dados históricos. Esses dados apontavam para uma probabilidade de acidentes graves de aproximadamente um por um milhão de horas de voo. Considerou-se que, para novos projetos de aeronaves, a probabilidade de acidentes graves causados por falhas de sistemas não devesse exceder este valor.

Definiu-se então como objetivo que a probabilidade de acidentes graves por falhas de sistemas não excedesse um em dez milhões de horas de voo, ou 1×10^{-7} por hora de voo (EASA AMC 25.1309, 2020, pp. 2-F-45).

No entanto, surgiu uma dificuldade: não seria possível confirmar o cumprimento deste objetivo sem uma análise numérica coletiva de todos os sistemas da aeronave. De forma a contornar este obstáculo, assumiu-se arbitrariamente a existência de dez falhas potenciais em dez subsistemas da aeronave, portanto, cem potenciais condições de falha numa aeronave que poderiam ser catastróficas (Pike, 2010).

Sob essa premissa, se a probabilidade de falha de um subsistema é dada por x, a sua probabilidade de cumprir a missão será dada por $(1 - x)$. De acordo com Pike (2010), ao atribuir um valor de 10^{-9} à probabilidade de falha de cada subsistema, portanto $x = 10^{-9}$, a probabilidade de cumprimento da missão da aeronave como um todo, considerando as 100 condições de falha, é calculada como $(1 - 10^{-9})^{100}$.

O objetivo é garantir que a probabilidade de falha catastrófica da aeronave permaneça abaixo do limite de 10^{-7} por hora de voo. Para isso, a probabilidade de falha dos subsistemas

individuais deve ser suficientemente baixa, de modo que a probabilidade combinada de falha catastrófica não exceda este valor.

Uma abordagem comum é limitar a probabilidade de falha de cada subsistema crítico a um máximo de 10^{-9} por hora de voo, assumindo uma arquitetura com redundâncias adequadas e decomposição de risco apropriada. Isso assegura que a soma das contribuições individuais dos subsistemas não ultrapasse o limite estabelecido para a aeronave como um todo.

Para verificar, basta utilizar a equação $(1-10^{-9})^{100}-(1-10^{-7}) \approx 7.77 \times 10^{-15} > 0$. Isto significa que a probabilidade de não falha total é suficientemente alta, garantindo que o limite de segurança (10^{-7}) é respeitado.

Ao considerar, por exemplo, 10^{-8} como probabilidade de falha de cada subsistema, temos que $(1-10^{-8})^{100}-(1-10^{-7}) \approx -8.99 \times 10^{-7} < 0$. Isto significa que a probabilidade de não falha total não é suficientemente alta, e a probabilidade de falha catastrófica da aeronave ultrapassa o limite aceitável de 10^{-7}. Portanto, valores como 10^{-8} não cumprem os critérios de segurança.

Estabeleceu-se assim que o limite superior para a probabilidade média por hora de voo de condições de falhas catastróficas seria de 10^{-9}, definindo um valor probabilístico aproximado para o termo "extremamente improvável". Condições de falha com efeitos menos severos poderiam ter uma probabilidade de ocorrência relativamente maior.

Este valor serve como referência para o design e análise de sistemas críticos e, no âmbito deste trabalho, é definido como o objetivo de fiabilidade para a probabilidade de falha catastrófica da aeronave.

Através da normativa AMC 25.1309 da EASA, (2020, pp. 2-F-50), os objetivos de segurança associados a condições de falha catastrófica devem ser satisfeitos ao demonstrar que:

- Nenhuma falha isolada resultará numa condição de falha catastrófica; isto é, o sistema deve ser projetado de tal forma que nenhum componente ou subsistema individual, ao falhar, possa causar por si só uma condição de falha catastrófica. Isto é considerado como *fail-safe design* e pode ser alcançado através de redundâncias ou sistemas de *backup*.

- Cada falha catastrófica é extremamente improvável;

- Considerando que uma única falha latente[12] ocorreu num determinado voo, cada condição de falha catastrófica, resultante de duas falhas, sendo qualquer uma delas latente por mais de um voo, é remota; isto é, mesmo se uma falha latente já estiver presente, a probabilidade de uma segunda falha (que também poderia ser latente) ocorrer e resultar numa condição catastrófica deve ser remota.

3.3.2 Análise de modos de falha

A análise de modos de falha e efeitos (FMEA – *Failure Modes and Effects Analysis*) sucede o processo de alocação de fiabilidade. Este método permite identificar, avaliar e mitigar riscos associados a potenciais falhas em sistemas críticos, garantindo que os requisitos de segurança regulamentares, como os estabelecidos pela EASA AMC 25.1309, sejam cumpridos.

3.3.2.1 Classificação da severidade e probabilidade de ocorrências de falhas

A EASA, a partir do documento AMC 25.1309, fornece uma análise qualitativa e quantitativa das classificações das condições de falha e das suas probabilidades de ocorrência. As análises quantitativas concentram-se em dados mensuráveis e numéricos permitindo decisões baseadas em informações concretas e objetivas. Já a análise qualitativa aborda aspetos mais subjetivos e difíceis de quantificar proporcionando *insights* valiosos sobre experiências e perceções cruciais para o tema em questão.

A severidade refere-se ao grau de impacto ou consequência que uma falha pode ter sobre a operação ou segurança de um sistema. No contexto aeronáutico, este grau é medido de forma qualitativa, categorizando as condições de falha com base nas definições fornecidas pela EASA no documento AMC 25.1309. As condições de falha são classificadas em cinco categorias dependendo do efeito que têm na tripulação, nos passageiros e na aeronave.

De acordo com a EASA (2020, pp. 2-F-47) a análise qualitativa das condições de falha é realizada da seguinte forma:

1) Sem efeito na segurança: Não apresentam impacto na segurança. Como por exemplo: falha no sistema de entretenimento, avaria da luz de leitura.

2) Menor: Apresentam redução mínima da segurança, dentro das capacidades da tripulação. Como por exemplo: pequeno aumento na carga de trabalho da tripulação, ligeiro desconforto dos passageiros.

[12] Defeito ou condição de erro que existe num sistema, mas que não é imediatamente visível ou detetável durante a operação normal.

3) Maior: Reduzem significativamente as capacidades ou margens de segurança. Como por exemplo: aumento significativo na carga de trabalho da tripulação, desconforto físico dos ocupantes.

4) Perigosa: Comprometem severamente as capacidades da aeronave ou tripulação. Como por exemplo: grande redução nas capacidades funcionais, possibilidade de ferimentos graves em alguns ocupantes.

5) Catastrófica: Resultam em múltiplas fatalidades e perda da aeronave. Como por exemplo: falha estrutural crítica, perda total de controlo da aeronave.

No que toca à probabilidade de ocorrência das falhas, a EASA também fornece uma análise quantitativa e qualitativa da probabilidade de ocorrências das falhas. São expressos em termos de intervalos aceitáveis para a probabilidade média por hora de voo e estão descritos na Tabela 3.

Tabela 3 - Análise qualitativa e quantitativa da probabilidade de ocorrência de falha
Fonte: adaptado de EASA (2020, pp. 2-F-48)

	Probabilidade de Ocorrência			
Definição Qualitativa	Extremamente improvável	Extremamente remota	Remota	Provável
Descrição	Não esperada em toda a frota	Muito rara, mas pode ocorrer na frota	Rara por aeronave, mas possível na frota	Ocorre pelo menos uma vez por aeronave
Definição Quantitativa	$\leq 10^{-9}$ por hora de voo	10^{-7} a 10^{-9} por hora de voo	10^{-5} a 10^{-7} por hora de voo	$> 10^{-5}$ por hora de voo

3.3.2.2 Classificação da tolerância do risco

Seguindo os princípios estabelecidos de classificação de severidade e de probabilidade de ocorrência de uma falha, "uma relação inversa lógica e aceitável deve existir entre a probabilidade média por hora de voo e a severidade dos efeitos das condições de falha." (EASA AMC 25.1309, pp. 2-F-48). Esta relação assegura que as falhas mais severas devem apresentar probabilidades de ocorrência significativamente mais baixas para garantir um nível aceitável de segurança.

Esta relação é tal que:

- Condições de falha **sem impacto na segurança** não têm restrições específicas quanto à sua probabilidade de ocorrência.

- Condições de falha **menores** podem ter uma probabilidade considerada **provável**.

- Condições de falha **maiores** não devem ter uma probabilidade superior a **remota**.

- Condições de falha **perigosas** não devem ter uma probabilidade superior a **extremamente remota**.

- Condições de falha **catastróficas** devem ser **extremamente improváveis**.

É possível juntar estas combinações de modo a obter o seguinte gráfico:

Figura 13 - Relação entre probabilidade e severidade das condições de falha.
Fonte: adaptado de **(EASA AMC 25.1309, 2020, pp. 2-F-49)**

O gráfico apresentado descreve a relação inversa entre a severidade e a probabilidade de ocorrência das condições de falha. Nos eixos do gráfico, o eixo horizontal representa a severidade da condição de falha, ou seja, o impacto que essa falha pode ter no sistema, já o eixo vertical representa a probabilidade de ocorrência dessa condição de falha.

A partir deste gráfico é possível obter uma ideia clara do risco associado a cada condição de falha dos sistemas da aeronave, porque, apesar do gráfico descrever diretamente as

condições de falha, o risco de cada uma delas é avaliado dentro do gráfico, a partir da interseção (linha a azul) que é criada a partir das relações inversas de probabilidade/severidade impostas pela EASA, referidas acima. A intersecção permite determinar se o risco associado é aceitável, inaceitável ou tolerável.

A linha formada representa os riscos toleráveis, que são aqueles considerados permissíveis apenas por um período limitado. Contudo, esses riscos exigem um plano rigoroso para redução do risco, monitorização contínua e avaliações regulares para garantir que o risco está a ser mitigado de forma eficaz.

Os riscos inaceitáveis situam-se acima da linha tolerável no gráfico de severidade/probabilidade. Estes representam condições que colocam um nível de perigo inaceitável para a segurança da operação, podendo levar a consequências graves ou catastróficas, mesmo que a probabilidade de ocorrência seja extremamente baixa.

Em termos práticos, estes riscos requerem ação imediata para reduzir o seu impacto até, pelo menos, um nível tolerável. Isso pode envolver o *redesign* de componentes, a implementação de medidas de redundância, ou mudanças operacionais. Operações ou atividades associadas a riscos inaceitáveis não devem ser permitidas até que sejam implementadas medidas corretivas suficientes, garantindo que o risco seja mitigado de forma a garantir a segurança da operação.

Por outro lado, os riscos aceitáveis encontram-se abaixo da linha tolerável. Estes são considerados suficientemente seguros para permitir operações normais sem a necessidade de medidas de mitigação adicionais. Contudo, é importante implementar um processo de monitorização regular para verificar se o nível de risco não aumenta devido a alterações nos sistemas, degradação de componentes, ou mudanças operacionais.

Entre os extremos de riscos aceitáveis e inaceitáveis encontram-se os riscos toleráveis, que estão localizados diretamente na linha divisória. Estes riscos requerem uma abordagem diferenciada: são permissíveis apenas temporariamente, com a condição de que seja estabelecido um plano de mitigação detalhado e rigoroso, que inclua prazos definidos para redução do risco. Além disso, os riscos toleráveis exigem monitorização contínua, verificações regulares e revisão periódica das estratégias de mitigação implementadas para garantir que o risco permanece sob controlo e dentro dos limites aceitáveis.

Por exemplo, um risco inaceitável pode ser considerado como a perda de ambos motores da aeronave (severidade catastrófica) quando estes apresentem uma probabilidade estimada de falha maior do que “extremamente improvável” ($< 10^{-9}$ por hora de voo). Uma solução para

este caso poderá ser projetar os motores com sistemas redundantes e realizar manutenções programadas com maior frequência para prevenir falhas.

A falha de um sistema de navegação secundário como o VOR durante o voo (severidade perigosa) com uma probabilidade de ocorrência extremamente remota ($< 10^{-7}$ por hora de voo) é considerado como um risco tolerável. Algumas medidas de mitigação para este risco pode ser a colocação de recetores VOR redundantes na aeronave ou treinar os pilotos a operarem com navegação alternativa (como por exemplo pilotagem[13] ou *dead reckoning*[14]).

Embora a análise FMECA seja uma ferramenta detalhada para identificar modos de falha e avaliar a sua criticidade ao nível de componentes e subsistemas, a abordagem adotada neste trabalho foca-se numa análise sistémica e preliminar.

Assim, os eventos básicos na árvore de falha representam condições críticas de falha de sistemas, com base na identificação de apenas os modos de falha mais relevantes. Esta escolha é consistente com a fase de projeto preliminar da aeronave LUS-222, onde os detalhes completos de sistemas e subsistemas ainda não estão disponíveis.

O objetivo é fornecer uma ferramenta flexível que permita realizar análises de fiabilidade numa etapa inicial, sem a necessidade de realizar análises mais detalhadas, como a FMECA, que são mais adequadas em fases posteriores do design.

3.3.3 Análise de segurança de sistemas

A fiabilidade e a segurança de produtos estão intimamente relacionadas. Enquanto a fiabilidade se foca na capacidade de um sistema ou componente desempenhar a sua função de forma consistente ao longo do tempo, a segurança é definida como a prevenção de condições que possam causar danos graves, como perdas de vida, prejuízos significativos a equipamentos ou ao ambiente circundante.

No contexto da análise de segurança de sistemas, o foco recai especialmente sobre falhas que possam gerar riscos significativos para a operação segura da aeronave. O objetivo deste processo é identificar como é que as falhas mais críticas podem ocorrer (avaliar a arquitetura do sistema), estimar a sua probabilidade e implementar ações corretivas para mitigar os riscos associados (Ebeling, 2003, p. 95).

[13] Método de navegação visual baseado em referências terrestres para manter o rumo e posicionar-se.

[14] Cálculo da posição da aeronave com base na direção, velocidade, tempo e posição inicial, sem usar referências externas.

A FTA e outras ferramentas analíticas desempenham um papel fundamental neste processo, permitindo mapear combinações de falhas que podem levar a eventos perigosos e determinar estratégias eficazes para reduzir os riscos a níveis aceitáveis.

Assim, a análise de segurança de sistemas é uma extensão da análise de fiabilidade, garantindo não apenas a continuidade do desempenho do sistema, mas também a segurança operacional em todas as condições previstas.

3.3.4 Árvore de análise de falhas

Como referido, a árvore de falhas é uma ferramenta comummente utilizada dentro de todo o processo de *design for reliability* e é a base de estudo deste trabalho.

A FTA é um método que usa lógica booleana[15] para representar as relações entre falhas de componentes e a falha do sistema como um todo. Esta identifica as combinações de eventos que podem levar a uma falha do sistema e calcular a probabilidade dessa falha ocorrer.

O processo de construção da FTA começa com a definição do evento topo e desenvolve-se hierarquicamente, representando as combinações lógicas de eventos que podem resultar nesse evento principal. A árvore de falhas é composta por portas lógicas, que descrevem como as entradas se combinam para originar uma saída, permitindo uma análise detalhada das interdependências entre as falhas.

"Importa salientar que uma árvore de falhas não é, por si só, um modelo quantitativo. É um modelo qualitativo que pode ser avaliado quantitativamente e frequentemente o é. Este aspeto qualitativo, naturalmente, é verdadeiro para praticamente todas as variedades de modelos de sistemas. O facto de uma árvore de falhas ser um modelo particularmente conveniente de quantificar não altera a natureza qualitativa do próprio modelo" (Fault Tree Handbook, 1981, p. 34)

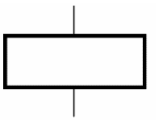
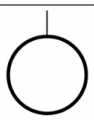
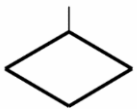
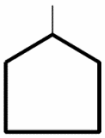
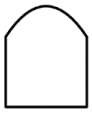
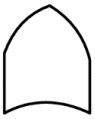
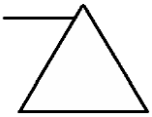
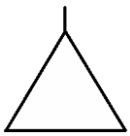
Apesar da árvore de falhas não cobrir todas as possíveis falhas de um sistema, esta foca-se nas falhas mais credíveis e relevantes que contribuem para o evento topo, que se encaixa precisamente com o tema deste trabalho, apenas serão analisadas as condições de falha que são consideradas críticas e cuja falha levam diretamente à falha do sistema.

A simbologia utilizada na análise qualitativa da árvore está descrita na Tabela 4.

[15] Ramo da matemática que lida com operações sobre valores lógicos.

Tabela 4 - Simbologia utilizada na FTA

Fonte: adaptado de *Fault Tree Handbook* (1981, p. 36)

Nome do símbolo	Representação gráfica	Descrição
Evento topo ou mediador		Evento de falha resultante da combinação lógica dos eventos de entrada, que estão a operar através da porta lógica.
Evento básico		Evento independente, usado apenas como entrada de uma porta lógica.
Evento não desenvolvido		Evento não desenvolvido devido a falta de informação ou devido a baixo nível de risco.
Evento casa		Evento esperado que seguramente irá ocorrer durante a função normal do sistema.
Porta AND		A falha na saída ocorre se em todas as entradas ocorrer falha.
Porta OR		A falha na saída ocorre se pelo menos numa das entradas ocorrer falha.
Transfer OUT		Indica que essa porção da árvore deve ser anexada no correspondente TRANSFER IN.
Transfer IN		Indica que a árvore é desenvolvida mais adiante na ocorrência do correspondente TRANSFER OUT (por exemplo, em outra página).

Para ilustrar o processo de análise de uma árvore de falhas, considera-se a estrutura genérica da figura 14, utilizada como referência ao longo do estudo. Esta árvore não representa um sistema específico, mas permite demonstrar a aplicação dos conceitos fundamentais, incluindo a identificação dos *minimal cutsets* e a análise dos parâmetros de importância. Através desta abordagem, é possível compreender a metodologia utilizada na avaliação da fiabilidade e do impacto de diferentes eventos na ocorrência do evento topo.

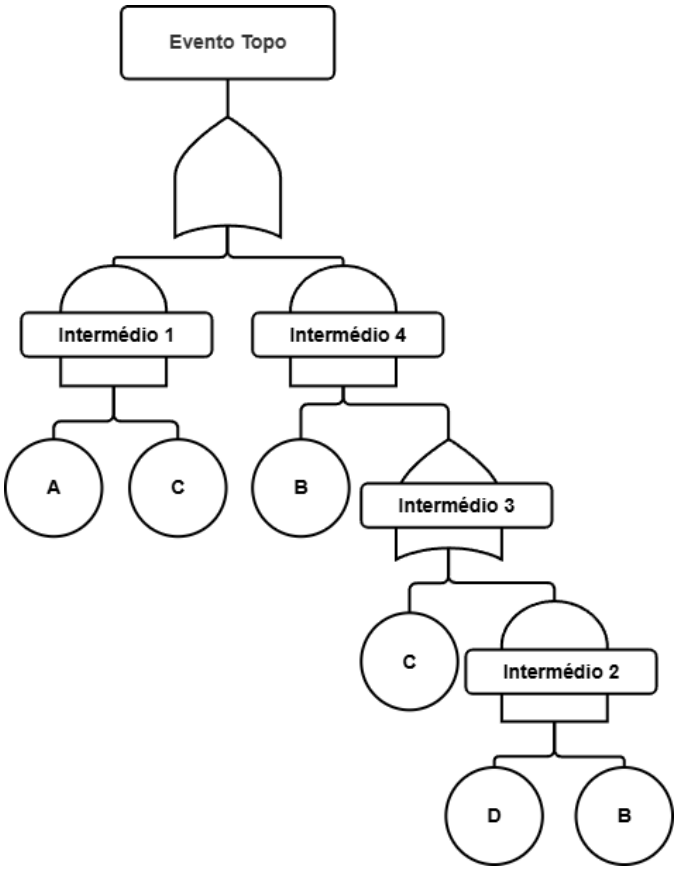


Figura 14 - Exemplo de uma árvore de falhas

Fonte: (autor, 2025)

A análise de uma árvore de falhas começa com a identificação de todas as combinações possíveis de eventos básicos que podem levar à ocorrência do evento topo. A árvore de falhas é estruturada hierarquicamente, utilizando portas lógicas como AND (união) e OR (interseção) para modelar as interações entre os eventos básicos que contribuem para o evento topo.

A modelação destas interações é realizada de forma algébrica. “Uma álgebra baseada nas operações de união, intersecção, entre outras é denominada álgebra booleana. Ao utilizar as operações básicas de união, intersecção e complementação, a álgebra booleana permite-nos expressar eventos em termos de outros eventos básicos.” (Fault Tree Handbook, 1981, p. 71).

A álgebra booleana lida com operações de eventos que são representadas por vários símbolos. No entanto, a simbologia de teoria dos conjuntos não é uniforme, isto é, difere entre as diferentes disciplinas da matemática, lógica e engenharia, como mostrado na seguinte tabela.

Tabela 5 - Simbologia de operações lógicas em diversas áreas

Fonte: Adaptado de (Fault Tree Handbook, 1981, p. 71)

Operação	Probabilidade	Matemática	Lógica	Engenharia
União de A e B	A ou B	$A \cup B$	$A \vee B$	$A + B$
Interseção de A e B	A e B	$A \cap B$	$A \wedge B$	$A \cdot B$ ou AB

No contexto deste trabalho, a simbologia da área da engenharia é a escolhida para representar as combinações de todos os eventos básicos.
Ao traçar todos os eventos da árvore acima obtém se a seguinte expressão:

$$(A \cdot C) + \Big(B \cdot \big(C + (D \cdot B)\big)\Big) \tag{26}$$

Esta expressão é considerada como a expressão dos *cut sets* da árvore de falhas. Um *cut set* é um grupo de eventos que, quando ocorrem simultaneamente, causam a ocorrência do evento topo (Fault Tree Analysis (FTA) Guidance Material, 2005, p. 30).

O próximo passo após traçar todos os caminhos possíveis da árvore é simplificar a expressão. Visto que a expressão é expressa de forma lógica, são utilizadas propriedades da álgebra booleana que ajudam na simplificação da expressão.

A álgebra booleana apresenta diversas propriedades, mas apenas algumas são tidas em conta porque se encaixam com o objetivo do trabalho. Estas são:

- Idempotência: Somar ou multiplicar um termo com ele mesmo não altera o resultado:
$$A + A = A \text{ e } A \cdot A = A$$

- Propriedade comutativa: A ordem dos termos não altera o resultado:
$$A + B = B + A \text{ e } A \cdot B = B \cdot A$$

- Propriedade associativa: A forma como os termos são agrupados não altera o resultado:
$$(A + B) + C = A + (B + C) \text{ e } (A \cdot B) \cdot C = A \cdot (B \cdot C)$$

- Propriedade distributiva: A soma pode ser distribuída sobre a multiplicação e vice-versa:
$$A \cdot (B + C) = (A \cdot B) + (A \cdot C)$$
$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

- Propriedade de absorção: Um termo pode absorver outro relacionado:

$$A + (A \cdot B) = A$$
$$A \cdot (A + B) = A$$

A partir destas propriedades, ao simplificar a expressão $(A \cdot C) + \Big(B \cdot \big(C + (D \cdot B)\big)\Big)$ (26) obtém-se:

$$(A \cdot C) + \Big(B \cdot \big(C + (D \cdot B)\big)\Big) = AC + BC + B \cdot DB$$

$$AC + BC + BD \tag{27}$$

Após a simplificação da expressão, é possível retirar todos os *minimal cut sets* da árvore de falhas. Enquanto um *cut set* é um grupo de eventos que, quando ocorrem simultaneamente, causam a ocorrência do evento topo, um *minimal cut set* é o menor grupo de eventos no qual todos devem ocorrer para que o evento topo aconteça. Se qualquer um dos eventos de um minimal cut set não ocorrer, isso impede a ocorrência do evento topo (Fault Tree Analysis (FTA) Guidance Material, 2005, p. 30).

No caso do exemplo da árvore de falhas os *cut sets* da árvore são os mesmo que os *minimal cut sets*, estes são: {A, C}, {B, C} e {B, D}. No entanto existem casos onde estes não serão sempre iguais, por exemplo, caso a expressão lógica de uma árvore seja dada por $(A \cdot B) + (A \cdot B \cdot C)$, os *cut sets* são {A, B}, {A, B, C} no entanto, o *minimal cut set* é apenas {A, B} isto porque a falha conjunta dos eventos A e B provoca automaticamente a falha total do sistema.

Portanto, a expressão inicial que é obtida a partir da análise dos caminhos de falha da árvore é a expressão que envolve todos os *cut sets* do evento topo. Ao simplificar esta expressão, é obtida a expressão que contém apenas os *minimal cut sets* do sistema. No exemplo acima, a expressão $(A \cdot B) + (A \cdot B \cdot C)$ é simplificada a partir da propriedade da absorção de forma a obter $(A \cdot B)$ que é precisamente o *minimal cut set* do sistema.

Toda a análise da árvore de falhas realizada até este ponto pertence à análise qualitativa, onde ajuda na compreensão das vulnerabilidades estruturais do sistema e os caminhos críticos para a falha.

A análise quantitativa utiliza os *minimal cut sets* para calcular a probabilidade de falha do sistema, integrando informações sobre as probabilidades de falha individuais dos eventos básicos.

Assumindo que os minimal cut sets são independentes entre si, isto é, a ocorrência de um *minimal cut set* não influencia a ocorrência de outro, ou seja, não existem eventos básicos

comuns entre diferentes *minimal cut sets*, a probabilidade de ocorrência do evento topo $Q_0(t)$, pode ser calculada utilizando a seguinte expressão:

$$Q_0(t) = 1 - \prod_{j=1}^{k} \left[1 - Q_j(t)\right] \tag{28}$$

Onde:

- k representa o número de *minimal cutsets* identificados;
- $Q_j(t)$ é a probabilidade de falha associada ao minimal cutset C_j, que é obtida multiplicando as probabilidades de falha dos eventos básicos que o compõem (Fault Tree Handbook, 1981, p. 181):

$$Q_j(t) = \prod_{i \in C_j} q_i(t)$$

onde $q_i(t)$ é a probabilidade de falha do evento básico i no tempo t.

Contudo, na prática, os *minimal cut sets* podem não ser completamente independentes devido à sobreposição de eventos básicos entre diferentes conjuntos. Este fenómeno é conhecido como dependência positiva, e pode levar a uma dupla contagem dos eventos básicos, o que aumenta artificialmente a probabilidade de falha do evento topo.

Uma dependência positiva indica que dois ou mais minimal cut sets partilham um ou mais eventos básicos. E o que acontece com esses eventos partilhados afeta diretamente a probabilidade de ocorrência dos minimal cut sets que os contêm. No exemplo da árvore de falhas acima, os *minimal cut sets* identificados são C_1 ={A, C}, C_2 ={B, C} e C_3{B, D}. Os eventos B e C estão contidos em dois *minimal cut sets* cada um.

Isto significa que a falha de qualquer um destes eventos aumenta simultaneamente a probabilidade dos MCS afetos ao evento básico.

Por exemplo, se $P(A) = P(B) = P(C) = P(D) = 0.1$, então $C_1 = A \cdot C = 0.1 \cdot 0.1 = 0.01$, $C_2 = 0.01$ e $C_3 = 0.01$. No caso do evento básico C falhar (considerando $P(C) = 1$) então $C_1 = 0.1 \cdot 1 = 0.1$, $C_2 = 0.1$ e $C_3 = 0.01$. A probabilidade de C_1 e C_2 aumentou visto que C falhou. Assim, o evento C causa um impacto simultâneo em ambos os minimal cut sets, criando a dependência positiva entre eles (o mesmo fenómeno ocorre com o evento B).

Ao utilizar a expressão $Q_0(t) = 1 - \prod_{j=1}^{k}\left[1 - Q_j(t)\right]$ (28), como os *minimal cut sets* não são independentes, e como esta fórmula considera todos os MCS como eventos completamente distintos, isto irá levar a uma dupla contagem dos eventos partilhados, ou seja, a fórmula subestima a sobreposição, levando a uma probabilidade do evento topo maior do que a real.

Ainda que o valor resultante da expressão não ser igual ao valor real[16] da probabilidade do evento topo, esta fórmula ainda consegue fornecer uma estimativa conservadora e útil para casos de minimal cut sets não independentes. A partir desta expressão, denominada de "*upper bound approximation*", é realizada o resto da análise quantitativa da árvore (Lundteigen & Rausand, p. 14):

$$Q_0(t) \leq 1 - \prod_{j=1}^{k} [1 - Q_j(t)] \tag{29}$$

No caso da árvore de falhas acima, visto que a mesma apresenta dependências entre minimal cut sets, a expressão da probabilidade de falha do evento topo é dada por:

$$Q_0(t) \leq 1 - (1 - A \cdot C) \cdot (1 - B \cdot C) \cdot (1 - B \cdot D)$$

Além de calcular a probabilidade do evento topo, a análise quantitativa abrange a análise de importância e a análise de sensibilidade.

A análise de importância tem como objetivo avaliar o papel de um evento básico na ocorrência do evento topo. Através de diferentes fatores de importância, como o fator marginal, o fator de criticidade ou os fatores de aumento ou redução de risco, esta análise permite identificar os principais contribuintes para a probabilidade do evento topo.

Assim, torna-se possível determinar quais elementos do sistema devem ser melhorados para aumentar a segurança de forma eficiente e orientada (Fault Tree Analysis (FTA) Guidance Material, 2005, p. 24).

A análise de sensibilidade observa como as variações nos parâmetros probabilísticos dos eventos básicos influenciam a probabilidade do evento topo. Este tipo de análise permite avaliar o impacto de alterações nos intervalos de manutenção, modificações de design ou melhorias na fiabilidade dos componentes.

Os resultados obtidos fornecem uma avaliação detalhada da contribuição de cada parâmetro para a métrica de segurança do evento topo, ajudando a priorizar intervenções específicas no sistema (Fault Tree Analysis (FTA) Guidance Material, 2005, p. 25).

3.3.4.1 Análise de importância

Como referido acima, a análise de importância consiste em calcular fatores de importância com o objetivo de responder à pergunta de: "Quais são os contribuidores mais importantes para a medida de segurança associada ao evento topo?" (Fault Tree Analysis (FTA) Guidance Material, 2005, p. 77).

[16] O valor real é calculado através das expressões de fiabilidade explicadas no capítulo 3.1.3

Os parâmetros de importância considerados na análise do trabalho são os seguintes:

Fator de importância de *birnbaum*: O fator de importância de birnbaum (ou fator de importância marginal) mede o impacto de um evento básico na probabilidade de falha do sistema, assumindo que todos os outros eventos permanecem constantes.

Avalia quão sensível a probabilidade de falha do sistema é em relação à falha de um evento básico específico. O cálculo deste parâmetro é dado por (Norwegian University of Science and Technology, p. 12):

$$I_B(E_i) = \frac{\partial Q_0(t)}{\delta q_i(t)} \tag{30}$$

Onde:

- $Q_0(t)$ é a probabilidade de falha do sistema;
- $q_i(t)$ é a probabilidade de falha do evento básico E_i.

O parâmetro de *birnbaum*, embora útil, não considera diretamente a probabilidade de ocorrência do evento básico. Isto pode levar à atribuição de altas medidas de importância a eventos que têm muita pouca probabilidade de ocorrer e podem ser muito difíceis de melhorar. Portanto, para se concentrar em eventos que não são apenas críticos para o evento topo, mas também têm maior probabilidade de ocorrer, uma medida de importância modificada, conhecida como o parâmetro de criticidade, é utilizado para determinar o evento básico a ser melhorado (Windchill Risk and Reliability).

O **índice de criticidade** mede a contribuição relativa de um evento básico para a probabilidade de falha do sistema, ponderando essa contribuição pela probabilidade real de falha do próprio evento. Em termos matemáticos, este parâmetro incorpora a probabilidade de ocorrência de cada evento básico, ajustada com base no valor de *birnbaum* previamente calculado, proporcionando uma visão mais realista da importância de cada componente para o desempenho global do sistema (Norwegian University of Science and Technology, p. 13)

$$I_C(E_i) = \frac{q_i(t) \cdot I_B(E_i)}{Q_0(t)} \tag{31}$$

O **índice de importância Fussell-Vesely** (FV) mede a contribuição relativa de um evento básico para a probabilidade de falha do sistema, com base no minimal cut *sets* que contêm esse evento. Em termos matemáticos, o parâmetro envolve a soma da probabilidade de falha de todos os minimal cut sets que contem o evento básico i e depois disso, a divisão pela

probabilidade de falha do evento topo, calculada através do valor real (Norwegian University of Science and Technology, p. 14):

$$FV = \frac{\sum_{j}^{n} Q_j^i}{Q_0(t)} \tag{32}$$

Onde:

- $\sum_{j}^{n} Q_j^i$ é a soma das probabilidades de falha do minimal cut *sets* que incluem o evento básico i;

O ***risk reduction worth*** (RRW) é um parâmetro que mede melhoria na fiabilidade do sistema se o evento básico em estudo for infalível, isto é, com a sua probabilidade de falha $q_i(\boldsymbol{t}) = 0$. Este índice avalia o impacto máximo de eliminar a falha de um componente e é dado pela seguinte expressão (Contini & Matuzas, 2010, p. 15):

$$RRW_i(t) = \frac{Q_0(t)}{Q_0(t)|_{q_i=0}} \tag{33}$$

Onde:

- $Q_0(t)|_{q_i=0}$ é a probabilidade de falha do sistema assumindo que o evento básico i é infalível.

Ao tornar qualquer evento básico infalível nunca aumentaria a probabilidade de falha do sistema, portanto $1 \leq RRW_i(t) < \infty$.

Um valor de RRW mais elevado indica que o evento básico é mais importante, pois torná-lo perfeitamente fiável teria um maior impacto na redução da probabilidade do evento de topo.

O ***risk achievement worth*** (RAW) é outro parâmetro muito parecido ao RRW, no entanto este analisa o aumento de risco do evento topo se o evento básico em análise estiver em estado de falha, ou seja, $q_i(\boldsymbol{t}) = 1$. Este parâmetro é calculado através da seguinte expressão (Contini & Matuzas, 2010, p. 16):

$$RAW_i(t) = \frac{Q_0(t)|_{q_i=1}}{Q_0(t)} \tag{34}$$

O valor numérico de RAW encontra-se sempre entre os valores $1 \leq RRW_i(t) < \frac{1}{Q_0(t)}$.

Para realizar a análise de importância dos eventos pertencentes à árvore de falhas de referência, é considerado, para esse sistema, um tempo de operação de 5 horas com cada evento com um valor de MTBF diferente: $MTBF_A = 1000h$, $MTBF_B = 2000h$, $MTBF_C = 750h$ e $MTBF_D = 900h$.

A partir da disposição dos eventos ao longo da árvore, analisando os componentes em paralelo e em série e através das funções de fiabilidade obtém-se o valor real da probabilidade de falha do evento topo de 4.97×10^{-5}.

Após o cálculo do valor real do evento topo é obtida a expressão simbólica da *upper bound approximation* a partir dos *minimal cut sets* obtidos anteriormente. (Os símbolos A, B, C e D representam diretamente a sua probabilidade de falha, por exemplo, $A \equiv q_A$).

- *Upper bound approximation*: $Q_0 = 1 - (1 - A \cdot C)(1 - B \cdot C)(1 - B \cdot D) = BD + BC + AC - BCBD - ACBD - ACBC + ACBCB$

Para o evento básico A:

- $I_B(A) \equiv BIR_A = \frac{\partial Q_0}{\partial A} = C - CBD - CBC + CBCBD = 0.006644292$
- $I_C(A) \equiv CRIT_A = \frac{q_A \cdot BIR_A}{Q_0} = 0.6659167$
- $FV_A = \frac{AC}{Q_0(t)} = 0.6659369$
- $RRW_A = \frac{Q_0}{Q_0|_{q_A=0}} = \frac{BD+BC+AC-BCBD-ACBD-ACBC+ACBCB}{BD+BC+0 \cdot C-BCBD-0 \cdot CBD-0 \cdot CBC+0 \cdot CBCB} = 2.0892$
- $RAW_A = \frac{Q_0|_{q_A=1}}{Q_0} = \frac{BD+BC+1 \cdot C-BCBD-1 \cdot CBD-1 \cdot CBC+1 \cdot CBCB}{BD+BC+AC-BCBD-ACBD-ACBC+ACBCB} = 105.011$

Os valores de todos os parâmetros para todos os eventos encontram-se na tabela seguinte:

Tabela 6 – Valores de parâmetros de importância de todos os eventos básicos
Fonte: (autor, 2025)

Eventos\Parâmetros	BIR	CRIT	FV	RRW	RAW
A	0.006644	0.6659167	0.6659369	2.0892	105.011
B	0.01218	0.61133	0.61136	1.918	191.633
C	0.007484	0.99923	0.99932	4.565	117.77
D	0.002496	0.278	0.278	1.278	40.063

Com todos os parâmetros de importância calculados para os eventos básicos da árvore, o próximo passo consiste em determinar quais eventos requerem maior atenção no contexto do sistema analisado. Esta etapa é crucial para identificar os componentes mais relevantes para a fiabilidade e segurança do sistema, uma vez que uma intervenção direcionada a esses elementos pode resultar em melhorias significativas na performance global.

No entanto, não existe uma abordagem única ou universalmente aceite para decidir qual evento é mais importante, já que essa decisão depende diretamente do objetivo específico da análise (Contini & Matuzas, 2010, p. 3).

Por exemplo, caso o foco seja reduzir a probabilidade do evento topo, pode-se priorizar eventos com elevados valores de CRIT ou FV. Esta análise tem como objetivo identificar eventos básicos que, se mitigados, reduziriam a probabilidade do evento topo de forma imediata. É uma análise mais direta e focada no estado atual do sistema, priorizando intervenções que tenham um efeito imediato na redução da probabilidade de ocorrência do evento topo.

Por outro lado, se a intenção for identificar melhorias com maior impacto potencial na segurança, parâmetros como RRW, BIR ou RAW podem ser mais relevantes. Identifica onde vale mais a pena investir esforços que aumentam a segurança global do sistema.

Por exemplo, se um componente tiver uma probabilidade de ocorrência baixa, no entanto, se a sua probabilidade ocorrência ao ser completamente eliminada levasse a uma melhoria significativa na fiabilidade do sistema (elevado RRW), isto sugere que investir na redundância ou na fiabilidade desse componente seria uma melhoria significativa para a fiabilidade global do sistema.

Devido a esta variabilidade, opta-se frequentemente por uma abordagem que considere simultaneamente os diferentes parâmetros de importância.

Assim, a solução proposta para este trabalho consiste na definição de uma equação composta, onde todos os parâmetros de importância calculados são combinados de forma ponderada, atribuindo pesos específicos conforme os objetivos da análise. Esta equação permite incorporar múltiplas perspetivas, fornecendo uma visão global e equilibrada sobre a relevância de cada evento básico.

A equação é expressa da seguinte maneira:

$$I_i = a \cdot BIR_i + b \cdot CRIT_i + c \cdot FV_i + d \cdot RRW_i + e \cdot RAW_i \tag{35}$$

Onde:

- I_i é um parâmetro adimensional que representa a importância de cada evento básico;
- a, b, c, d, e são os pesos atribuídos a cada parâmetro de importância, definidos de acordo com o objetivo da análise;
- BIR_i, $CRIT_i$, FV_i, RRW_i e RAW_i correspondem aos valores normalizados dos parâmetros de importância calculados para o evento básico i.

De notar que todos os parâmetros devem estar normalizados ao realizar o cálculo da equação, isto é, os parâmetros cujos valores sejam acima de um, devem ser divididos pelo valor

máximo entre todos os eventos. No caso do RRW, como o valor máximo é de 4.565 todos os valores de RRW são divididos por este valor.

A tabela seguinte apresenta os valores normalizados para a análise da equação de importância:

Tabela 7 - Parâmetros de importância normalizados
Fonte: (autor, 2025)

Eventos\Parâmetros	BIR	CRIT	FV	RRW (Normalizado)	RAW (Normalizado)
A	0.006644	0.6659167	0.6659369	0.458	0.548
B	0.01218	0.61133	0.61136	0.420	1
C	0.007484	0.99923	0.99932	1	0.6145
D	0.002496	0.278	0.278	0.2799	0.209

Com todos os parâmetros normalizados obtidos, define-se o objetivo da análise para determinar quais eventos devem ser priorizados em termos de atenção e intervenção.

Conforme referido anteriormente, caso o objetivo seja reduzir a probabilidade do evento topo, é conveniente priorizar os eventos com valores elevados de CRIT ou FV. Neste caso, a análise pode ser realizada atribuindo um peso de 100% a algum destes dois parâmetros, resultando na seguinte fórmula:

$$I_i = 1 \cdot CRIT_i$$

Por outro lado, se o objetivo da análise for identificar melhorias com maior impacto potencial na segurança, parâmetros como RRW ou BIR podem ser considerados mais relevantes. Assim, pode-se atribuir pesos iguais a estes dois parâmetros (50% cada) para essa abordagem específica.

A seguinte tabela demonstra os valores de importância para cada evento para os dois tipos de análise diferente.

Tabela 8 - Valores de importância para cada evento básico
Fonte: (autor, 2025)

Eventos\Análises	$b = 1$	$a, d = 0.5$
A	0.666	0.232
B	0.611	0.216
C	0.999	0.504
D	0.278	0.141

Com base na primeira análise, verifica-se que o evento C é o mais importante, devendo receber maior atenção durante o funcionamento do sistema. Este evento apresenta maior relevância para a probabilidade do evento topo, justificando a sua priorização.

Na segunda análise, onde o foco está em aspetos relacionados à segurança e ao impacto potencial de melhorias, o evento básico C também surge como o mais relevante. Assim, ao considerar ambas análises, o evento C é aquele que merece maior atenção no contexto geral.

Este impacto justifica a necessidade de implementar medidas de mitigação e estratégias práticas para reduzir a sua contribuição para a probabilidade de falha do evento topo. Estas medidas variam desde a inserção de redundância, substituição por um componente com maior fiabilidade, modificar escolhas de *design* no componente ou a implementação de ações de manutenção preventiva.

3.3.4.2 Análise de sensibilidade

A análise de sensibilidade consiste em avaliar como variações nos parâmetros probabilísticos dos eventos básicos de uma árvore de falhas afetam a probabilidade do evento topo.

É uma técnica utilizada para entender o impacto relativo de diferentes eventos básicos nos resultados globais do sistema, fornecendo insights sobre quais parâmetros têm maior influência e quais podem ser ajustados para melhorar a fiabilidade ou segurança do sistema (Fault Tree Analysis (FTA) Guidance Material, 2005, p. 78).

Na prática, as variações dos parâmetros consistem em alterar as probabilidades de falha dos eventos básicos ou outros parâmetros, como taxas de falha, tempos de reparação (*mean down times*) e intervalos de teste (Fault Tree Analysis (FTA) Guidance Material, 2005, p. 78). Para a análise de sensibilidade realizada no contexto deste trabalho, é calculado o impacto absoluto de cada evento básico no evento topo através de variações percentuais nas probabilidades de falha de cada evento.

Para cada variação percentual de cada componente, o impacto absoluto é calculado da seguinte maneira:

$$I_{\Delta} = \frac{\left|Q_{q_i}(t) - Q_0(t)\right|}{Q_0(t)} \tag{36}$$

Onde:

- I_{Δ} é um parâmetro adimensional que representa o impacto absoluto de cada evento básico ao evento topo, dada uma taxa de variação percentual Δ.
- $Q_{q_i}(t)$ representa a probabilidade do evento topo com a probabilidade de falha do evento i variada
- $Q_0(t)$ representa a probabilidade de falha base do evento topo.

Realizando uma variação percentual de 25% e 50% aos eventos da arvore de falhas acima obtém-se os seguintes valores:

Tabela 9 - Impacto absoluto para diferentes variações percentuais
Fonte: (autor, 2025)

Eventos\Variações	q_i para 25%	$I_{25\%}$	q_i para 50%	$I_{50\%}$
A	0,0062	0,130	0,0075	0,261
B	0,0031	0,119	0,0037	0,239
C	0,0083	0,195	0,0099	0,391
D	0,0069	0,054	0,0083	0,109

A partir dos valores apresentados, observa-se que o evento C é aquele que impacta mais o sistema, e, em concordância com os resultados da análise de importância, é aquele que merece maior parte da atenção.

Concluída a análise de importância e sensibilidade da árvore de falhas, as informações obtidas tornam-se essenciais para orientar decisões que visam melhorar a fiabilidade do sistema durante as fases subsequentes do projeto e ao longo do seu ciclo de vida.

Conforme ilustrado na Figura 11, a garantia da fiabilidade da aeronave não se encerra na fase de projeto. Durante a sua vida operacional, a aeronave está sujeita a diversas condições e fatores externos que podem impactar o desempenho esperado dos sistemas e subsistemas. Para assegurar o cumprimento contínuo das métricas de fiabilidade, como a disponibilidade, o tempo médio entre falhas (MTBF) e o tempo médio para reparação (MTTR), é necessário adotar práticas da gestão operacional da fiabilidade.

Neste contexto, entra em destaque o papel do *Integrated Logistics Support* (ILS), que funciona como um complemento ao DFR. O ILS permite não apenas monitorizar e avaliar o desempenho da aeronave em operação, mas também otimizar os recursos logísticos, como peças sobressalentes, planeamento de manutenção e formação de equipas.

O capítulo seguinte explora as práticas de *Integrated Logistics Support* que contribuem para a gestão da fiabilidade, com foco nas métricas que suportam esta abordagem e que se encontram relacionadas com este trabalho.

3.4 Suporte logístico integrado

O *Integrated Logistics Support* (ILS) é uma abordagem sistemática e proativa que visa otimizar as atividades logísticas durante a fase de operação e minimizar os custos do ciclo de vida (LCC[17]) de um sistema. O ILS é um elemento essencial da engenharia de sistemas em programas aeroespaciais e de defesa, garantindo que os requisitos de suporte sejam integrados desde as fases iniciais do desenvolvimento até à fase de operação e eventual descontinuação do sistema (Vaskic & Paetzold, 2019, p. 3541).

Na aviação, este processo integra vários elementos de logística com o objetivo de assegurar que todos os aspetos lógicos trabalhem em harmonia para atingir os objetivos do processo.

Esses são: planeamento de manutenção; suporte a recursos informáticos; Instalações; apoio ao abastecimento; equipamento de suporte e teste; embalagem, manuseamento, armazenamento e transporte; pessoal e recursos humanos; formação e sistemas de formação; e dados técnicos (Jacobsen, 1996, p. 18).

De acordo com Paul McIlvaine (citado em Jacobsen, 1996, p.17), além da integração dos elementos de logística, a ILS deve integrar mais duas áreas no processo total de aquisição. Estas são o tempo (ciclo de vida da aeronave) e disciplinas relacionadas a logística, conforme ilustrado na figura 15.

[17] Custo total de um sistema ou produto ao longo de toda a sua vida útil, incluindo custos de aquisição, operação, manutenção e abate ou *phase out*.

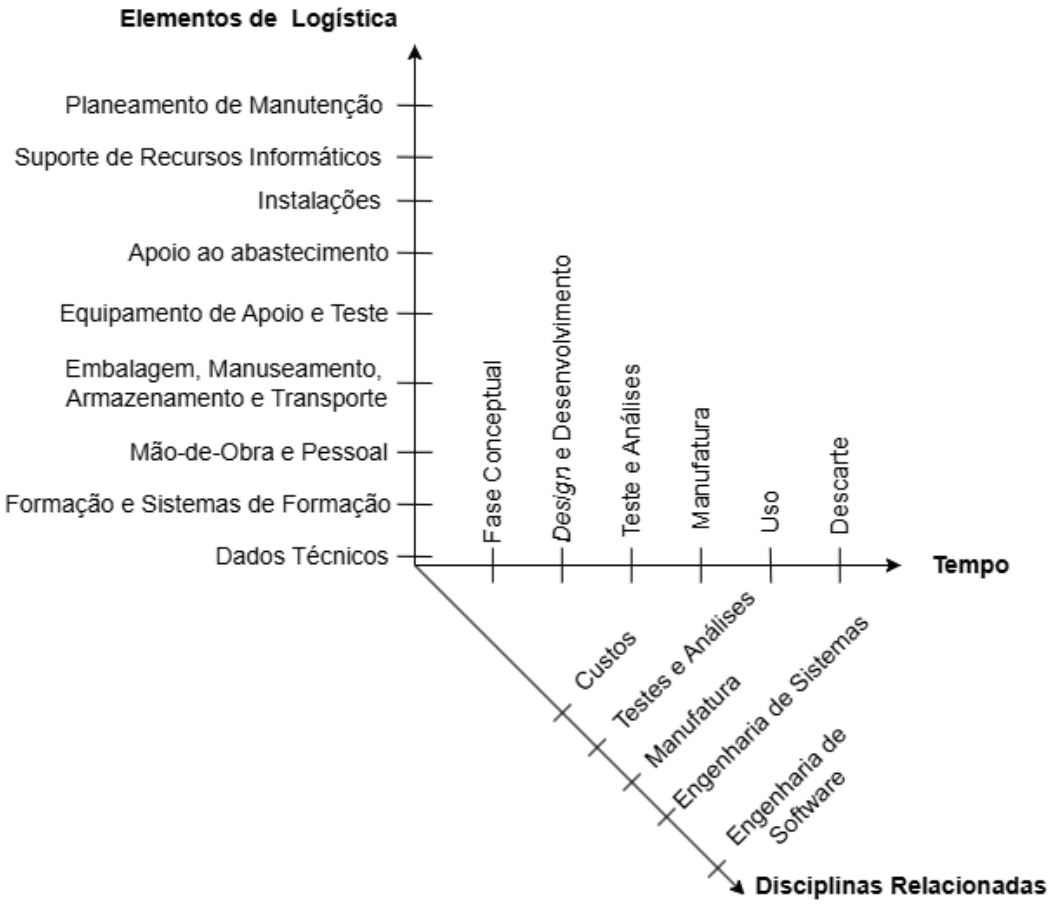


Figura 15 - Dimensões da Logística
Fonte: adaptado de Jacobsen (1996, p. 18)

Estas disciplinas do ILS estão intrinsecamente ligadas a conceitos comummente utilizados por todo o ciclo de vida da aeronave. Esta dissertação explora os conceitos de manutibilidade, manutenção e fiabilidade para definir a fase operacional da aeronave. Estes conceitos interagem com as disciplinas mencionadas de forma complexa com influência direta nos objetivos definidos pelo processo.

Um exemplo disto é, uma maior fiabilidade de um sistema (na ótica do MTBF) pode levar a uma maior disponibilidade do mesmo, ou então uma melhor manutibilidade pode reduzir o tempo de manutenção, aumentando também a disponibilidade.

3.4.1 Disponibilidade

A disponibilidade é definida como a "capacidade de um sistema ou produto estar em condições de desempenhar a função requerida" e é uma medida do tempo durante o qual um sistema está operacional e pronto para cumprir a sua função em comparação com o tempo total (Denning, 2012, p. 6).

De acordo com Denning (2012, p. 6), a disponibilidade pode ser expressa como:

$$A = \frac{Tempo\ em\ operação}{Tempo\ total} = \frac{Uptime}{Uptime + Downtime} \tag{37}$$

Onde o tempo de operação refere-se ao intervalo em que o sistema está funcional e em condições de realizar a sua função prevista e o tempo total representa o período de avaliação completo, que inclui tanto o tempo em que o sistema está operacional como o tempo em que está indisponível devido a manutenção, reparos ou falhas.

É definido como a soma de tempo em operação (*Uptime*) e o tempo de inatividade (*Downtime*) (Denning, 2012, p. 6).

No contexto da indústria aeronáutica, a disponibilidade refere-se à capacidade de uma aeronave estar operacional e pronta para executar as missões programadas, minimizando o tempo de inatividade associado a manutenção.

Considerando um cenário onde os elementos da logística (Figura 15) são ideais, isto é, pressupondo que um técnico de manutenção qualificado, as peças sobressalentes, as ferramentas e o equipamento de teste necessários para realizar ações de manutenção estão todos imediatamente à disposição, a disponibilidade é dada por (NASA, 1997, p. 8):

$$A = \frac{MTBF}{MTBF + MTTR} \tag{38}$$

Onde o MTBF representa a média do tempo em que o sistema permanece operacional antes de falhar, e o MTTR representa o tempo médio necessário para restaurar o sistema à sua condição operacional após uma falha. A soma dos dois valores ($MTBF + MTTR$) reflete o ciclo completo de operação e recuperação do sistema. Consequentemente, a razão entre o MTBF e este ciclo total indica a fração de tempo em que o sistema está operacional.

Por exemplo, dentro do sistema de navegação da aeronave, se um recetor VOR apresentar uma distribuição de falha exponencial com um MTBF de 500h e um MTTR de 5h então o componente tem uma disponibilidade de $A = \frac{MTBF}{MTBF+MTTR} = \frac{500}{500+5} = 0.99$, o que indica que o componente estará operacional cerca de 99% do tempo.

3.4.1.1 MTBF

A fiabilidade de um sistema é a probabilidade de este desempenhar a sua função corretamente dentro de um período definido, sob condições específicas. Em termos práticos, a fiabilidade, frequentemente expressa pelo MTBF, mede o período em que uma máquina pode operar continuamente sem avarias, desde que esteja a funcionar dentro das especificações para as quais foi projetada (Afsharnia, 2017, p. 101).

A relação entre fiabilidade e disponibilidade operacional é direta, como evidenciado pela expressão $A = \frac{MTBF}{MTBF+MTTR}$ (38) e pelo que já foi referido acima "…uma maior fiabilidade de um sistema (na ótica do MTBF) pode levar a uma maior disponibilidade do mesmo…".

Esta relação demonstra que ao aumentar o MTBF de um sistema não só reduz a frequência de intervenções corretivas, mas também assegura maior operacionalidade do sistema.

Portanto, ao considerar a frequência das falhas como um fator determinante da fiabilidade, o MTBF surge como uma métrica fundamental para avaliar e melhorar o desempenho de sistemas. No âmbito deste trabalho, além da construção da árvore de análise de falhas e da consequente análise de importância e de sensibilidade, o programa de fiabilidade estudado também inclui o cálculo do MTBF de todos os sistemas a analisar.

Este subcapítulo apresenta a metodologia utilizada pelo programa para o cálculo do parâmetro de MTBF (considerando apenas os componentes que seguem distribuição exponencial).

Do capítulo 3.1.3.1 retiramos que a fiabilidade de um sistema com componentes em série é dada pela expressão $R_s(t) = R_1(t) \times R_2(t) \times \ldots \times R_n(t)$ (21):

$$R_s(t) = R_1(t) \times R_2(t) \times \ldots \times R_n(t) \tag{21}$$

Assumindo que todos os componentes apresentam uma taxa de falha constante ($\boldsymbol{\lambda_i}$), as suas fiabilidades individuais podem ser descritas pela função exponencial $R(t) = e^{-\lambda_i t}$. Assim, ao expandir a expressão para obter R_s obtém-se:

$$R_s(t) = \prod_{i=1}^{n} R_i(t) = \prod_{i=1}^{n} \exp(-\lambda_i t) = \exp(-\sum_{i=1}^{n} \lambda_i \times t) \tag{22}$$

Onde $\lambda_s = \sum_{i=1}^{n} \boldsymbol{\lambda_i}$. Visto que todos os componentes seguem distribuições exponenciais com taxas de falha constantes, a taxa de falha do sistema, por ser a soma de todas as taxas de falha individuais, esta será também constante. Por ser constante, a curva de fiabilidade do sistema irá seguir uma distribuição exponencial (ilustrado na figura 16 pela curva a roxo).

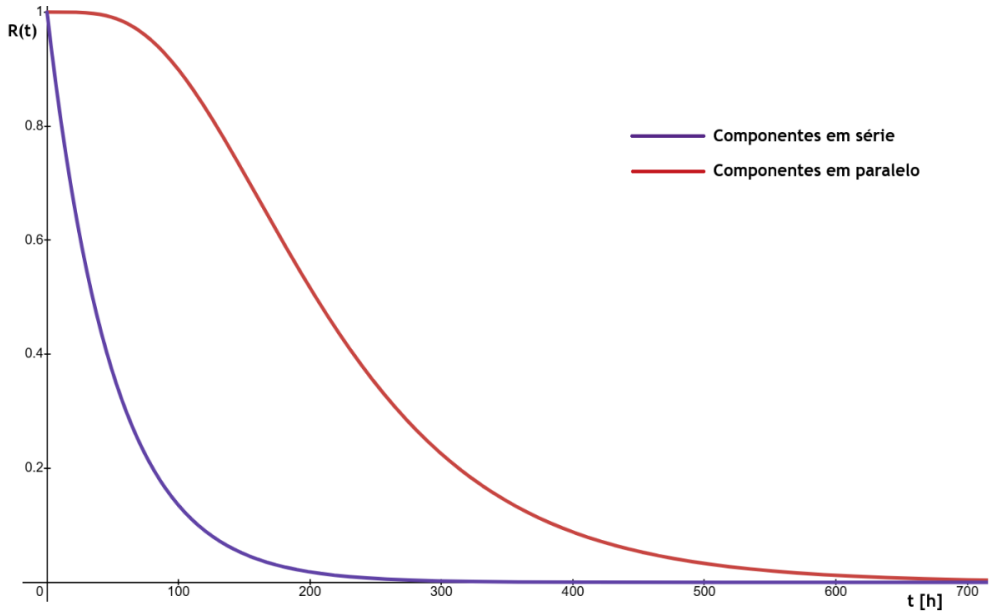


Figura 16 - Função fiabilidade para sistemas em série e em paralelo

Fonte: (autor, 2025)

Como taxa de falha do sistema é constante, o valor de MTBF é também constante e é dado por:

$$MTBF_s = \frac{1}{\lambda_s} = -\frac{ln\left(R_s(t)\right)}{t} \tag{39}$$

Considerando um sistema com n componentes idênticos com taxas de falha λ_c em paralelo, a fórmula de fiabilidade do sistema é dada pela expressão (Afsharnia, 2017, p. 113):

$$R_s = 1 - \left(1 - e^{-\lambda_c t}\right)^n \tag{40}$$

Para uma taxa de falha ($\boldsymbol{\lambda_c}$) específica e um número de componentes em paralelo (n) em específico, é obtida a curva a vermelho da figura 16.

Ao observar esta curva, conclui-se que a mesma não segue uma distribuição exponencial, o valor de fiabilidade mantém-se máximo nos primeiros instantes de operação do sistema, devido à redundância imposta inicialmente, e à medida que os componentes vão falhando, a fiabilidade do sistema diminui.

Como a configuração em paralelo não segue uma curva exponencial de falha, a taxa de falha do sistema não permanece constante, impossibilitando a modelação da fiabilidade de um sistema de componentes com taxa de falha constante através de um modelo de taxa de falha do sistema constante (Afsharnia, 2017, p. 113).

Para determinar a taxa de falha de um sistema com n componentes em paralelo, utiliza-se a relação entre a função de fiabilidade, a função de densidade de probabilidade e a taxa de falha (Afsharnia, 2017, p. 113), apresentada na expressão $\lambda(t) = \frac{f(t)}{R(t)}$ (12):

$$\lambda(t) = \frac{f(t)}{R(t)} \tag{12}$$

Ao expandir esta expressão obtém-se a expressão $\lambda(t) = \frac{f(t)}{R(t)} = \frac{-\frac{dR(t)}{dt}}{R(t)}$ (13):

$$\lambda(t) = \frac{f(t)}{R(t)} = \frac{-\frac{dR(t)}{dt}}{R(t)} \tag{13}$$

Este cálculo é feito pelo programa de análise de fiabilidade, realizado através de programação em *python*.

Após a construção dos sistemas, o programa, através da disposição dos componentes ao longo da árvore de falhas, calcula a fiabilidade de todo o sistema em função do tempo. Após esse cálculo o programa realiza uma análise temporal com valores numéricos de t, obtendo então valores numéricos de $R(t)$.

Por não lidar com a função contínua de $R(t)$, o programa substitui, na fórmula, a derivada contínua $\frac{dR(t)}{dt}$ pela diferença finita $\frac{\Delta R(t)}{\Delta t}$.

Ou seja, para cada intervalo de tempo $t_{i+1} - t_i$, o programa determina a variação da fiabilidade $\Delta R = R(t_{i+1}) - R(t_i)$ e divide depois pela variação de tempo $\Delta t = t_{i+1} - t_i$ para obter uma aproximação da derivada:

$$\frac{\Delta R}{\Delta t} \approx \frac{dR(t)}{dt}$$

O passo seguinte consiste em calcular a taxa de falha efetiva para cada instante de tempo t_i:

$$\lambda_e(t_i) = \frac{-\frac{\Delta R}{\Delta t}}{R(t_i)} \tag{41}$$

Como a taxa de falha para estes sistemas (em paralelo) irá variar, é calculada a taxa de falha média $\lambda_{média}$ tendo em conta todos os instantes de tempo calculados.

Com a taxa de falha média, o cálculo do MTBF do sistema é dado por:

$$MTBF = \frac{1}{\lambda_{média}}$$

Este método é universal, pode ser aplicado a sistemas com qualquer configuração, sejam componentes em série, paralelo ou configurações mistas e é o método utilizado para o cálculo do MTBF de todos os sistemas analisados pelo programa.

Além do parâmetro de MTBF, a disponibilidade operacional de um sistema também depende do tempo necessário para restaurá-lo ao estado funcional após uma falha. Este aspeto é capturado pelo conceito de manutibilidade, frequentemente representado pelo MTTR (*Mean Time to Repair*).

3.4.2 Manutibilidade

A manutibilidade é definida como a probabilidade de que um componente ou sistema que sofreu uma avaria possa ser restaurado ao seu estado operacional dentro de um período especificado (Pecht, 1999, p. 20). Este conceito reflete a facilidade e a rapidez com que se pode reparar um sistema, garantindo que ele volte a funcionar conforme os parâmetros estabelecidos pelo fabricante ou operador.

No contexto da operação da aeronave, esta disciplina integra o desenvolvimento dos parâmetros necessários para restaurar os sistemas dentro das condições de operação. Estes parâmetros são denominados como índices de manutibilidade, que também são fundamentais para o desenvolvimento do processo DfR na fase de projeto da aeronave, como por exemplo, o tempo médio para reparar (MTTR), ilustrado na figura 17 como tempo de reparação.

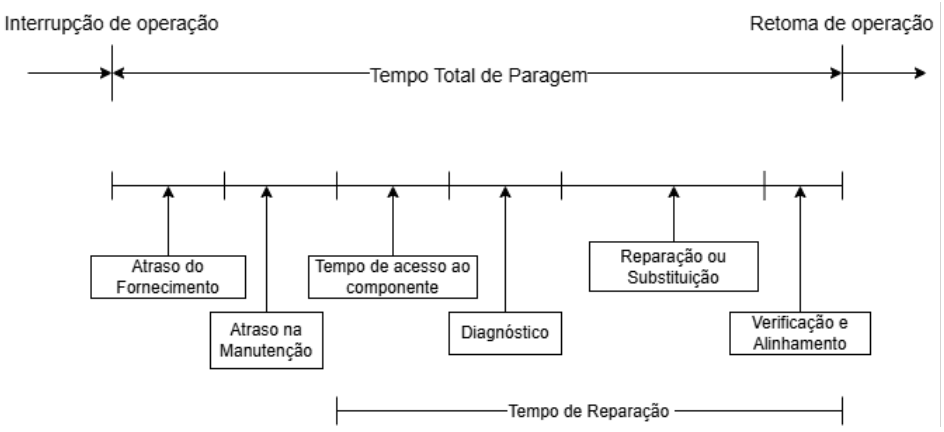


Figura 17 - Ciclo do tempo de paragem
Fonte: adaptado de Ebeling (2003, p. 190)

Em cenários reais, onde os elementos de logística não são considerados ideais, o tempo de paragem (*downtime*) de um produto já não representa apenas o tempo de reparação do produto (MTTR), mas sim o tempo desde a interrupção de operação da aeronave até à retoma de operação da mesma, incluindo todos os intervalos desde o atraso do fornecimento de peças à verificação e alinhamento final do produto à aeronave.

Ebeling (2003, p. 190) reconhece que os tempos de atrasos no fornecimento e manutenção são influenciados por fatores externos e define o tempo de reparação inerente do produto como sendo apenas a soma das durações das ações de manutenção desde o tempo demorado a obter acesso ao componente, o que pode variar dependendo onde o componente estiver situado dentro da aeronave, até ao tempo demorado a realizar a verificação e alinhamento final do componente.

Tal como o MTBF se encontra intimamente ligado com o conceito de fiabilidade, o parâmetro de MTTR também se encontra intimamente ligado ao conceito de manutibilidade. Quanto menor o MTTR, maior a manutibilidade, porque indica que o sistema foi projetado para facilitar intervenções rápidas e eficientes o que leva a uma maior disponibilidade do produto.

3.4.3 Manutenção

A manutibilidade é uma característica inerente ao *design* de um produto, que influencia diretamente a facilidade e eficiência com que a manutenção pode ser realizada. Com base nesta característica, desenvolve-se o conceito de manutenção, que engloba o conjunto de atividades e procedimentos necessários para restaurar ou preservar a funcionalidade de um produto ao longo do tempo.

Enquanto a manutibilidade define o potencial de uma unidade ser reparada, a manutenção representa a aplicação prática desse potencial, descrevendo as tarefas específicas necessárias para garantir o desempenho contínuo e adequado do produto.

A EASA e a FAA, através do documento (Maintenance Annex Guidance, 2021, p. 64) definem a manutenção como a execução de qualquer uma ou mais das seguintes ações: inspeção, revisão geral, reparação, preservação ou substituição de peças, materiais, equipamentos ou componentes de um produto aeronáutico civil, com o objetivo de garantir a aeronavegabilidade contínua de tal produto; ou a instalação de alterações ou modificações previamente aprovadas, realizadas em conformidade com os requisitos estabelecidos pela autoridade técnica competente.

Estas ações estão geralmente agrupadas em dois grupos diferentes, a manutenção programada e a manutenção não programada.

A manutenção programada é uma manutenção que é repetidamente realizada em intervalos definidos de acordo com um programa de manutenção específico, sendo frequentemente denominada de manutenção preventiva (Pita, p. 16). De forma simples, a manutenção programada, ou preventiva, consiste na prática de realizar inspeções e serviços regulares em equipamentos de modo a evitar falhas antes que elas ocorram.

Esta abordagem de gestão de manutenção é predominantemente um planeamento baseado em tempo ou tarefas recorrentes, concebidas para manter níveis aceitáveis de fiabilidade e disponibilidade (Mobley, 2004, p. 9).

Neste contexto, além de implementar redundâncias, uma abordagem para garantir os níveis de fiabilidade e segurança exigidos nos sistemas é a utilização de estratégias baseadas em critérios *hard time* e *on condition*. Estas estratégias permitem mitigar o risco de falhas catastróficas, ajustando os intervalos de manutenção e otimizando os níveis de desempenho.

Hard time é um processo onde um componente é submetido a manutenção ou substituição após um limite de vida ou um número de ciclos de operação especificado, independentemente do seu estado atual (Kinnison & Siddiqui, 2012, p. 19).

On condition é um processo onde, dentro da vida útil do componente, são realizadas inspeções regulares, testes e análises para avaliar a sua integridade e determinar o seu estado. Este método permite detetar problemas antes que ocorram falhas, assegurando que os componentes funcionem corretamente e de forma segura durante toda a sua vida útil (Kinnison & Siddiqui, 2012, p. 19).

Em muitos casos, o valor de MTBF inicial de determinados componentes pode não ser suficiente para atingir o objetivo de probabilidade de falha catastrófica de 10^{-9}.

Por essa razão, para este trabalho, assume-se que, para alguns componentes, será necessário adotar o critério de *hard time*, onde estes componentes serão substituídos ou reparados antes de atingirem o seu MTBF calculado.

Este processo reduz a probabilidade de falha desses componentes, aumentando de forma prática e virtual o valor de MTBF do mesmo. Por exemplo, ao remover um componente a metade do seu tempo de MTBF original, reduz-se o intervalo durante o qual ele pode falhar, garantindo maior controlo sobre o seu desempenho e contribuindo para a redução da probabilidade de falha global do sistema.

O novo valor de MTBF aumentado é calculado através da seguinte expressão:

$$MTBF_1 = \frac{MTBF_0}{1 - e^{-\frac{T}{MTBF_0}}} \quad (42)$$

Onde:

$MTBF_1$: representa o novo valor de MTBF aumentado devido à ação de manutenção programada;

$MTBF_0$: representa o valor de MTBF original, sem qualquer intervenção de manutenção programada;

$\boldsymbol{T}$: representa o valor de tempo no qual o componente é submetido a manutenção.

A dedução desta expressão encontra-se no Apêndice D.

Por exemplo se uma bomba de combustível apresentar um valor de MTBF de 10.000 h e no mesmo componentes for implementado o critério de *hard time* às 8.000 h então o valor de MTBF resultante desta ação de manutenção é dado por:

$$MTBF_1 = \frac{10000}{1 - e^{-\frac{\mathbf{8000}}{\mathbf{10000}}}} = 18159\ h$$

Devido à implementação do critério de hard time na bomba de combustível, o MTBF desta sofreu um aumento de 81.6% resultando num maior valor de fiabilidade do componente ao longo do tempo.

Para além do critério de manutenção *hard time*, implementado para determinados componentes, é também introduzido no programa de análise de fiabilidade o critério de manutenção *on condition*. Este critério será aplicado a componentes que não têm o critério *hard time* implementado.

Neste critério, a manutenção é realizada de forma preditiva e baseada no estado real do componente, com intervenções planeadas apenas quando determinados limites de fiabilidade previamente definidos são atingidos.

Para este propósito, o programa desenvolvido utiliza limites de alerta como indicadores para acionar ações de manutenção. Estes limites de alerta são definidos como percentagens da fiabilidade inicial do sistema, como, por exemplo, 98%.

Quando o sistema atinge este limite, o programa calcula o tempo correspondente, sinalizando que é necessário realizar uma ação de manutenção antes que o sistema atinja o limite crítico, definido com base numa percentagem ainda mais baixa (por exemplo, 97%).

A manutenção não programada é um tipo de manutenção realizada com o objetivo de restaurar a condição original projetada da aeronave após a ocorrência de uma falha ao sistema, frequentemente designada de manutenção reativa (Pita, p. 16).

Um processo que é considerado uma ação de manutenção não programada é o processo que decorre da opção de *condition monitoring* (CM).

O processo de CM consiste na monitorização de parâmetros de funcionamento. Como os componentes não apresentam características que permitam estabelecer critérios de substituição programada, também não existem tarefas de manutenção específicas para avaliar a sua vida útil, nem a necessidade de os substituir (Kinnison & Siddiqui, 2012, p. 21).

Desta forma, os itens sujeitos a CM são operados de forma controlada/monitorizada mantendo-se em operação enquanto os parâmetros relativos ao desempenho se mantenham dentro dos limites previstos.

Embora uma manutenção eficaz vise prevenir falhas, é impossível eliminar completamente a possibilidade de falhas durante o serviço. Portanto, a indústria aeronáutica desenvolveu três técnicas específicas para minimizar o impacto dessas falhas durante a operação da aeronave antes que a manutenção programada ocorra. Estas técnicas incluem o uso de *Line Replaceable Units* (LRU), sistemas redundantes e a implementação da *Minimum Equipment List* (MEL) (Kinnison & Siddiqui, 2012, p. 11).

Uma LRU é um componente ou sistema que foi concebido de tal forma que as peças que mais frequentemente avariam podem ser rapidamente removidas e substituídas na aeronave. Isto permite que a mesma seja devolvida ao serviço programado sem atrasos indevidos para manutenção (Kinnison & Siddiqui, 2012, p. 12).

A redundância consiste na implementação de componentes ou sistemas adicionais que garantem a continuidade da operação em caso de falha de um componente principal. Os sistemas redundantes em paralelo envolvem o uso de dois ou mais sistemas ou componentes independentes, mas operando simultaneamente, com cada um realizando a mesma função.

Caso um dos sistemas ou componentes falhe, o outro continua a operar sem interrupção do serviço, permitindo a continuidade da operação sem afetar a segurança da aeronave.

A terceira técnica utilizada para minimizar atrasos na manutenção na aviação é a *Minimum Equipment List* (MEL). Esta lista permite que a aeronave seja despachada para o serviço com certos componentes inoperativos desde que a perda de funcionalidade não afete a segurança e a operação do voo. Esses itens são cuidadosamente determinados pelo fabricante e sancionados pela autoridade reguladora durante as fases iniciais de design e teste da aeronave

(Kinnison & Siddiqui, 2012, p. 12).

O fabricante emite uma *Master Minimum Equipment List* (MMEL), ilustrada na Tabela 10 que inclui todo o equipamento e acessórios disponíveis para o modelo da aeronave. É da responsabilidade da companhia aérea que a mesma adapte esta lista para a sua própria configuração da aeronave de forma a criar a MEL.

A MEL é então aprovada pelas autoridades nacionais de aeronavegabilidade do operador. Em Portugal, a autoridade responsável para esta função é a ANAC.

A MEL não pode ser menos restritiva que a MMEL, isto é, a MEL deve cumprir, no mínimo, os mesmos critérios de segurança e restrições estabelecidos na MMEL. A MEL pode ser mais restritiva (ou seja, incluir mais itens que precisam estar operacionais), mas não pode ser mais permissiva (ou seja, permitir mais itens inoperacionais do que a MMEL).

Tabela 10 - Exemplo de template de Master Minimum Equipment List
Fonte: adaptado de EASA (2018, p. 21)

Aircraft:	**Page:**
System and Sequence Numbers Item (1)	(2) Rectification Interval (3) Number Installed (4) Number Required for Dispatch (5) Remarks or Exceptions

A MMEL, ilustrada na Tabela 10, é composta por cinco colunas com cada uma a definir um objetivo na construção e definição da MEL. Estas colunas, numeradas de (1) a (5), são definidas da seguinte maneira (EASA, 2018, pp. 17-18):

(1) Item – Contém a descrição do equipamento, sistema, componente ou função. A caracterização do equipamento é realizada utilizando os capítulos ATA 2200;

(2) *Rectification Interval* – Esta coluna indica a categoria de intervalo de retificação:

- *Category A*: Avarias dentro desta categoria devem ser retificadas em um a dois dias, a menos que especificado de outra forma, podendo ser uma MMEL de voo de um dia, dependendo das restrições;

- *Category B*: Avarias de unidades dentro desta categoria devem ser retificados em três dias consecutivos de calendário, excluindo o dia em que a anomalia foi detetada;

- *Category C*: Avarias de unidades dentro desta categoria devem ser retificadas em dez dias de calendário, excluindo o dia em que a anomalia foi detetada;

- *Category D*: Avarias de unidades dentro desta categoria devem ser retificadas em cento e vinte dias de calendário, excluindo o dia em que a anomalia foi detetada;

(3) *Number Installed* – Indica a quantidade de unidades normalmente instaladas na aeronave;

(4) *Number Required for Dispatch* – Indica a quantidade mínima de unidades requeridas para libertar a aeronave para operação, de acordo com as condições estipuladas na coluna (5);

(5) *Remarks or Exceptions* – Inclui uma declaração que proíbe ou permite a operação com um número específico de unidades inoperáveis.

Estas técnicas de manutenção ajudam na suavização da carga de trabalho de manutenção. No entanto, muitos componentes e sistemas de uma aeronave não permitem este tipo de ajustes de manutenção flexíveis. As autoridades aeronáuticas e os fabricantes frequentemente exigem inspeções e modificações específicas dentro de prazos rigorosos. É necessário, portanto, que o departamento de engenharia e manutenção de uma companhia aérea implemente um programa de manutenção abrangente e eficaz.

Em concordância com o tema do trabalho, além da implementação de ações de manutenção programada (*hard time)* a redundância é a única técnica considerada que, de forma direta, contribui para a fiabilidade dos sistemas, o que aumenta diretamente a disponibilidade operacional dos componentes.

3.4.4 MIL-STD-1808C

A aeronave é composta por múltiplos sistemas que, em conjunto, determinam as características e capacidades da plataforma aérea. Para efetuar uma caracterização precisa destes sistemas o programa adota a classificação dos sistemas e componentes da aeronave de acordo com a especificação MIL-STD-1808C (Interface Standard System Subsystem Sub-Subsystem Numbering, 2020).

A classificação é composta por seis números, XX-YY-ZZ, os primeiros dois números (XX) representam o sistema principal da aeronave (chamado de *system number*), os números YY representam uma subdivisão do sistema principal, (chamado de *subsystem number*) e os últimos números ZZ identificam um subsistema ou componente específico dentro da seção (chamado de *sub-subsystem number*).

Capítulo 4 – Modelo de determinação da fiabilidade

4.1 Definição dos sistemas

O primeiro passo para a análise da fiabilidade do LUS-222 consiste em definir todos os sistemas que contribuem de forma direta para o funcionamento da aeronave ao longo de todo o voo e cuja falha afeta diretamente a segurança do voo e da aeronave.

Apenas são analisados os sistemas cuja falha contribui diretamente para a falha catastrófica da aeronave e, dentro de cada sistema, foram considerados apenas os subsistemas ou componentes cuja falha conduz à perda do sistema a que pertencem. Neste estudo, a análise incide sobre os seguintes sistemas críticos: sistema elétrico, sistema propulsivo, aviónicos, FMS e superfícies sustentadoras.

4.1.1 Sistema elétrico

Sistema que integra os componentes responsáveis pela geração, armazenamento e distribuição de energia elétrica necessária para o funcionamento dos diversos equipamentos e sistemas a bordo da aeronave.

Para análise do LUS-222 considerou-se como perda do sistema elétrico, a perda total de corrente elétrica. As condições de falha consideradas para este sistema são a falha intrínseca de ambos alternadores ou a perda do sistema propulsivo que leva à falha dos alternadores.

A arquitetura do sistema elétrico considerada contém os seguintes subsistemas:

- Alternador: Componente que converte energia mecânica do motor em energia elétrica alternada (AC) e é o responsável por alimentar sistemas elétricos da aeronave.

- Sistema propulsivo: sistema que engloba os componentes necessários para a produção de potência da aeronave.

4.1.2 Sistema propulsivo

Sistema que integra todos os subsistemas e componentes necessários para fornecer potência propulsiva à aeronave.

Para o estudo do LUS, considerou-se a perda do sistema propulsivo como a perda de ambos motores da aeronave. O sistema inclui o estudo dos seguintes subsistemas:

- Motor: Componente crítico que converte energia química (combustível) em energia mecânica para gerar impulso à aeronave;

- *Nacelle*: Subsistema que engloba as estruturas responsáveis pelo suporte, proteção e montagem do motor na aeronave. O *nacelle* é uma carenagem aerodinâmica que envolve o motor, protegendo-o e otimizando o fluxo de ar e permitindo a passagem de sistemas como o de combustível e cabeamento elétrico. Para este estudo, é analisado o sistema de combustível que atravessa o *nacelle* de cada motor.

- Combustível: Subsistema que inclui componentes que contribuem para o fornecimento de combustível até ao motor da aeronave. Os componentes em análise são as bombas de combustível.

4.1.3 Sistema de controlo de voo

Este sistema está dividido em diferentes subsistemas independentes, cada um responsável por auxiliar na manobra da aeronave em torno dos seus três eixos principais: lateral, longitudinal e vertical.

Assume-se que todos estes subsistemas possuem uma arquitetura mecânica, sendo constituídos por componentes como cabos, roldanas, *bellcranks*, entre outros. Por se tratar de sistemas mecânicos, os modos de falha associados estão predominantemente relacionados com a deterioração dos componentes ao longo do tempo, resultante do uso contínuo, exposição a condições ambientais variáveis e tensões mecânicas recorrentes. Contudo, neste trabalho, considera-se que a manutenção preventiva é realizada de forma eficaz, eliminando a ocorrência de falhas decorrentes da degradação ou desgaste dos componentes.

Desta forma, a análise de fiabilidade destes sistemas restringe-se à avaliação de uma condição de falha específica: a falha do sistema de anti gelo em cada atuador de cada superfície aerodinâmica. A inclusão de outros modos de falha, nomeadamente aqueles associados a componentes hidráulicos do sistema de controlo de voo, ultrapassaria o âmbito desta análise preliminar. Opta-se por um critério mais restritivo, permitindo manter o foco num subconjunto específico de falhas e limitar a complexidade do estudo.

Relativamente ao sistema de anti gelo, a sua análise também foi excluída desta avaliação de fiabilidade. A explicação desta decisão encontra-se no capítulo 4.2.1.2 que contém todas as restrições consideradas para esta análise.

Desta forma, a análise dos sistemas de controlo de voo, que, apesar de serem considerados críticos à operação da aeronave, não são considerados para este trabalho.

Assume-se da mesma forma que o movimento dos flaps é realizado de forma mecânica, através do seletor de flaps localizado no cockpit, que está ligado a um motor elétrico. A condição de falha do sistema dos flaps a considerar é, portanto, a perda do sistema elétrico.

4.1.4 Aviónicos

Os aviónicos de uma aeronave referem-se aos sistemas eletrónicos que suportam a navegação, comunicação, monitorização e controlo de vários aspetos do voo. Esses sistemas incluem componentes como o radar, os sistemas de gestão de voo (FMS), sistemas de navegação e outros instrumentos que auxiliam a operação segura e eficiente da aeronave.

Para a análise do LUS-222, o sistema FMS foi removido do grupo dos aviónicos para ser analisado como um sistema independente e crítico. No âmbito dos aviónicos, o foco desta análise centra-se exclusivamente no sistema de navegação, uma vez que este desempenha um papel essencial na determinação da posição e trajetória da aeronave.

O sistema de navegação foi subdividido nos seus dois principais subsistemas: GNSS (*Global Navigation Satellite System*) e VOR (*VHF Omnidirectional Range*), os quais foram considerados independentes entre si, pois operam de forma distinta e utilizam princípios de funcionamento diferentes. O GNSS baseia-se em sinais de constelações de satélites para fornecer informações de posição e navegação, enquanto o VOR utiliza sinais transmitidos por estações terrestres para fornecer referências de direção.

A análise de fiabilidade destes sistemas centrou-se nos seus componentes mais críticos, os recetores GNSS e VOR. Estes recetores são responsáveis por processar os sinais externos e converter a informação recebida em dados úteis para a navegação da aeronave. A falha conjunta dos recetores de um sistema resulta na perda total da capacidade desse sistema em voo, comprometendo a disponibilidade da informação de navegação.

4.1.5 Flight management system (FMS)

O *Flight Management System* (FMS) é um sistema que combina vários subsistemas da aeronave que otimizam o desempenho e a eficiência do voo. Este atua como o centro de gestão do voo, processando dados de navegação, desempenho e planeamento de combustível.

Para a análise deste sistema, estudou-se uma arquitetura semelhante à que existe nas aeronaves airbus A330 e A340, esta contém uma redundância quádrupla para as interfaces de entrada e saída de dados (MCDU) e dupla redundância para os computadores da gestão do voo (FMGEC), ilustrado na Figura 18.

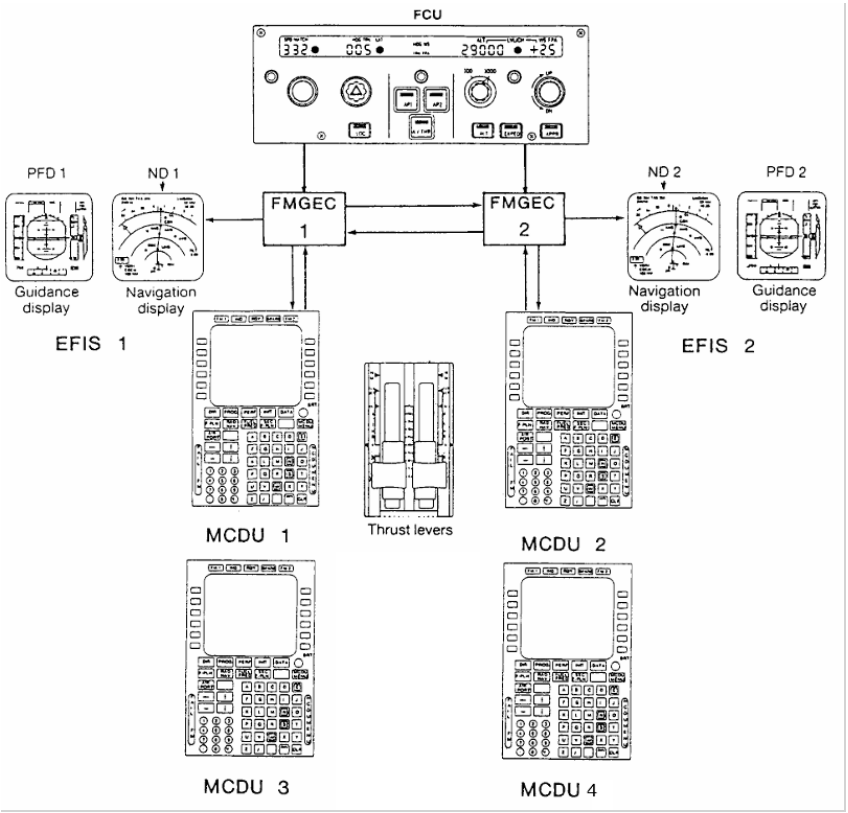


Figura 18 - Arquitetura do FMS

Fonte: adaptado de (Griguere, 1991, p. 483)

Os subsistemas analisados são os seguintes:

- Multi-Purpose Control Display Unit (MCDU): Estes subsistemas funcionam como a interface principal entre os pilotos e o sistema de gestão de voo, permitem aos pilotos inserir dados do plano de voo, como rotas e altitudes e também fornecem informações de "longo prazo" aos pilotos como o plano de voo a ser seguido, velocidades e altitudes recomendadas pelo sistema de gestão de voo bem como previsões para pontos de passagem intermédios e destino final (Griguere, 1991, p. 485).

- Flight Management and Guidance and Envelope Computers (FMGEC): Estes subsistemas são considerados como os "cérebros" do sistema de gestão de voo. Processam todos os dados de navegação e do plano de voo e calculam a trajetória ótima da aeronave. Esses dados são redirecionados para o EFIS (*Electronic Flight Instrument System*) de cada piloto.

É importante referir que, apesar dos FMGEC serem os "cérebros" do FMS, uma falha dos mesmos não significa uma falha total do sistema, isto porque os MCDU ativos têm a capacidade de realizar navegação de emergência. Uma falha de todos os MCDUS também não significa falha do sistema geral porque, apesar de não se poder introduzir novos dados de voo ou modificar

planos de voo, o sistema pode continuar a gerir a navegação e desempenho da aeronave e a aeronave seguirá o plano de voo predefinido, mesmo que os pilotos não possam alterar parâmetros.

4.1.6 Superfícies sustentadoras

As superfícies sustentadoras são os principais componentes responsáveis pela geração de sustentação da aeronave, permitindo o voo sustentado e o controlo da trajetória em diferentes fases de voo, como as asas e superfícies híper sustentadoras.

No que toca à asa, visto que a fadiga é um parâmetro que não é incluído neste estudo, a condição de perda de sustentação da asa devido a falha estrutural da mesma não é considerada. Como já mencionado, visto que o sistema anti gelo também não é considerado nesta análise a perda de sustentação da asa não é considerada para este estudo.

As superfícies híper sustentadoras incluem os flaps e todos os seus mecanismos de acionamento. Os flaps já se encontram explicados no sistema de controlo de voo da aeronave e, por ser um sistema mecânico, a condição de falha considerada é a falha do sistema elétrico.

4.2 Análise Qualitativa

A análise qualitativa da aeronave serve como base para o desenvolvimento do programa de análise de fiabilidade, apresentado no subcapítulo seguinte. Esta análise é realizada através de uma árvore de falhas, que estrutura a arquitetura da aeronave e dos seus sistemas, permitindo uma melhor compreensão das interdependências entre eles e dos respetivos modos de falha.

4.2.1 Definição do evento topo e restrições

Antes de iniciar a análise através de uma árvore de falhas, é essencial definir parâmetros iniciais que orientarão o processo. É necessário definir o evento topo da árvore da falhas e restrições sobre as quais a análise é realizada.

4.2.1.1 Evento topo

O evento topo da árvore de falhas, definido para esta análise como a "falha catastrófica da aeronave", representa uma situação em que ocorre uma perda completa e irreversível da capacidade da aeronave em operar de forma segura.

Este evento resulta de uma cadeia de falhas em um ou mais sistemas críticos da aeronave, identificados no capítulo 4.1. A falha catastrófica implica que a arquitetura da árvore e

redundâncias construídas falharam ou foram insuficientes, conduzindo a um desfecho que inclui a perda da aeronave e de vidas humanas.

4.2.1.2 Restrições

A construção da árvore de falhas foi efetuada com algumas restrições e limites. Em primeiro lugar, optou-se por excluir da análise os componentes suscetíveis a fadiga, assumindo que os efeitos deste fenómeno só se tornarão relevantes numa fase muito avançada do ciclo de vida da aeronave. Considera-se que, durante a vida operacional prevista da aeronave, os ciclos acumulados de fadiga não serão significativos o suficiente para comprometer a fiabilidade dos sistemas analisados.

Adicionalmente, assume-se que a falha catastrófica da aeronave não resulta de fatores como corrosão, dado que que a sua deteção e correção se fará no contexto do programa de manutenção e prevenção da corrosão, ficando sempre abaixo dos limites que pudessem introduzir falência.

De igual forma, e tal como referido acima, não são considerados os sistemas mecânicos nesta análise, pois as ações de manutenção preventiva previstas garantirão a mitigação dos riscos associados à deterioração dos seus componentes.

Relativamente ao sistema de anti gelo, a sua análise foi também excluída.

A formação de gelo nas superfícies aerodinâmicas depende de condições meteorológicas específicas, como baixa temperatura, elevada humidade e a presença de superfícies frias na aeronave. Tais condições consideram-se que não ocorrer durante grande parte das operações, porque se assume que serão seguidas as limitações e procedimentos definidos nos manuais de voo[18].

Fatores humanos, tais como falhas na segurança operacional, erros de pilotagem ou a instalação de peças defeituosas, não são considerados na análise. Estes fatores são excluídos porque o foco está exclusivamente nos modos de falha internos dos sistemas eletrónicos. Esta abordagem garante uma delimitação clara do âmbito da análise, concentrando-se apenas em falhas relacionadas diretamente com os sistemas da aeronave.

Neste contexto, importa contextualizar "perda" dando conta que se recorre a este termo quando um sistema ou subsistema deixa de desempenhar parcial ou totalmente a função para a qual foi projetado. Por outro lado, o termo "falha" refere-se quando um componente sofre desgaste, fratura ou deixa de cumprir a função para a qual foi concebido.

[18] Estes manuais descrevem as condições de operação da aeronave e, neste caso, podem incluir restrições que evitam a entrada em ambientes potencialmente suscetíveis à formação de gelo.

4.2.2 Árvore de falhas

Para a construção da árvore de falhas, começou-se por identificar as causas potenciais que podem resultar no evento de topo. Após a enumeração dessas possíveis causas, a árvore de falhas correspondente ao evento em análise foi representada graficamente. Esta representação gráfica segue a simbologia descrita na Tabela 4.

1. Falha catastrófica da aeronave

As razões de podem levar à falha catastrófica da aeronave são:

- Perda do sistema elétrico;
- Perda de propulsão;
- Perda dos aviónicos;
- Perda do FMS;
- Perda de sustentação.

A Figura 19 ilustra a árvore de falhas para a falha catastrófica da aeronave.

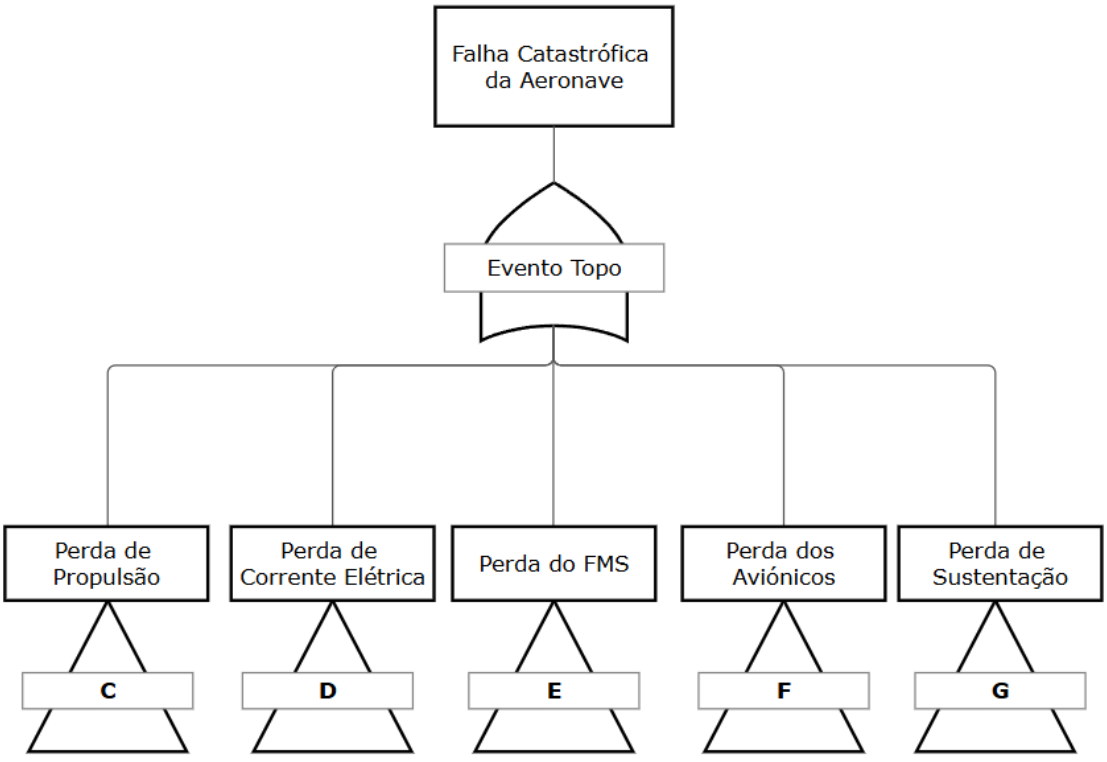


Figura 19 – Árvore de falha catastrófica da aeronave

Fonte: (autor, 2025)

A perda do sistema propulsivo ocorre devido à perda de ambos motores da aeronave. As razões que podem levar à perda do motor da aeronave são:

- Falha intrínseca do motor;
- Perda do sistema de combustível (falha conjunta das bombas de combustível);

A Figura 20 ilustra a árvore de falhas para a perda de propulsão da aeronave.

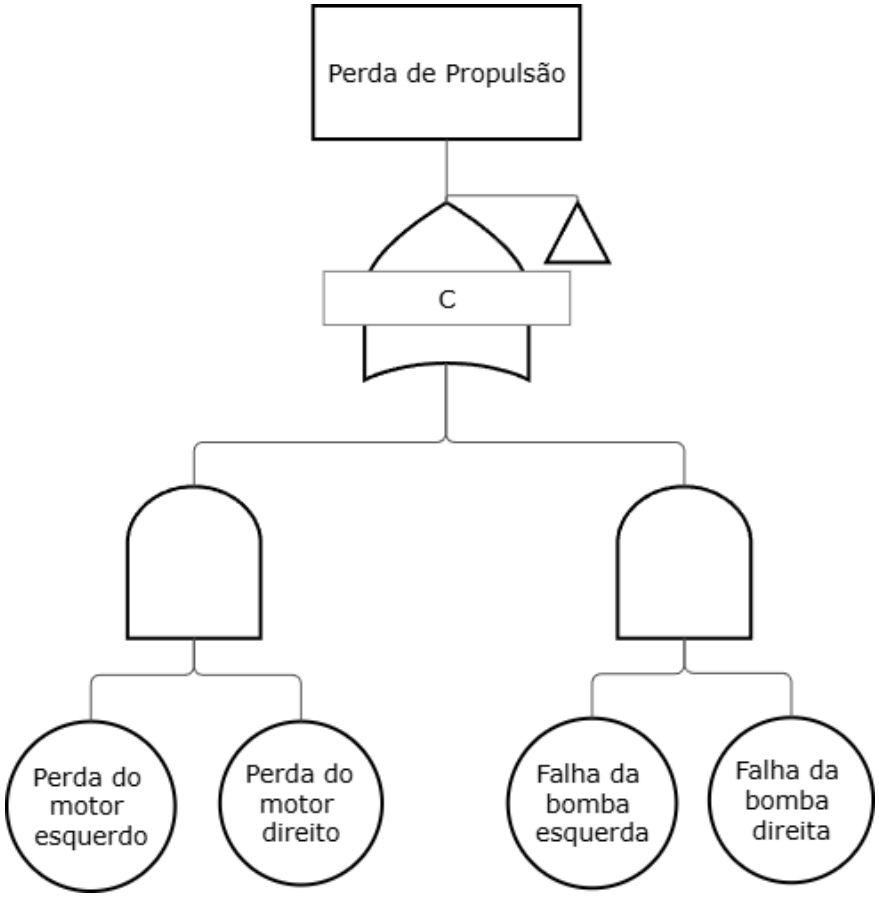


Figura 20 – Árvore de perda de propulsão da aeronave
Fonte: (autor, 2025)

As razões que levam à perda total de corrente elétrica da aeronave são as seguintes:

- Falha conjunta dos alternadores;
- Perda do sistema propulsivo;

A Figura 21 ilustra a árvore de falhas para a perda do sistema elétrico da aeronave.

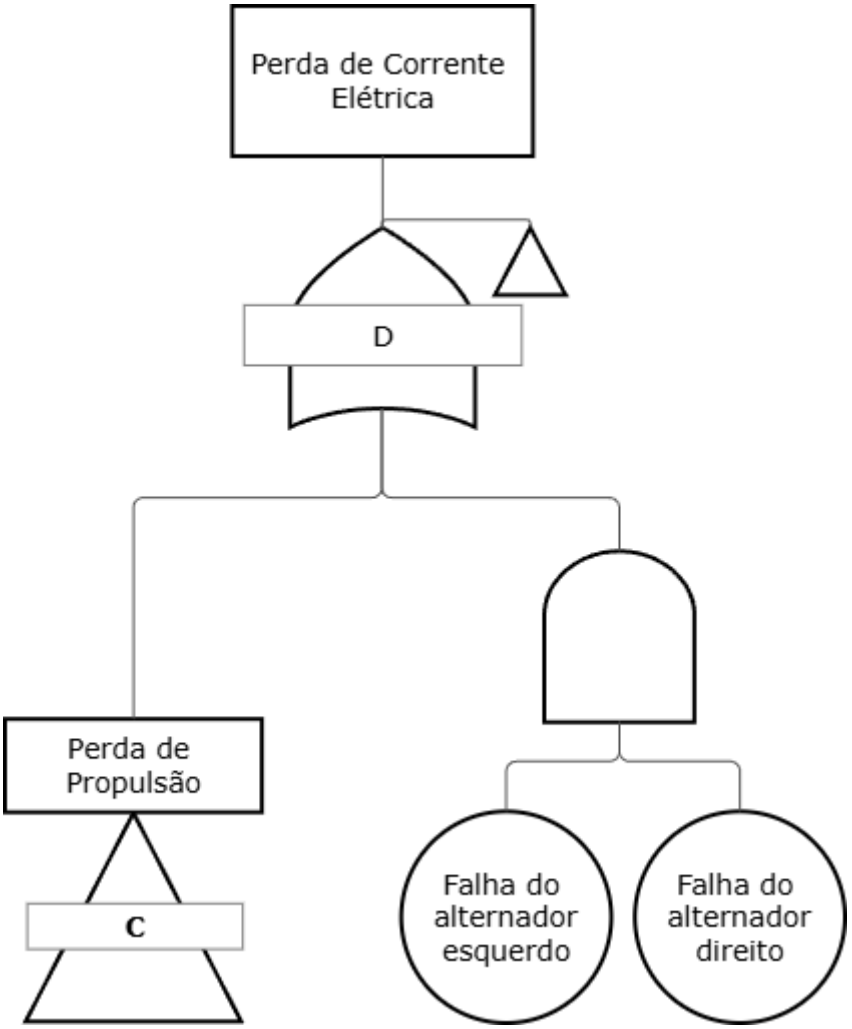


Figura 21 – Arvore de falha do sistema elétrico da aeronave

Fonte: (autor, 2025)

As razões que podem levar à perda dos aviónicos da aeronave são:

- Perda do sistema de navegação;
- Perda do sistema elétrico.

A Figura 22 ilustra a árvore de falhas para a perda dos aviónicos da aeronave.

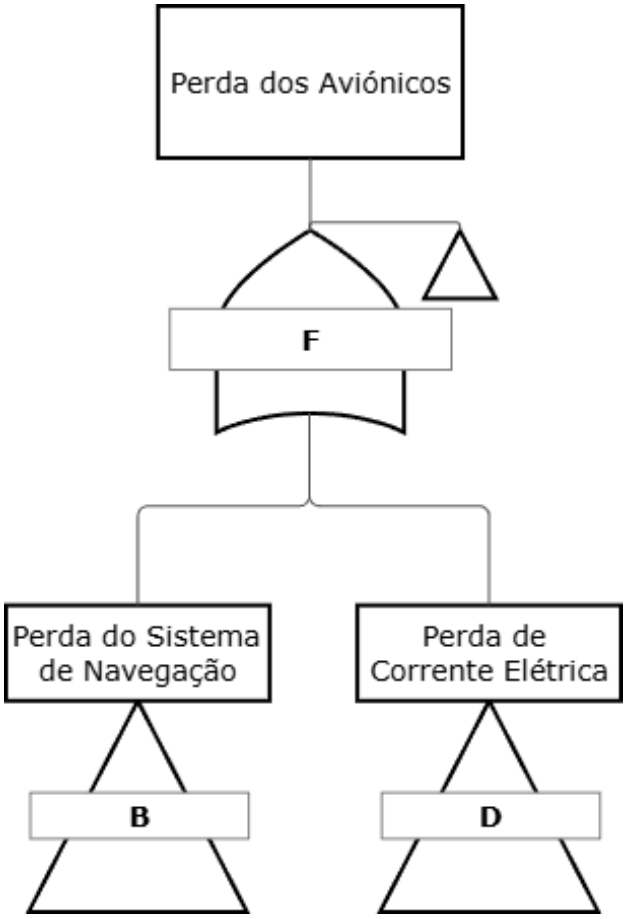


Figura 22 – Árvore de perda dos aviónicos da aeronave

Fonte: (autor, 2025)

As razões que podem levar à perda do FMS da aeronave são:

- Falha conjunta dos MCDU's e dos FMGEC
- Perda do sistema elétrico;

A Figura 23 ilustra a árvore de falhas para a perda do FMS da aeronave.

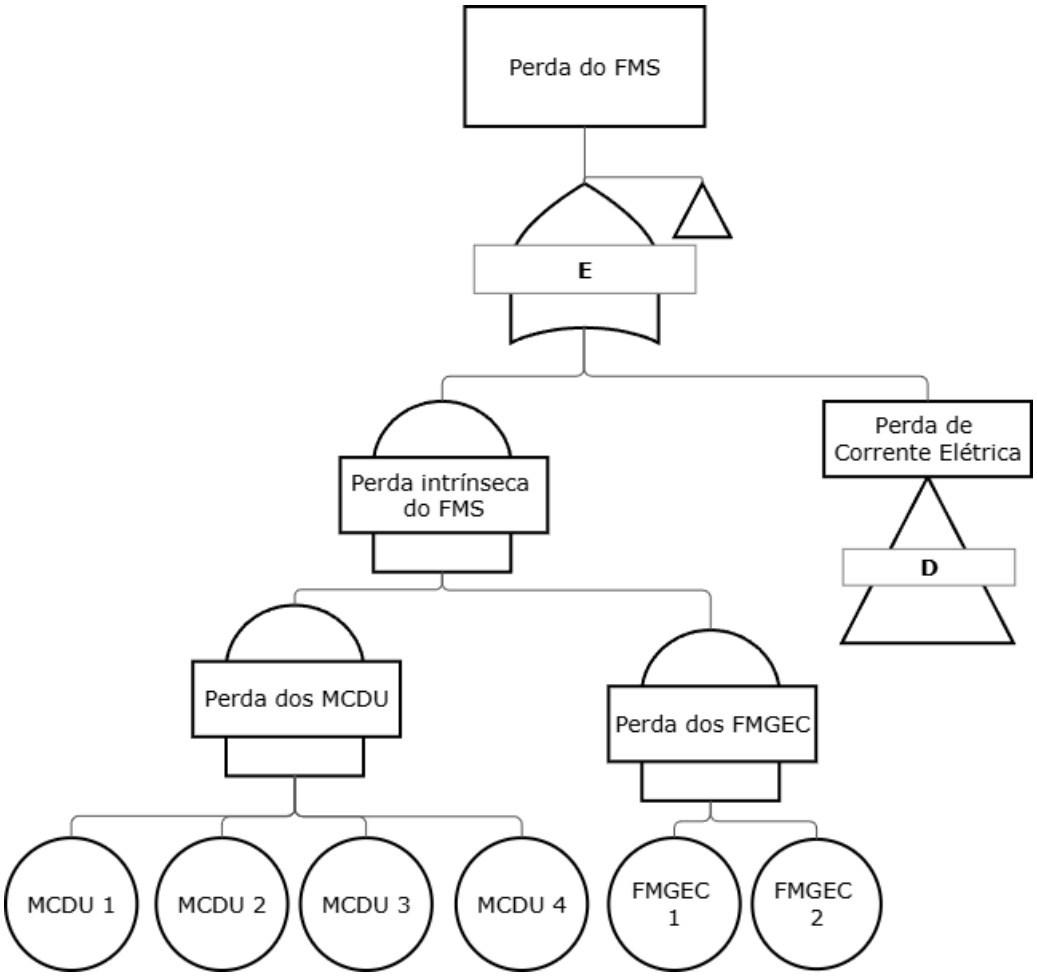


Figura 23 – Árvore de perda do sistema FMS da aeronave

Fonte: (autor, 2025)

As razões que podem levar à perda de sustentação da aeronave são:

- Perda do sistema propulsivo;
- Falha dos flaps (perda do sistema elétrico);

A Figura 24 ilustra a árvore de falhas para a perda do FMS da aeronave.

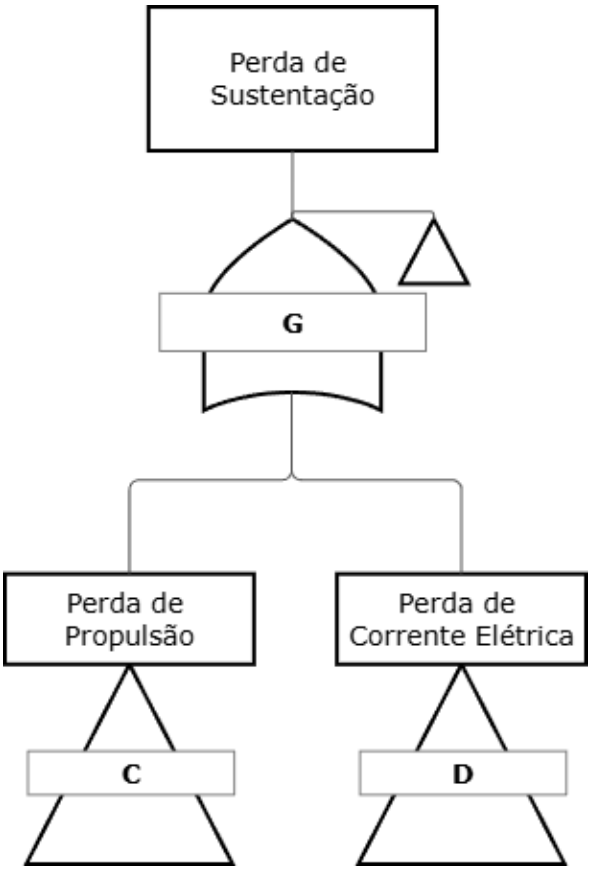


Figura 24 – Árvore de perda de sustentação da aeronave

Fonte: (autor, 2025)

As razões que podem levar à perda de navegação são:

- Falha dos recetores VOR;
- Falha dos recetores GNSS;

A Figura 25 ilustra a árvore de falhas para a perda de navegação da aeronave.

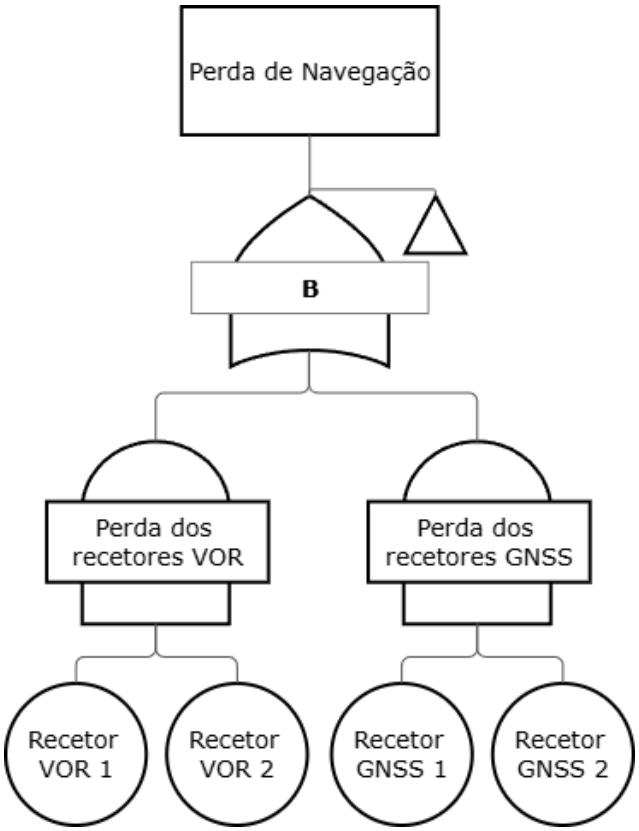


Figura 25 – Árvore de perda de navegação da aeronave

Fonte: (autor, 2025)

4.3 Análise quantitativa

Uma vez modeladas as condições de falhas consideradas na árvore de falhas, o passo seguinte consiste em realizar a análise quantitativa da árvore através do programa de análise de fiabilidade desenvolvido neste trabalho.

O programa, além de calcular a probabilidade de ocorrência do evento topo da árvore de falhas, também apresenta outras funcionalidades:

- Realização de uma análise de importância e de sensibilidade para todos os eventos de falha;
- Cálculo do valor de MTBF de todos os sistemas construídos a partir dos dados de falha individuais de cada componente;
- Realização de uma análise temporal de todos os sistemas/subsistemas/componentes incluídos na análise bem como da aeronave como um todo de modo a estudar o comportamento dos sistemas ao longo do tempo com a inclusão de limites de manutenção.

Este programa é uma ferramenta projetada para analisar a fiabilidade de qualquer sistema ou subsistema de uma aeronave. Embora, em contexto com este trabalho, a aplicação esteja a ser utilizada para avaliar sistemas críticos do LUS-222, a sua arquitetura permite adaptações e expansões para outros componentes ou sistemas da aeronave.

O programa foi concebido para ser acessível a qualquer utilizador, independentemente do nível de experiência prévia no tema da fiabilidade ou em ferramentas computacionais. Além de ser aberto a todos, a sua estrutura e a utilização de ferramentas amplamente conhecidas facilitam a sua adoção e implementação em diferentes contextos.

Para executar todas as funções do programa, é necessário que o utilizador disponha de duas ferramentas: Microsoft Excel e um programa de execução compatível com *python*[19].

A construção dos sistemas, a introdução dos dados de falha, a definição das portas lógicas e a elaboração da árvore de falhas da aeronave são realizadas diretamente no Excel. Esta escolha baseia-se não só na facilidade de utilização da aplicação, mas também no facto da realização de uma arvore de falhas ser um processo muito "manual" e "visual" [20] permitindo que os utilizadores compreendam de forma clara as relações entre os eventos de falha.

As análises mais complexas, como a avaliação de importância e sensibilidade dos eventos básicos, o cálculo do MTBF e a análise temporal são realizadas em Python. Esta linguagem de programação foi escolhida devido à sua capacidade de lidar com cálculos matemáticos avançados, que excedem as capacidades do Excel.

O diagrama da figura 26 ilustra de forma visual e clara todo o processo de análise de fiabilidade desenvolvido.

[19] Inclui qualquer ambiente que seja compatível com python. Quer seja ter o python instalado no sistema operativo ou ter alguma IDE instalada (PyCharm, Jupyter Notebook, entre outras).

[20] Uma FTA é um processo no qual se desenha manualmente um diagrama que representa visualmente as possíveis causas e interconexões de falhas em um sistema. Não é um processo totalmente automatizado, depende do julgamento e conhecimento do profissional.

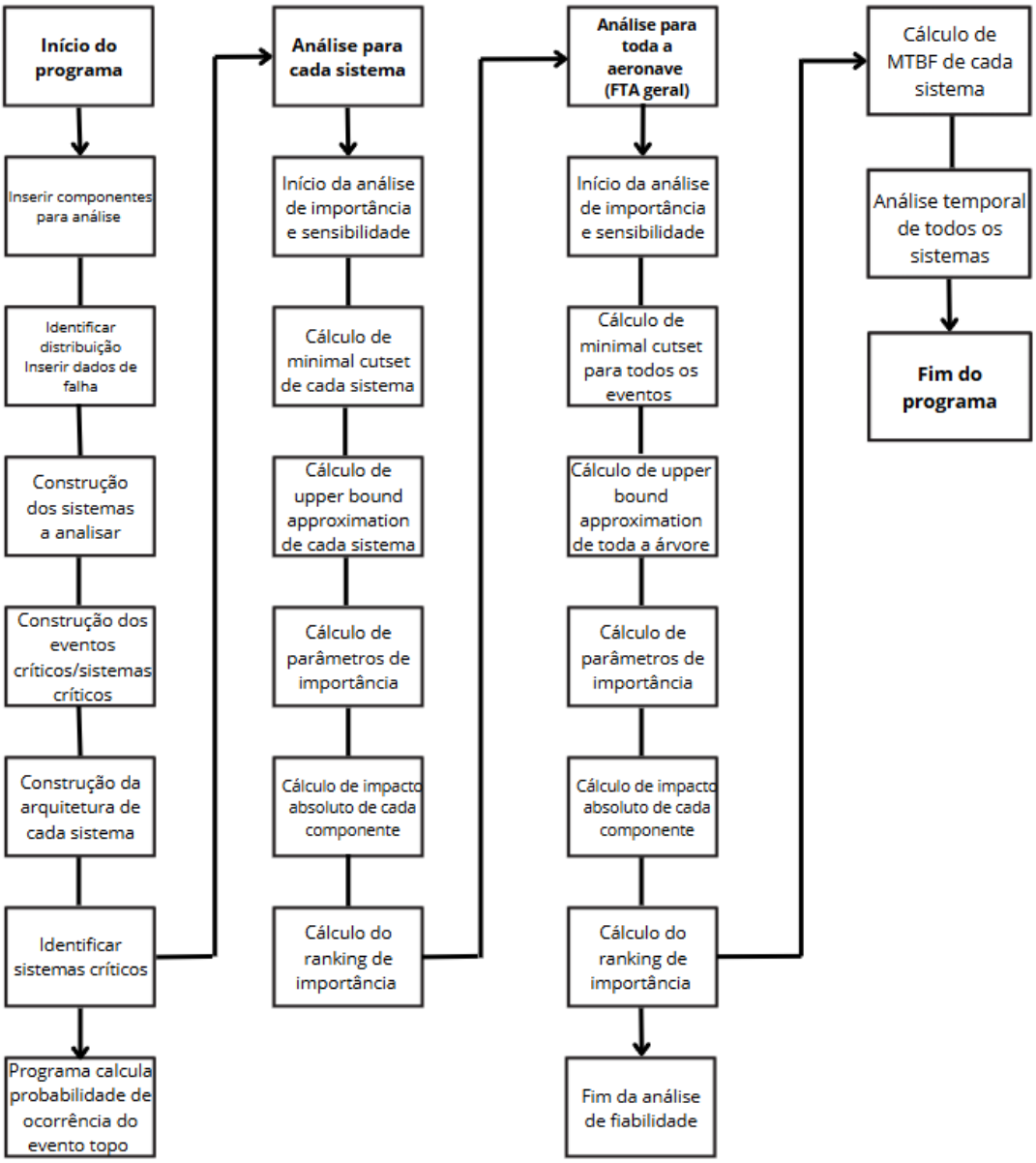


Figura 26 - Esquema de funcionamento do programa de análise de fiabilidade

Fonte: (autor, 2025)

O programa começa com a introdução dos componentes que serão utilizados na análise de fiabilidade.

A folha de excel denominada de “Base de Dados” representa a interface da introdução de todos os componentes a analisar.

MIL-STD-1808C	
System/Title	Subsystem
21 AIR CONDITIONING	21-00 GENERAL
	21-10 COMPRESSION
	21-20 DISTRIBUTION
	21-30 PRESSURIZATION CONTROL
	21-40 HEATING
	21-50 COOLING
	21-60 TEMPERATURE CONTROL
	21-70 MOISTURE.AIR CONTAMINANT CONTROL
	21-80 EQUIPMENT COOLING
	21-90 LIQUID COOLING
22 AUTO FLIGHT	22-00 GENERAL
	22-10 AUTOPILOT
	22-20 SPEED-ALTITUDE CORRECTION
	22-30 AUTO THROTTLE
	22-40 SYSTEM MONITOR
	22-50 AERODYNAMIC LOAD ALLEVIATING
23 COMMUNICATION	23-00 GENERAL
	23-10 LOW/VERY LOW FREQUENCY (LF/VHF)
	23-20 HIGH/VERY HIGH FREQUENCY (HF/VHF)
	23-30 ULTRA/SUPER/EXTREMELY HIGH FREQUENCY (UHF/SHF/EHF)
	23-40 PASSENGER ADDRESS/INTERPHONE
	23-50 AUDIO INTEGRATING
	23-60 STATIC DISCHARGING
	23-70 AUDIO AND VIDEO MONITORING
	23-80 INTEGRATED AUTOMATIC TUNING

Figura 27 - Lista de sistemas utilizados em voo (Norma MIL-STD-1808C)

Fonte: (autor, 2025)

Esta folha começa por apresentar todos os sistemas que a aeronave utiliza durante o voo, quer estes sejam críticos ou não. Todos estes sistemas estão descritos através do documento MIL-STD-1808C, com os seus respetivos subsistemas ao lado, como ilustrado na Figura 27

O utilizador deve identificar o subsistema ao qual o componente que pretende analisar pertence e introduzi-lo na lista “Componentes”. A descrição do componente deve seguir as regras de classificação do documento MIL-STD-1808C apresentadas no capítulo 3.4.4.

Se um dos componentes a analisar for, por exemplo, um alternador, este como pertence ao sistema elétrico (24) e ao subsistema *ac generation* (24-20) então o componente é introduzido como “24-20-01 Alternador”.

Após a introdução de todos os componentes pretendidos, o passo seguinte consiste em introduzir, a cada um deles, os respetivos dados de falha, isto é realizado na folha “Dados de Fiabilidade”. Este processo baseia-se numa sequência que inclui a identificação do componente, a definição do tipo de distribuição de falha e a introdução dos parâmetros associados.

O processo de análise de fiabilidade é realizado a partir de um tempo de operação específico, onde é possível alterar o seu valor na folha de excel (célula D2).

A identificação de cada componente é feita utilizando as três primeiras colunas da interface de introdução de dados, ilustrado na Figura 28, que correspondem às seguintes categorias:

- **Sistema:** Nome do sistema principal ao qual o componente pertence (por exemplo, Sistema de Navegação);
- **Subsistema:** Identificação do subsistema que integra o sistema principal (por exemplo, FMS);
- **Componente:** Nome específico do componente cujo desempenho será analisado (por exemplo, MCDU). No caso de o utilizador não ter dados de componentes específico, mas sim de algum subsistema, a coluna componente deve ser deixada vazia.

A Figura 28 ilustra um exemplo de introdução de diversos componentes.

Sistema	Subsistema	Componente
24 ELECTRICAL POWER	24-20 AC GENERATION	24-20-01 Alternador
30 ICE AND RAIN PROTECTION	30-10 AIRFOIL	
34 NAVIGATION	34-60 FLIGHT MANAGEMENT COMPUTING	34-60-01 MCDU
34 NAVIGATION	34-60 FLIGHT MANAGEMENT COMPUTING	34-60-02 FMGEC

Figura 28 - Interface de identificação de cada componente

Fonte: (autor, 2025)

Após identificar o componente, é necessário definir o tipo de distribuição de falha que o mesmo segue e introduzir os dados dos parâmetros afetos a essa distribuição. O programa suporta as seguintes opções:

- **Distribuição Exponencial**: Requer apenas o parâmetro MTBF (horas);
- **Distribuição de Weibull:** Requer os parâmetros ß e θ (horas).

	Distribuição Exponencial		Distribuição de Weibull			
Distribuição	MTBF [h]	λ	β	θ	R(t)	F(t)
Weibull			1,58	4500	0,999978507	2,14928E-05
Exponencial	20000	0,00005			0,999750031	0,000249969
Weibull			1,2	4820	0,999737548	0,000262452

Figura 29 - Interface de introdução de dados de falha

Fonte: (autor, 2025)

Uma vez introduzidos os dados de falha, como ilustrado na Figura 29, o programa calcula a fiabilidade e probabilidade de falha de cada componente de acordo com o tempo de operação definido.

O próximo passo consiste na construção dos sistemas e dos eventos de falha, processo que ocorre na folha “Árvore de Falhas”. Na aba “Sistemas”, encontram-se as tabelas que são utilizadas para introduzir os componentes de cada sistema a analisar.

Sistema A	Evento Básico	Definição	F(t)
34-60 FLIGHT MANAGEMENT COMPUTING	34-60-01 MCDU	A1	0,000262452
	34-60-01 MCDU	A2	0,000262452
	34-60-01 MCDU	A3	0,000262452
	34-60-01 MCDU	A4	0,000262452
	34-60-02 FMGEC	A5	4,99988E-05
	34-60-02 FMGEC	A6	4,99988E-05

Figura 30 - Componentes do Sistema A (FMS)

Fonte: (autor, 2025)

A Figura 30 ilustra uma destas tabelas. O utilizador deve introduzir os componentes[21] que pertencem ao sistema, de acordo com as redundâncias que serão consideradas. Por exemplo, no sistema FMS, como será considerado uma redundância quadrupla dos MCDU e dupla redundância para os FMGEC, são introduzidos 4 MCDU e 2 FMGEC, como ilustrado na figura acima.

A definição de cada evento básico é dos pontos mais importantes para o programa pois esta permite seguir de forma ordenada cada evento básico para as restantes análises do programa (importância e sensibilidade). Cada tabela tem um nome, no caso da tabela da figura acima, esta chama-se “Sistema A”, e o programa contém tabelas que vão até ao “Sistema N”, portanto, 14 tabelas disponíveis para realizar a análise.

Dentro de cada sistema (tabela) a definição dos eventos básicos é realizada a partir do nome do sistema, acoplado de um número para cada componente do sistema, tal como ilustrado na Figura 30. Para realizar a definição dos eventos básicos, o utilizador deve primeiro definir todos os sistemas que se pretendem analisar e depois pressionar um botão “Definir Eventos Básicos” que ativa uma macro que realiza a definição automática de todos os eventos básicos incluídos.

Para construir as interdependências entre sistemas, visto que cada sistema construído tem uma definição de A até N o utilizador deve, manualmente, introduzir a definição do sistema na coluna “Definição”. Por exemplo, para construir a perda do FMS, este contém interdependências do próprio FMS (sistema A) e do sistema elétrico (sistema D), visto que a macro não coloca definições de sistemas anteriormente construídos, é necessário inserir as definições correspondentes aos sistemas inseridos, neste caso, A e D, tal como ilustrado na figura 31.

[21] A partir deste ponto, qualquer componente ou subsistema é considerado como um evento básico para a árvore de falha.

Sistema E	Evento Básico	Definição	F(t)
FMS	34-60 FLIGHT MANAGEMENT COMPUTING	A	2,84412E-27
	24 ELECTRICAL POWER	D	6,94027E-10

Figura 31 - Construção para sistemas com interdependências

Fonte: (autor, 2025)

Uma vez inseridos os componentes de cada sistema, o passo seguinte consiste em realizar a construção dos sistemas através de uma arvore de falhas, com o uso de portas lógicas.

Sistema A	Quantidade Eventos Intermédios	Evento	Porta Lógica
34-60 FLIGHT MANAGEMENT COMPUTING	2	Intermédio 1	AND
		Intermédio 2	AND
		Evento Topo	AND

Figura 32 - Interface de construção do sistema FMS

Fonte: (autor, 2025)

A figura 32 ilustra uma parte do processo de construção do sistema FMS, onde o utilizador deve identificar a quantidade de eventos intermédios na arvore do sistema e para cada evento, a porta lógica correspondente.

Evento	Porta Lógica	A1	A2	A3	A4	A5	A6	Intermédio 1	Intermédio 2	Evento Topo
Intermédio 1	AND	1	1	1	1					
Intermédio 2	AND					1	1			
Evento Topo	AND							1	1	

Figura 33 - Interface de construção do sistema FMS (2)

Fonte: (autor, 2025)

Uma vez identificados os eventos intermédios e as respetivas portas lógicas, o utilizador deve introduzir os componentes que estão incluídos para cada evento. Como por exemplo, para o sistema FMS, ilustrado na figura 33, o evento intermedio 1 representa a perda simultânea dos MCDU, que apresentam quadrupla redundância, portanto, para esse evento, os eventos A1 a A4 (que representam os MCDU) encontram-se assinalados com "1".

O evento intermédio 2, que representa a perda simultânea dos FMGEC, contém os eventos A5 e A6.

O evento topo, que representa a perda simultânea dos eventos intermédios, inclui os eventos intermédios 1 e 2, ligados através de uma porta lógica AND.

Os eventos básicos devem ser sempre assinalados por "1", pois isto indica a existência desse evento dentro do evento intermédio.

A redundância dos eventos básicos é definida na parte da definição dos componentes. É possível que o utilizador, na aba "Sistemas", inclua apenas um evento básico e depois na construção do sistema coloque um valor maior que "1" indicando o número redundância desse componente no sistema. O programa está construído para que também funcione dessa maneira, e irá devolver a mesma probabilidade de falha do sistema.

No entanto, realizar a construção dos sistemas a partir deste método não é viável para as análises de importância e de sensibilidade, porque, por exemplo, ao definir o componente MCDU como A1 e indicando "4" no evento intermédio 1 (indicando quádrupla redundância) isto diz ao programa que os MCDU's são dependentes uns dos outros (visto que são todos definidos como A1) portanto, ao falhar A1, o programa considera que todos os componentes definidos por A1 também falham resultando numa falha total do evento intermédio 1.

Ao definir cada MCDU como um evento básico independente (A1, A2, A3 e A4), se falhar por exemplo apenas A1, o programa reconhece que os restantes eventos (MCDU's) ainda se encontram ativos e não falharam.

Evento/Sistema	34-60 FLIGHT MANAGEMENT COMPUTING
Intermédio 1	4,74463E-15
Intermédio 2	2,49988E-09
Evento Topo	1,1861E-23

Figura 34 - Probabilidade de ocorrência dos eventos intermédios e do evento topo

Fonte: (autor, 2025)

Após realizada a construção de cada sistema, o programa calcula e apresenta a probabilidade de ocorrência dos eventos intermédios bem como do evento topo, tal como ilustrado na figura 34.

A próxima folha "Prob de Falha Catastrófica" contém a interface onde o programa realiza o cálculo da probabilidade de falha catastrófica da aeronave.

Probabilidade de Falha dos Sistemas Calculados		
Sistemas	Sistemas Calculados	Probabilidade de Falha, F(t)
A	34-60 FLIGHT MANAGEMENT COMPUTING	
B	VOR	

Figura 35 - Resumo da probabilidade de falha dos sistemas calculados

Fonte: (autor, 2025)

A figura 35 representa a interface que contém o resumo da probabilidade de falha de todos os sistemas construídos na folha anterior.

O passo seguinte consiste em escolher quais destes sistemas são considerados críticos, ou seja, aqueles que se encontram imediatamente abaixo do evento topo na árvore de falhas da aeronave (falha catastrófica).

Para o LUS-222 estes sistemas já foram esclarecidos nos capítulos passados e encontram-se ilustrados na figura 36.

Escolha dos Sistemas Críticos		
Sistemas Enumerados	Sistemas Críticos	F(t)
E	FMS	6,94027E-10
C	72.1 ENGINE	5,89282E-10
G	SUPERFICIES SUSTENTADORAS	1,28331E-09
F	AVIONICOS	4,30498E-09
D	24 ELECTRICAL POWER	6,94027E-10

Figura 36 - Escolha dos sistemas críticos

Fonte: (autor, 2025)

Uma vez escolhidos os sistemas críticos da aeronave, o programa realiza o cálculo da probabilidade de falha catastrófica da mesma e realiza a comparação com o objetivo de fiabilidade imposto pelas regulamentações (10^{-9}), ilustrado na figura 37.

Probabilidade de Falha Catastrófica da Aeronave		
Probabilidade FC	Objetivo de Fiabilidade, [F(t)/fh]	Cumprimento do Objetivo
	0,000000001	

Figura 37 - Cálculo da probabilidade de falha catastrófica da aeronave

Fonte: (autor, 2025)

Ao chegar a este ponto, a construção da árvore de falhas da aeronave encontra-se finalizada. A partir daqui, é necessário o uso de programas executáveis em python que realizam as análises de importância e de sensibilidade dos eventos básicos.

O programa é acompanhado por quatro ficheiros executáveis, chamados de “Programa Análise Sensibilidade.exe”, “MTBF.exe”, “Análise Temporal.exe” e “Analise Temporal Aeronave.exe”.

Antes de executar qualquer um destes ficheiros, o utilizador deve guardar e sair da aplicação excel e então executar o ficheiro pretendido.

No caso de não ser possível executar os ficheiros diretamente através do executável, o utilizador pode correr o programa a partir de uma IDE com o código disponibilizado na pasta do programa.

Na folha "Minimal Cut Set", o programa calcula todas as expressões booleanas de todos os eventos para cada sistema calculado.

O utilizador deve, para cada sistema, pressionar o botão correspondente "Expandir Expressões" que ativa uma macro que transforma as expressões de cada evento na expressão booleana expandida correspondente. No caso da figura 38, para o evento topo, ambos eventos intermédios estão ligados por uma porta AND (símbolo de multiplicação) e a macro irá substituir os termos “Intermedio 1” e “Intermedio 2” pelas suas expressões correspondentes.

Importante referir que, antes de pressionar o botão "Expandir Expressões", o utilizador deve verificar que deve estar selecionada uma célula qualquer que se encontre dentro da terceira coluna da tabela da figura abaixo, para o sistema em análise.

Sistema A		
34-60 FLIGHT MANAGEMENT COMPUTING	Expressão de Cada Evento	Expressão de Cada Evento
Intermédio 1	A1*A2*A3*A4	A1*A2*A3*A4
Intermédio 2	A5*A6	A5*A6
Evento Topo	Intermédio 1*Intermédio 2	(A1*A2*A3*A4)*(A5*A6)

Figura 38 - Expressões booleanas para cada evento do sistema A

Fonte: (autor, 2025)

Com as expressões para todos os sistemas obtidas o passo seguinte consiste em salvar e fechar o ficheiro excel e executar o programa "Programa Análise Sensibilidade.exe" que é constituído por três programas distintos. O primeiro chama-se "Minimal Cutset Sistemas.exe" cuja função é simplificar as expressões do evento topo obtidas da figura 38 de acordo com as propriedades da álgebra booleana e expandir as mesmas até à sua forma mais simplificada[22] de modo a poder encontrar os *minimal cutsets* de cada sistema.

O programa realiza estas tarefas a partir da seguinte função no código:

```
    def simplify_boolean_expression(expression):

        if not expression:
            return ""
        try:
            expression = expression.replace(" ", "")
            expression = expression.replace("*", "&").replace("+", "|")

            vars_in_expr = extract_variables(expression)
            sym_dict = {var: symbols(var) for var in vars_in_expr}
            expr_with_symbols = expression

            for var in vars_in_expr:
                expr_with_symbols = expr_with_symbols.replace(var,
f"symbols('{var}')")

            expr_obj = eval(expr_with_symbols, {"symbols": symbols, "Or": Or,
"And": And})

            simplified_expr = sympy.simplify_logic(expr_obj)

            expanded_expr = distribute_or_over_and(simplified_expr)

            expanded_str = str(expanded_expr)
            expanded_str = expanded_str.replace("&", "*").replace("|", "+")
            expanded_str = remove_final_parentheses(expanded_str)
            print(expanded_str)

            return expanded_str
```

[22] Isto significa que a expressão resultante será dada na forma genérica *ab+cd+e*, com os *minimal cutsets* sempre separados por sinais de adição.

```
        except Exception as e:
            return f"Erro: {str(e)}"
```

Uma vez que o cálculo dos *minimal cutsets* é realizado, o passo seguinte do executável "Programa Análise Sensibilidade.exe" consiste em correr o *script* "Análise Sensibilidade Sistemas.exe". Este tem como função calcular todos os parâmetros de importância bem como variações da probabilidade de falha do sistema de acordo com variações percentuais de cada evento básico. Todos os resultados são devolvidos ao excel na folha "Análise Sensibilidade".

Sistema A	Cut Set do Sistema	Eventos	Probabilidade de Falha
34-60 FLIGHT MANAGEMENT COMPUTING	1 - (1 - A1 * A2 * A3 * A4 * A5 * A6)	A1	0,000262452
		A2	0,000262452
		A3	0,000262452
		A4	0,000262452
		A5	4,99988E-05
		A6	4,99988E-05

Figura 39 - Interface eventos básicos do sistema A

Fonte: (autor, 2025)

A figura 39 ilustra uma parte da interface na qual serão apresentados todos os resultados resultantes das análises realizadas pelo programa. A coluna "Cut Set do Sistema" contém a expressão da aproximação de *upper bound,* e, a partir desta expressão, são calculados os parâmetros de importância, ilustrados na figura 40.

BIR	BIR (Valores)	CRIT	Minimal Cut Set do Sistema	MCS	F(MCSi)
A2*A3*A4*A5*A6	7,80469E-18	172,6973887	A1 * A2 * A3 * A4 * A5 * A6	A1 * A2 * A3 * A4 * A5 * A6	3,90137E-21
A1*A3*A4*A5*A6	7,80469E-18	172,6973887			
A1*A2*A4*A5*A6	7,80469E-18	172,6973887			
A1*A2*A3*A5*A6	7,80469E-18	172,6973887			
A1*A2*A3*A4*A6	1,56074E-17	65,79154238			
A1*A2*A3*A4*A5	1,56074E-17	65,79154238			

Figura 40 - Parâmetros de importância do sistema A (1)

Fonte: (autor, 2025)

A coluna BIR e BIR (Valores) apresentam os dados simbólicos e numéricos dos parâmetro de importância de *birnbaum* e a coluna CRIT representa o parâmetro de importância de criticidade.

A coluna “Minimal Cut Set do Sistema” indica a expressão booleana do sistema expandida e simplificada (resultado do primeiro *script*) e a coluna MCS apresenta todos os minimal cut sets separados. A coluna F(MCS_i) contém a probabilidade de falha de cada *minimal cutset* da coluna MCS.

Fussell-Vesely	RRW	RAW
328,9248186	∞	2000,500042
328,9248186	∞	2000,500042
328,9248186	∞	2000,500042
328,9248186	∞	2000,500042
328,9248186	∞	4000,500021
328,9248186	∞	4000,500021
0		
0		
0		
0		
0		
0		
0		
0		
0		

Figura 41 - Parâmetros de importância do sistema A (2)
Fonte: (autor, 2025)

A figura 41 contém os restantes parâmetros de importância calculados pelo programa (Fussell-Vesely, RRW e RAW).

O cálculo do parâmetro de *birnbaum* simbólico e numérico é realizado através das seguintes linhas de código:

```
    for i, event in enumerate(event_list):
                    try:
                        derivative = sp.diff(expr_obj, sym_dict[event])
                        derivate_list.append(derivative)

                        numeric_derivative = derivative
                        for e, prob in event_prob_dict.items():
                            numeric_derivative=
numeric_derivative.subs(sym_dict[e], prob)

                        numeric_result = float(numeric_derivative.evalf())
                        numeric_results.append(numeric_result)
```

O parâmetro de criticidade é calculado pelo próprio excel, bem como o parâmetro de Fussell-Vesely.

Os parâmetros RRW e RAW são calculados através da seguinte função:

```
    def calculate_component_measures(expression, event_prob_dict, sym_dict,
event):
        base_prob=calculate_system_probability(expression,  event_prob_dict,
sym_dict)

        rrw_dict = event_prob_dict.copy()
        rrw_dict[event] = 0
        prob_with_zero = calculate_system_probability(expression, rrw_dict,
sym_dict)

        raw_dict = event_prob_dict.copy()
        raw_dict[event] = 1
        prob_with_one = calculate_system_probability(expression, raw_dict,
sym_dict)

        rrw = float('inf') if prob_with_zero == 0 else base_prob /
prob_with_zero
        raw = prob_with_one / base_prob

        return rrw, raw
```

Desta forma, o cálculo dos parâmetros para a analise de importância dos sistemas é finalizado e o *script* segue para o cálculo do impacto absoluto a partir de diferentes variações percentuais de cada evento básico (-50%, -25%, 25%, 50%), conforme ilustrado na figura 42.

Variação Percentual de Probabilidade de Falha dos Componentes

-25%	Impacto Absoluto (-25%)	25%	Impacto Absoluto (25%)
0,000196839	0,25	0,000328066	0,25
0,000196839	0,25	0,000328066	0,25
0,000196839	0,25	0,000328066	0,25
0,000196839	0,25	0,000328066	0,25
3,74991E-05	0,25	6,24984E-05	0,25
3,74991E-05	0,25	6,24984E-05	0,25

Figura 42 - Análise de sensibilidade VS variações do evento básico do sistema A

Fonte: (autor, 2025)

O cálculo do impacto absoluto é realizado através da seguinte função:

```
    def calculate_absolute_impact(base_prob, varied_prob):
        return abs(varied_prob - base_prob) / base_prob
```

Uma vez calculado o impacto absoluto de cada sistema, o *script* “Analise Sensibilidade Sistemas.exe” é finalizado e o programa segue para o último *script* pertencente ao programa “Programa Analise Sensibilidade.exe” chamado “Analise Sensibilidade Aeronave.exe”. Este

realiza os mesmos cálculos para os mesmos parâmetros, no entanto estes não são realizados para cada sistema, mas sim para a árvore de falhas da aeronave como um todo.

A primeira tabela da folha "Analise Sensibilidade Aeronave" inclui o cálculo dos parâmetros através da aproximação de *upper bound* considerando apenas os sistemas críticos escolhidos pelo utilizador. A segunda tabela, para cada sistema crítico, considera todos os seus eventos básicos, obtendo uma análise mais detalhada para a importância de cada evento em toda a árvore da aeronave.

Após a execução do ficheiro "Programa Análise Sensibilidade.exe", o utilizador realiza a avaliação ou ranking de importância de cada evento básico de cada sistema ou de toda a aeronave.

A folha "Ranking Importância Sistemas" contém tabelas que apresentam os valores dos parâmetros de importância calculados para cada sistema. O utilizador, para a equação que define o ranking de importância, definida através da expressão $I_i = a \cdot BIR_i + b \cdot CRIT_i + c \cdot FV_i + d \cdot RRW_i + e \cdot RAW_i$ (35) deve indicar os pesos, em percentagem, para cada parâmetro, ilustrado na figura 43.

Parâmetros	a	b	c	d	e
Valor (%)	0	0	0	0	0

Figura 43 - Interface de dados de peso para parâmetros de importância
Fonte: (autor, 2025)

A folha "Ranking Importância Aeronave" funciona de forma a calcular o valor de importância de cada sistema crítico escolhido e, dentro de cada sistema critico, permite também calcular o ranking de importância de todos os eventos básicos que pertencem à árvore de falhas de toda a aeronave.

Uma vez chegado a este ponto, a analise de importância e de sensibilidade para os sistemas considerados e para a aeronave é concluída. Como referido acima, o utilizador dispõe de três mais executáveis que permitem recolher mais informações sobre o comportamento do sistema construído ao longo do tempo.

O executável "MTBF.exe" realiza o cálculo do MTBF de todos os sistemas construídos, sempre que estes contenham apenas componentes que sigam distribuição exponencial, como é o caso deste trabalho, uma vez que a fadiga não é considerada, a distribuição de Weibull não é utilizada.

Como explicado no capítulo 3.4.1, o cálculo do MTBF é realizado através do inverso da taxa de falha efetiva média, calculada através da expressão $\lambda_e(t_i) = \frac{-\frac{\Delta R}{\Delta t}}{R(t_i)}(41)$.

Através da folha "Interface MTBF", o programa calcula a fórmula da fiabilidade do sistema em função do tempo, considerando todos os componentes que compõem o sistema.

		Distribuição Exponencial	Distribuição de Weibull		
Eventos	Componente	MTBF [h]	β	θ	Cut Set Sistema
A1	34-60-01 MCDU	10000			(A1 & A2 & A3 & A4) & (A5 & A6)
A2	34-60-01 MCDU	10000			
A3	34-60-01 MCDU	10000			
A4	34-60-01 MCDU	10000			
A5	34-60-02 FMGEC	20000			
A6	34-60-02 FMGEC	20000			

Figura 44 - Interface do cálculo da fiabilidade em função do tempo do sistema A
Fonte: (autor, 2025)

O programa, através da expressão do cut set do sistema, analisa quais os eventos que estão ligados em série (|) ou em paralelo (&) e, para cada conexão, o programa calcula a fórmula de fiabilidade do sistema em função do tempo. Através do seguinte trecho de código, o programa começa por encontrar o operador principal, isto é, aquele imediatamente abaixo do evento topo, aquele que se encontra fora de todos os parênteses:

```
def find_operator(expr):
        depth = 0
        for i, char in enumerate(expr):
            if char == '(':
                depth += 1
            elif char == ')':
                depth -= 1
            elif depth == 0 and (char == '&' or char == '|'):
                return char, i
        return None, -1
```

Após descobrir o operador principal, o programa, a partir desse operador, separa a expressão entre lado esquerdo e lado direito.

A partir deste ponto, o programa começa a trabalhar com o lado esquerdo do cutset do sistema, no caso do sistema A, este é (A1&A2&A3&A4). Com isto, o programa procura remover quaisquer parenteses redundantes que existam na expressão, isto é realizado a partir do seguinte trecho de código:

```
def remove_parenteses_redundantes(expr):
```

```
            while expr.startswith('(') and expr.endswith(')'):
                depth = 0
                for i, char in enumerate(expr):
                    if char == '(':
                        depth += 1
                    elif char == ')':
                        depth -= 1

                    if depth == 0 and i < len(expr) - 1:
                        return expr
                expr = expr[1:-1]
            return expr
```

Após realizar este primeiro ciclo, o programa volta a encontrar o operador principal, que é neste caso, o primeiro que encontrar, e a partir daí volta a separar a expressão em dois, lado esquerdo e direito, até ambos lados forem eventos básicos. Quando forem eventos básicos o programa calcula a fiabilidade em função do tempo da ligação de ambos eventos, dependendo se esta for em série (|) ou em paralelo (&), realizado a partir da função (*evaluate*) do código:

```
     def evaluate(expr):
            expr = str(expr).strip()
            expr = remove_parenteses_redundantes(expr)
            operator, op_index = find_operator(expr)

            if operator is None:
                if expr in reliability_funcs:
                    result = reliability_funcs[expr]
                    print(f"Componente básico {expr}: {result}")
                    return result
                try:
                    result = sp.sympify(expr)
                    print(f"Convertido para símbolo: {result}")
                    return result
                except Exception as e:
                    raise ValueError(f"Erro ao processar componente: {expr}.
Erro: {e}")

            if operator == '&':

                left_expr = expr[:op_index].strip()
                right_expr = expr[op_index + 1:].strip()

                print(f"Componentes em Paralelo")
                print(f"Componente esquerdo: {left_expr}")
                print(f"Componente direito: {right_expr}")

                left_reliability = evaluate(left_expr)
                right_reliability = evaluate(right_expr)

                reliability = sp.simplify(1 - (1 - left_reliability) * (1 -
right_reliability))

                print(f"Fiabilidade em paralelo calculada: {reliability}")
                return reliability

            if operator == '|':
```

```
            left_expr = expr[:op_index].strip()
            right_expr = expr[op_index + 1:].strip()

            print(f"Componentes em Série")
            print(f"Componente esquerdo: {left_expr}")
            print(f"Componente direito: {right_expr}")

            left_reliability = evaluate(left_expr)
            right_reliability = evaluate(right_expr)

            reliability=sp.simplify(left_reliability* right_reliability)

            print(f"Fiabilidade em série calculada: {reliability}")
            return reliability

    return evaluate(expr)
```

Este ciclo continua até o programa calcular a fiabilidade do lado esquerdo e do lado direito da expressão do operador principal inicial e, a partir daí, calcula a fiabilidade em função do tempo de todo o sistema.

Após o cálculo da fiabilidade em função do tempo do sistema, na folha "Cálculo MTBF", o programa calcula, para valores instantâneos de t, o valor respetivo de fiabilidade do sistema. Estes valores de t são espaçados por um intervalo Δt, que pode ser alterado pelo utilizador. Um menor valor de Δt indica um menor erro no cálculo de MTBF.

	34-60 FLIGHT MANAGEMENT COMPUTING		
t (h)	R(t)	ΔR(t)	λ efetiva
0		0	
2		0	
4		0	
6		0	
8		0	
10		0	
12		0	
14		0	
16		0	
18		0	
20		0	
22		0	
24		0	
26		0	
28		0	
30		0	

Figura 45 - Cálculo da fiabilidade e taxa de falha para o sistema FMS

Fonte: (autor, 2025)

O cálculo da taxa de falha efetiva média é realizado na folha "Interface MTBF" a partir da média de todas as taxas de falha efetiva instantânea. Com a taxa de falha média, é calculado diretamente o valor de MTBF do sistema.

O penúltimo executável disponível chama-se de “Análise Temporal.exe” e este, como já referido no capítulo 3.4.3 realiza o cálculo da fiabilidade do sistema ao longo de 500.000 horas de operação, a partir do valor de MTBF calculado anteriormente. Para todos estes valores de tempo, o programa cria um gráfico de fiabilidade vs tempo para todos os sistemas e componentes analisados e guarda estes todos em formato de imagem png.

Os limites de manutenção existentes podem ser alterados pelo utilizador na folha de excel “Análise Temporal”. Estes valores são calculados através das seguintes linhas de código:

```
    if dado["Distribuição"] == "Exponencial":
                    R = reliability_exponential(t, dado["MTBF"])
                    label = f"{comp} (Exp)"
                    t_alerta = -dado["MTBF"] * np.log(limite_alerta)
                    t_critico = -dado["MTBF"] * np.log(limite_critico)
                else:
                    R = reliability_weibull(t, dado["Beta"], dado["Theta"])
                    label = f"{comp} (Weibull)"
                    try:
                        t_alerta = dado["Theta"] * (-np.log(limite_alerta))
** (1 / dado["Beta"])
                        t_critico = dado["Theta"] * (-np.log(limite_critico))
** (1 / dado["Beta"])
                    except (TypeError, ZeroDivisionError):
                        t_alerta = t_critico = float('inf')
```

	Distribuição Exponencial	Distribuição de Weibull		Tempo até limite de manutenção	
Componentes/Sistemas	MTBF [h]	β	θ	Limite de Alerta [h]	Limite Crítico [h]
24-20-01 Alternador		1,58	4500		
24-30-02 Bateria	20000				
24-30-01 Transformador		1,2	4820		
28-40-02 Manómetro de Combustível	100000				
28-40-01 Sensor de Quantidade de Combustível	10000				
27-10 ROLL CONTROL	50000				
27-20 YAW CONTROL	120000				
27-30 PITCH CONTROL	100000				

Figura 46 - Interface de dados de falha e tempos de manutenção

Fonte: (autor, 2025)

Após o cálculo dos tempos de manutenção, estes valores para cada sistema/componentes aparecem na folha de excel “Análise Temporal” ilustrada na figura 46.

Sistemas	Sistemas Críticos	MTBF [h]	CutSet Aeronave	Tempo para F(t)>10^-9	Tempo para F(t)=1
E	FMS	94860,56462	E|C|G|F|D	1,06434E-05	72828,41752
C	72.1 ENGINE	117668,4243			
G	SUPERFICIES SUSTENTADORAS	55728,34459			
F	AVIONICOS	20652,97284			
D	24 ELECTRICAL POWER	105858,8834			

Figura 47 - Análise temporal da aeronave

Fonte: (autor, 2025)

O último executável realiza uma análise temporal da aeronave como um todo, como todos os sistemas críticos encontram-se em série, o programa, através do MTBF calculado de cada sistema, calcula a fiabilidade geral da aeronave e realiza uma análise temporal da probabilidade de falha catastrófica do avião ao longo de 200.000 horas de voo acumuladas e gera um gráfico dessa probabilidade ao longo do tempo e guarda em formato imagem png.

Além desta análise, o programa também calcula dois pontos relevantes para análise de fiabilidade da aeronave que são: o instante de tempo a partir do qual a probabilidade de FC é maior que o objetivo de 10^{-9} e o instante de tempo a partir do qual a probabilidade de FC da aeronave é aproximadamente igual a 1. Estes dois valores de tempo são exportados para a folha "Análise Temporal Aeronave" tal como ilustrado na figura 47.

Desta forma, é concluído o programa de análise de fiabilidade de sistemas e componentes de uma aeronave. O seguinte capítulo realiza a análise quantitativa com os dados para os sistemas da aeronave LUS-222.

4.4 Análise LUS-222

Com todos os sistemas da aeronave definidos, o passo para a análise quantitativa consiste em recolher dados dos valores de MTBF de cada componentes/subsistema a considerar para introduzir no programa.

Devido à fase inicial de desenvolvimento do projeto, a quantidade de informações disponíveis sobre o desempenho dos componentes é reduzida. Esta limitação afeta diretamente a capacidade de estimar com confiança a fiabilidade dos sistemas analisados. Sem uma base de dados mais ampla e representativa, as previsões podem não refletir com precisão o desempenho real da aeronave em operação, tornando necessária a utilização de dados estimados ou genéricos provenientes de fontes externas.

De forma a colmatar a limitação de dados sobre o MTBF no âmbito deste projeto, foi adotada uma abordagem abrangente de pesquisa. Os dados foram obtidos através da consulta de diversas fontes, incluindo pesquisa na internet, artigos científicos e publicações técnicas. Esta metodologia permitiu reunir informações relevantes e diversificadas, reconhecendo que a natureza variada das fontes pode introduzir incerteza nos resultados.

A Tabela 11 inclui todos os valores de MTBF para todos os componentes e sistemas considerados neste estudo.

Além dos valores originais de MTBF dos componentes a tabela contém os valores de MTBF aumentados ($MTBF_1$) para os componentes nos quais se considera uma ação de manutenção preventiva com critério *hard time* a cada 2500 horas de voo acumuladas.

Tabela 11 - Valores de MTBF para os componentes analisados
Fonte: (autor, 2025)

Componentes/Sistemas	MTBF [h]	$MTBF_1$ [h]
24-20-01 Alternador	15.000	97.708
28-20-01 Bomba de combustível	10.000	45.208
34-40-01 Recetor GNSS	30.000	-
34-50-01 Recetor VOR	20.000	-
34-60-01 MCDU	25.000	-
34-60-02 FMGEC	30.000	-
72.1-00 GENERAL (Motor)	100.000	-

Uma vez introduzidos os valores numéricos de MTBF na folha "Dados de Fiabilidade", o passo seguinte consiste em construir os sistemas e os eventos de falha de acordo com os sistemas definidos no capítulo 4.1.

O cálculo da fiabilidade é realizado de acordo com as horas de voo acumuladas da aeronave. Para a arquitetura proposta, a probabilidade de falha catastrófica da aeronave é calculada para 1 hora de voo.

Para este tempo de operação, a probabilidade de falha dos sistemas calculados encontra-se na Tabela 12.

Tabela 12 - Probabilidade de falha dos sistemas analisados para 1 hora de voo
Fonte: (autor, 2025)

Definição	Sistemas Calculados	Probabilidade de Falha
A	34-60 FLIGHT MANAGEMENT COMPUTING	2.844×10^{-27}
B	34 NAVIGATION	3.61×10^{-9}
C	72.1 ENGINE	5.89×10^{-10}
D	24 ELECTRICAL POWER	6.94×10^{-10}
E	FMS	6.94×10^{-10}
F	AVIONICOS	4.30×10^{-9}
G	SUPERFICIES SUSTENTADORAS	1.28×10^{-9}

Com todos os sistemas calculados é realizada a escolha dos sistemas críticos. Estes são: FMS, 72.1 engine, 24 electrical power, superficies sustentadoras e aviónicos.

Uma vez escolhidos os sistemas críticos, o programa realiza o cálculo da probabilidade de falha catastrófica da aeronave em voo, ilustrado na Figura 48.

Probabilidade de Falha Catastrófica da Aeronave		
Probabilidade FC	Objetivo Imposto, [F(t)/fh]	Cumprimento do Objetivo
7,56562E-09	0,000000001	Não Cumpre Objetivo

Figura 48 - Probabilidade de falha catastrófica da aeronave para t=1 hora

Fonte: (autor, 2025)

Este resultado demonstra que a probabilidade de falha catastrófica da aeronave é marginalmente superior ao objetivo de fiabilidade imposto de 10^{-9} para um tempo de operação de 1 hora. Por causa disto, a arquitetura escolhida é classificada como "Não Cumpre Objetivo".

A principal razão para este desempenho reside na simplicidade da arquitetura proposta. Esta arquitetura apresenta um modelo demasiado simples para representar todos os sistemas da aeronave.

A árvore de falhas proposta considera apenas os componentes críticos de cada sistema, ao não considerar mais componentes, não existem redundâncias adicionais o que torna os sistemas mais vulneráveis à falha.

Para melhorar significativamente a fiabilidade da aeronave e atingir o objetivo imposto, seria necessário desenvolver uma arquitetura mais detalhada. Esta nova abordagem incluiria: mais componentes críticos; consideração de sistemas de proteção adicionais e uma análise detalhada das interdependências entre os sistemas o que levaria a uma representação mais precisa das falhas comuns.

Ainda que a arquitetura proposta não apresenta um resultado satisfatório no que toca à probabilidade de falha catastrófica da aeronave, é importante realçar que alcançar este objetivo não é o foco principal deste trabalho.

O propósito central é propor a implementação de um programa de análise de fiabilidade, na lógica da falha catastrófica e estabelecer um ponto de partida sólido para a análise dos sistemas da aeronave. Este programa tem como objetivo fornecer uma base estruturada para

futuras iterações e melhorias na arquitetura, permitindo uma abordagem progressiva para atingir os níveis de fiabilidade desejados.

Embora os resultados atuais revelem limitações na simplificação da arquitetura analisada, o trabalho cumpre o papel de lançar as bases para um estudo mais detalhado, que incluirá mais componentes críticos, redundâncias e mecanismos de proteção necessários para aumentar a fiabilidade da aeronave ao longo da sua vida útil.

Embora a arquitetura analisada não cumpra o objetivo de fiabilidade, ainda é possível e relevante continuar com a análise de importância e de sensibilidade dos sistemas. Estas análises irão fornecer informações que ajudam a determinar onde melhorias na redundância ou substituição de componentes terão o maior impacto na fiabilidade global do sistema.

Assim, estas ferramentas contribuem diretamente para o objetivo principal deste trabalho: estabelecer um ponto de partida sólido para a análise de fiabilidade e suportar a proposta de implementação de um programa de análise para os sistemas da aeronave.

Obtido o valor da probabilidade de falha catastrófica da aeronave, o passo seguinte do programa consiste em verificar quais são os componentes/sistemas que mais comprometem a segurança operacional da mesma.

Os minimal cutsets de cada sistema, calculados na folha "Minimal Cut Set", são dados pela seguinte tabela:

Tabela 13 - Minimal Cutsets dos sistemas analisados
Fonte: (autor, 2025)

Sistemas	Minimal Cutsets
34-60 FLIGHT MANAGEMENT COMPUTING (A)	$\{A1, A2, A3, A4, A5, A6\}$
34 NAVIGATION (B)	$\{B1, B2\}, \{B3, B4\}$
72.1 ENGINE (C)	$\{C1, C2\}, \{C3, C4\}$
24 ELECTRICAL POWER (D)	$\{C\}, \{D1, D2\}$
FMS (E)	$\{A\}, \{D\}$
AVIONICOS (F)	$\{B\}, \{D\}$
SUPERFICIES SUSTENTADORAS (G)	$\{C\}, \{D\}$

De forma a entender melhor quais os eventos básicos que mais influenciam o evento topo, a definição de cada evento básico encontra-se explícita na tabela seguinte.

Tabela 14 - Definição dos sistemas e eventos básicos

Fonte: (autor, 2025)

Evento Básico	Componente/Sistema
A1	34-60-01 MCDU
A2	34-60-01 MCDU
A3	34-60-01 MCDU
A4	34-60-01 MCDU
A5	34-60-02 FMGEC
A6	34-60-02 FMGEC
B1	34-50-01 Recetor VOR
B2	34-50-01 Recetor VOR
B3	34-40-01 Recetor GNSS
B4	34-40-01 Recetor GNSS
C1	72.1-00 GENERAL (Motor)
C2	72.1-00 GENERAL (Motor)
C3	28-20-01 Bomba de Combustível
C4	28-20-01 Bomba de Combustível
D1	24-20-01 Alternador
D2	24-20-01 Alternador
A	34-60 FLIGHT MANAGEMENT COMPUTING
B	34 NAVIGATION
C	72.1 ENGINE
D	24 ELECTRICAL POWER

A análise do ranking de importância dos eventos básicos é realizada com o objetivo de identificar os eventos básicos que têm maior impacto na segurança do sistema. Para esta análise, consideram-se os parâmetros BIR, RRW e RAW com um peso de 33.3% para cada um.

De salientar que esta análise específica tem como objetivo focar-se na potencialidade de melhorar a segurança do sistema como um todo, não focando na probabilidade de falha atual de cada evento, mas sim como é que a sua variação afetaria a probabilidade de falha global do sistema.

A seguinte discussão de resultados é feita considerando a árvore de falhas como um todo e não para cada sistema específico. Os dados obtidos para cada sistema específico (valores de importância e sensibilidade) encontram-se no apêndice B. O apêndice C contém os valores, para cada sistema, do ranking de importância dos eventos básicos pertencentes ao respetivo sistema.

Os parâmetros de importância calculados para os sistemas críticos e para todos os eventos pertencentes à arvore de falhas analisada encontram-se nas tabelas seguintes.

Tabela 15 - Parâmetros de importância dos sistemas críticos

Fonte: (autor, 2025)

Sistemas Críticos	BIR	CRIT	FV	RRW	RAW
E	1	0.0917	0.0917	1.11	∞
C	1	0.07788	0.07788	1.08	∞
G	1	0.169	0.169	1.20	∞
F	1	0.569	0.569	2.32	∞
D	1	0.0917	0.0917	1.10	∞

Tabela 16 - Parâmetros de importância dos eventos básicos

Fonte: (autor, 2025)

Eventos Básicos	BIR	CRIT	FV	RRW	RAW
A1	0	0	0	1	1
A2	0	0	0	1	1
A3	0	0	0	1	1
A4	0	0	0	1	1
A5	0	0	0	1	1
A6	0	0	0	1	1
B1	0.00005	0.33	0.33	2.4	11614
B2	0.00005	0.33	0.33	2.4	11614
B3	0.00003	0.15	0.15	1.34	7743
B4	0.00003	0.15	0.15	1.34	7743
C1	0.00001	0.013	0.013	1	2323
C2	0.00001	0.013	0.013	1	2323
C3	0.000022	0.065	0.065	1.13	5139
C4	0.000022	0.065	0.065	1.13	5139
D1	0.00001	0.014	0.014	1	2378
D2	0.00001	0.014	0.014	1	2378

A equação para calcular os valores de importância é dada por:

$$I_i = 0.33 \cdot BIR_i + 0.33 \cdot RRW_i + 0.34 \cdot RRW_i$$

Dito isto, o ranking de importância dos sistemas críticos obtido é ilustrado através do seguinte gráfico.

Figura 49 - Ranking de importância dos sistemas críticos

Fonte: (autor, 2025)

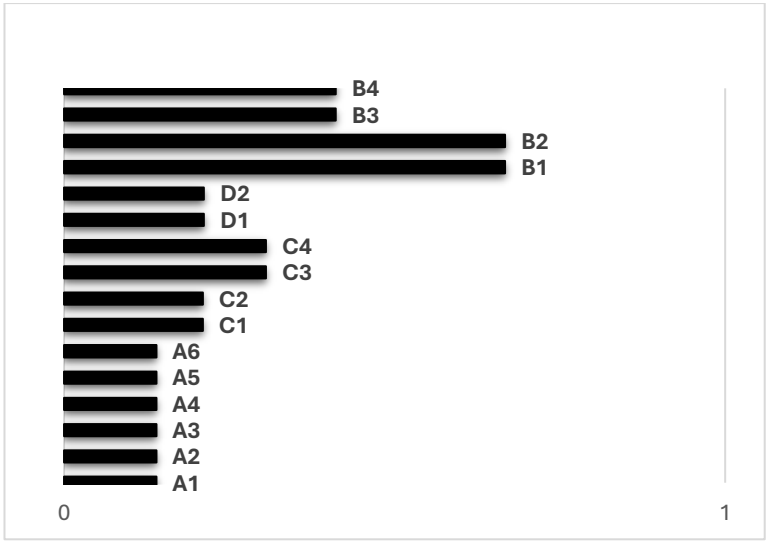


Figura 50 - Ranking de importância dos eventos básicos de todos os sistemas críticos

Fonte: (autor, 2025)

A figura 49 demonstra que o sistema no qual se deve dar maior importância é o sistema F, que, neste caso, são os aviónicos da aeronave. Ao analisar todos os eventos básicos da árvore de falhas da aeronave (Figura **50**) chega-se à conclusão de que o evento B1 e B2 (recetores VOR) são os componentes mais críticos de toda a arquitetura considerada.

Além dos recetores VOR, os restantes componentes apresentam um valor semelhante de importância para a árvore de falhas e, portanto, também devem ser considerados.

Apesar de alguns destes eventos apresentarem uma probabilidade de ocorrência baixa, como por exemplo ambos motores (eventos C1 e C2) ou os alternadores (eventos D1 e D2), a forma como estes estão conectados à arvore de falhas influencia no cálculo da importância, e,

como estes eventos encontram-se conectados em série, esta análise reconhece que é necessário priorizar estes eventos, independentemente da probabilidade de falha atual.

No que toca à análise de sensibilidade, a variação percentual de $[-50\%;\ -25\%;\ 25\%;\ 50\%]$ para os sistemas críticos e para todos os eventos da árvore de falhas encontra-se ilustrado nos seguintes gráficos.

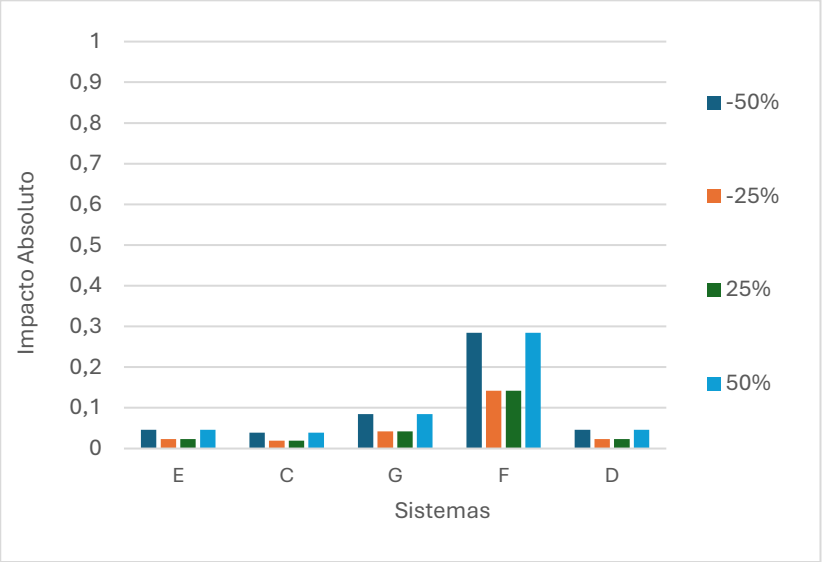


Figura 51 - Impacto absoluto dos sistemas críticos

Fonte: (autor, 2025)

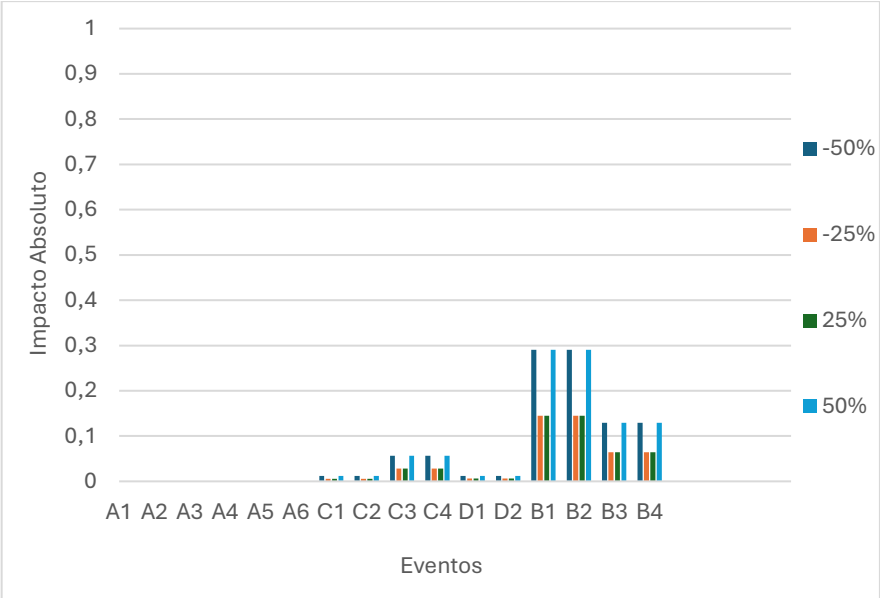


Figura 52 - Impacto absoluto dos eventos analisados

Fonte: (autor, 2025)

A análise de sensibilidade confirma as conclusões obtidas na análise de importância, destacando o impacto significativo das variações de probabilidade do sistema de aviónicos na probabilidade de falha catastrófica da aeronave. Entre os eventos básicos, os componentes do sistema de navegação (B1 a B4) apresentam o maior impacto absoluto, reforçando a sua relevância dentro dos aviónicos como um todo. Em contraste, os restantes eventos mostram impactos absolutos consideravelmente menores, indicando que as ações de mitigação devem priorizar o sistema de navegação.

A principal diferença entre as duas análises está no foco e nos pressupostos utilizados. A análise de sensibilidade avalia como alterações nas probabilidades atuais de cada evento afetam diretamente a probabilidade de ocorrência do evento topo, enquanto a análise de importância examina a relevância estrutural dos componentes, independentemente das suas probabilidades de ocorrência, com o objetivo de identificar componentes que poderiam ser alvo de melhorias estruturais a longo prazo.

A análise de sensibilidade é particularmente útil por considerar as probabilidades reais, permitindo identificar onde intervenções imediatas podem ter maior impacto. Já a análise de importância foca-se em cenários hipotéticos, sendo mais estratégica e orientada para melhorias estruturais. A conjugação destas duas abordagens oferece uma visão mais completa do sistema.

Por exemplo, na análise de importância, os eventos do sistema elétrico, D1, D2 e eventos do sistema propulsivo (C) apresentam um valor de importância elevado, indicando a sua relevância estrutural (encontram-se conectados em série, o que indica que ao falhar, o sistema elétrico falharia). No entanto, na análise de sensibilidade, estes eventos praticamente não têm

impacto devido à sua probabilidade de ocorrência mais baixa. Isto demonstra que, embora estruturalmente relevantes, estes eventos não devem ser prioritários para ações imediatas, mas podem ser considerados numa perspetiva de longo prazo.

Embora os resultados possam ser semelhantes para os eventos mais críticos (como os recetores), a utilização conjunta destas duas análises permite uma perspetiva mais abrangente. A análise de sensibilidade ajuda a priorizar ações práticas e imediatas, enquanto a análise de importância fornece uma visão estratégica, identificando oportunidades de melhoria estrutural de longo prazo.

O próximo passo do programa é o cálculo do valor de MTBF para cada sistema, sendo este o foco das análises subsequentes. A Tabela 17 contém os valores de MTBF obtidos para todos os sistemas construídos.

Tabela 17 - Valores de MTBF para os sistemas construídos
Fonte: (autor, 2025)

Sistemas	MTBF [h]
34-60 FLIGHT MANAGEMENT COMPUTING	912.801,94
34 NAVIGATION	25.655,166
72.1 ENGINE	117.668,42
24 ELECTRICAL POWER	105.858,88
FMS	94.860,56
AVIONICOS	20.652,97
SUPERFICIES SUSTENTADORAS	55.728,34

O cálculo do MTBF dos sistemas reforça as conclusões obtidas nas análises de importância e sensibilidade, que identificaram os eventos B1 a B4 eventos pertencentes ao sistema 34 navigation como os mais críticos e prioritários para mitigação. O baixo valor de MTBF do sistema de navegação justifica a sua relevância nas análises anteriores, uma vez que este sistema tem um impacto significativo na fiabilidade global dos aviónicos da aeronave. O valor de MTBF dos aviónicos, por sua vez, é fortemente influenciado pelo desempenho limitado do sistema de navegação.

Para a análise temporal de cada sistema considerou-se um limite de alerta equivalente a 98% e um limite crítico de 90%. A seguinte tabela contém os valores de tempo que os sistemas demoram a atingir os limites considerados.

Tabela 18 - Tempo para limites de manutenção de cada componente

Fonte: (autor, 2025)

Componentes/Sistemas	Tempo até limite de alerta [h]	Tempo até limite crítico [h]
24-20-01 Alternador	1973.96	10294.5
28-20-01 Bomba de combustível	913.3	4763.1
34-40-01 Recetor GNSS	606.1	3160.8
34-50-01 Recetor VOR	404.05	2107.2
34-60-01 MCDU	505.1	2634
34-60-02 FMGEC	606.1	3160.8
72.1-00 GENERAL	2020.3	10536.1
34-60 FLIGHT MANAGEMENT COMPUTING	18441.1	96173.3
34 NAVIGATION	518.3	2703.04
72.1 ENGINE	2377.2	12397.6
24 ELECTRICAL POWER	2138.6	11153.3
FMS	1916.4	9994.6
AVIONICOS	417.2	2176
SUPERFICIES SUSTENTADORAS	1125.9	5871.6

Como mencionado anteriormente, para os componentes que não estão sujeitos ao critério de *hard time*, propõe-se a adoção do critério *on condition*, ativando ações de manutenção preventiva assim que o componente atinge o limite de alerta.

Neste trabalho, a análise foi realizada para todos os componentes, aplicando o conceito de limite de alerta de forma generalizada. Embora, na prática, os limites de manutenção variem de acordo com as características e a criticidade de cada componente, esta abordagem uniforme foi adotada como um primeiro passo no desenvolvimento do programa de fiabilidade. Esta simplificação permite estabelecer as bases da análise e testar a viabilidade do modelo proposto.

Os valores destes limites foram definidos de forma arbitrária para simplificar a análise e demonstrar a aplicabilidade do programa. Contudo, na prática, a definição dos limites de manutenção exige uma análise mais detalhada, considerando fatores como: criticidade do componente; custos de manutenção; condições operacionais; histórico de falhas, entre outros.

A análise revelou que o componente que atinge o limite de alerta mais rapidamente é o 34-50-01 Recetor VOR, com apenas 404 horas de voo acumuladas. Este resultado indica a necessidade de aplicar o critério *on condition* a este componente, já que não está sujeito ao critério de *hard time*.

Importa referir que esta análise temporal não tem como objetivo definir limites de manutenção específicos. O propósito desta etapa é monitorizar o comportamento da fiabilidade dos componentes e sistemas ao longo de 500.000 horas de voo acumuladas, ajudando a compreender o desempenho e apoiar decisões futuras. O gráfico seguinte ilustra o comportamento destes sistemas ao longo do tempo.

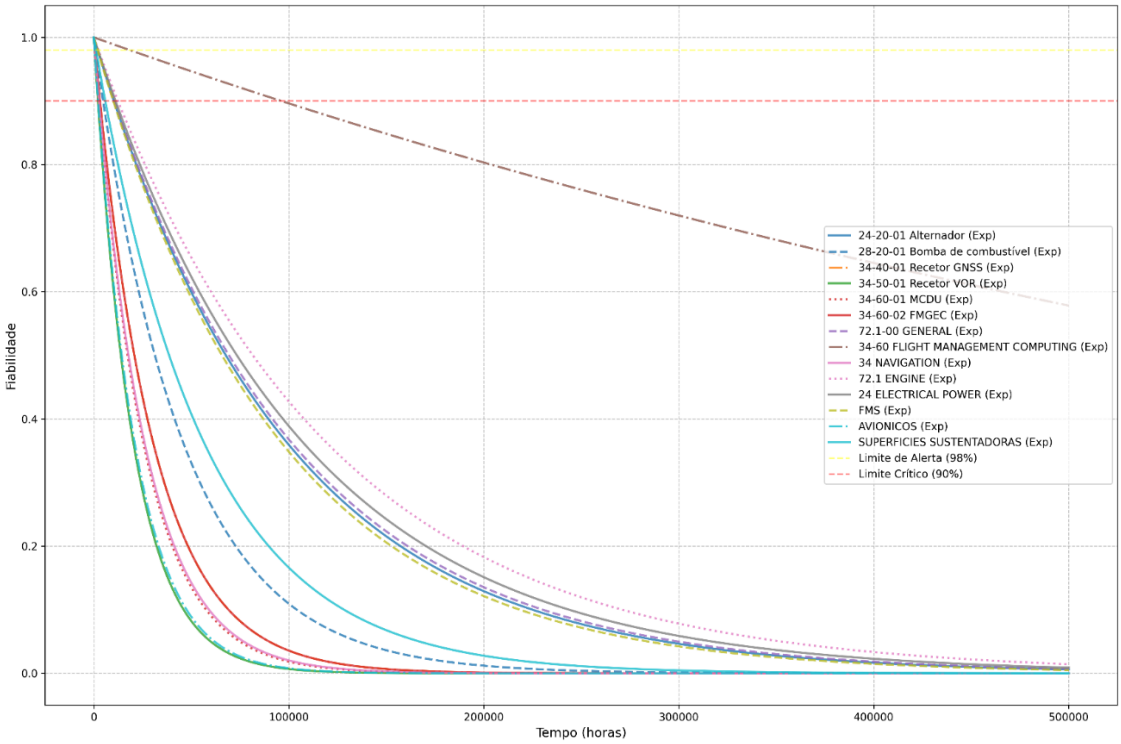


Figura 53 - Análise temporal dos componentes e sistemas analisados

Fonte: (autor, 2025)

A análise temporal da probabilidade da FC da aeronave encontra-se ilustrada na figura 53. Os valores calculados nesta análise são o tempo a partir do qual a probabilidade de falha catastrófica da aeronave é maior que 10^{-9} e o tempo a partir do qual a probabilidade de falha catastrófica da aeronave é aproximadamente 1.

Os valores calculados são os seguintes:

Tempo para $F(t) > 10^{-9}$ [h]	1,06434E-05
Tempo para $F(t) \approx 1$ [h]	72828,41752

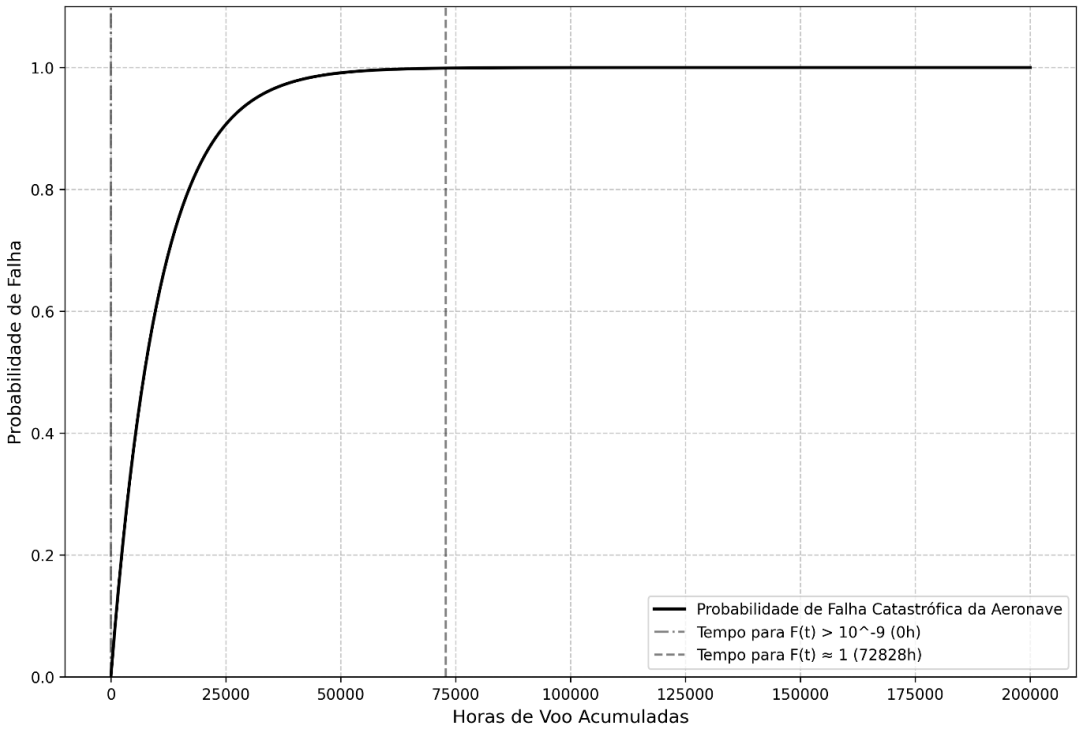


Figura 54 - Probabilidade de falha catastrófica da aeronave ao longo do tempo

Fonte: (autor, 2025)

Observa-se através do gráfico que a probabilidade de falha cresce rapidamente, aproximando-se de 1 à medida que o tempo avança, o que reflete a inevitabilidade de falhas em períodos extensos sem intervenções.

Além disso, a análise mostra que o critério de fiabilidade estabelecido, que exige que a probabilidade de falha catastrófica permaneça abaixo de 10^{-9} ao longo da vida útil da aeronave (estimada em 100.000 horas), não é cumprido desde o início do período analisado. A probabilidade de falha ultrapassa este limite imediatamente no início da operação (0 horas), evidenciando que o sistema, conforme modelado, não atende ao requisito de fiabilidade.

Adicionalmente, a probabilidade de falha catastrófica atinge valores próximos de 1 após cerca de 72.828 horas de operação, muito antes das 100.000 horas de vida útil esperada para a aeronave. Este comportamento resulta, em grande parte, do facto de a análise considerar apenas os componentes críticos de cada sistema, sem incluir redundâncias que poderiam mitigar falhas individuais. A ausência de redundâncias torna os sistemas mais vulneráveis, uma vez que falhas em componentes críticos afetam diretamente a fiabilidade global da aeronave.

Esta etapa final do programa de análise de fiabilidade tem como objetivo estudar a variação da probabilidade de FC da aeronave ao longo do tempo de operação. Além disso, verifica se a probabilidade se mantém abaixo dos níveis aceitáveis durante a vida útil prevista, permitindo avaliar a conformidade com os critérios de fiabilidade definidos.

Capítulo 5 – Modelo de validação

Visto que o tema da dissertação é o estudo de viabilidade de adoção de um programa de avaliação da fiabilidade para a aeronave LUS-222 tendo por referência a falha catastrófica, a qual ainda se encontra numa fase inicial do projeto. Este capítulo dá uma perspetiva sobre como poderá ser conduzido o modelo de validação do processo descrito no capítulo 4.

5.1 Processo de validação

A validação do programa de fiabilidade deve seguir um processo alinhado com as normas e padrões da indústria aeronáutica para garantir sua aceitação e uso dentro da empresa, bem como poder ter validação no contexto do processo de engenharia, que envolve validação por parte de engenheiros de certificação, no contexto da EASA, ou seja, *Compliance Verification Engineer* (CVE).

Em concordância com este trabalho, é essencial que o programa cumpra a norma SAE ARP4761 (*Aerospace Recommended Practice - Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*).

Esta norma estabelece as diretrizes fundamentais para a avaliação de segurança de sistemas aeronáuticos, com ênfase particular em técnicas quantitativas como a análise de árvore de falhas.

O apêndice D da ARP4761 (SAE International, 1996, p. 50) detalha especificamente a metodologia da FTA, que constitui o núcleo do programa. Além disso, este documento fornece orientações sobre a avaliação quantitativa da árvore de falhas (SAE International, 1996, p. 76), incluindo análises de importância e de sensibilidade, ambas integradas no programa desenvolvido.

A escolha da ARP4761 como referência principal justifica-se pelo facto de esta norma definir um quadro estruturado e amplamente aceite para a análise de fiabilidade e segurança na indústria aeronáutica.

Como o objetivo do programa é demonstrar a viabilidade da sua adoção no contexto do LUS222, seguir as diretrizes da ARP4761 assegura que a metodologia implementada está alinhada com os padrões regulamentares.

Desta forma, para validar o programa de acordo com a ARP4761, é necessário garantir que a lógica e os cálculos matemáticos utilizados no programa sejam consistentes com os métodos descritos na norma. Esta verificação pode incluir a comparação dos resultados obtidos

pelo programa com casos de referência estabelecidos na literatura, garantindo a exatidão dos modelos implementados.

Além da conformidade com a ARP4761, que assegura que a metodologia de análise de fiabilidade está alinhada com as melhores práticas da indústria aeronáutica, é igualmente essencial considerar a validação do software que implementa estas análises.

O programa desenvolvido utiliza código em *python* para realizar os cálculos de importância, sensibilidade e análises temporais, pelo que a sua fiabilidade depende não só da precisão dos modelos matemáticos, mas também da correção, robustez e rastreabilidade do software.

Para garantir um desenvolvimento de software consistente com os padrões da aviação, pode ser aplicada a norma RTCA DO-178C (*Software Considerations in Airborne Systems and Equipment Certification*). Esta norma define um conjunto de processos e requisitos para o desenvolvimento e certificação de software utilizado em sistemas embarcados na aviação, garantindo que este cumpre rigorosos critérios de segurança e fiabilidade.

Embora o programa desenvolvido não pertença a nenhum sistema embarcado, alguns princípios da DO-178C podem ser aplicados para garantir que o software segue um processo estruturado, possui rastreabilidade dos requisitos e é testado de forma rigorosa.

Por exemplo, a norma define níveis de rigor diferentes conforme o impacto do software na segurança (Parasoft, p. 12), e mesmo para softwares não críticos, recomenda práticas como testes de verificação, análise de cobertura de código e documentação detalhada do desenvolvimento.

Assim, pode ser útil adotar algumas diretrizes da DO-178C para reforçar a qualidade e fiabilidade do programa, mesmo que a sua aplicação integral não seja necessária.

No entanto, dado que a DO-178C é específica para software embarcado, a norma ISO/IEC 25010 (*Systems and Software Quality Requirements and Evaluation (SQuaRE)*) pode ser mais apropriada para avaliar a qualidade geral do software. Esta norma não é específica para aviação, mas estabelece critérios para avaliar a correção, eficiência, segurança e manutenibilidade do código (International Organization for Standardization, 2011, p. 9).

Uma vez estabelecido este enquadramento normativo, o passo seguinte consiste na implementação de um plano de verificação e validação estruturado. Este plano deve iniciar-se com o desenvolvimento de uma matriz de conformidade (*compliance matrix*[23]) que mapeia os

[23] Definida como um quadro estruturado que conecta e coordena diversos componentes institucionais de conformidade, com o objetivo de garantir a conformidade institucional com normas e regulamentos (Standford University, p. 1).

requisitos específicos da SAE ARP4761, com especial atenção ao Apêndice D e às diretrizes para análise quantitativa.

Esta matriz deve identificar cada requisito da norma aplicável ao programa, estabelecer critérios de aceitação mensuráveis e definir métodos de verificação específicos, sejam eles por análise, teste, demonstração ou inspeção.

Paralelamente, é fundamental criar um conjunto abrangente de casos de teste baseados em exemplos conhecidos da literatura aeronáutica, casos de referência da própria ARP4761 e cenários específicos do contexto da empresa.

Estes casos devem cobrir análises de árvore de falhas simples e complexas, cálculos de importância e sensibilidade, bem como análises temporais com diferentes distribuições de probabilidade, assegurando assim a robustez do programa em diferentes condições operacionais.

Desta maneira, é possível validar diferentes características do software do programa, tais como:

- Consistência (os mesmos *inputs* conduzem aos mesmos *outputs*);
- Escalabilidade (consegue processar árvores de falha de diferentes dimensões e complexidades);
- Eficiência computacional (o programa deve realizar os cálculos num tempo razoável, independentemente do tamanho da árvore a analisar)
- Sensibilidade a erros e exceções (o programa deve identificar corretamente e lidar com entradas inválidas, inconsistências lógicas nas árvores de falhas ou valores de probabilidade fora dos limites aceitáveis);
- Rastreabilidade dos resultados (o programa deve manter um registo claro das operações realizadas, permitindo aos utilizadores entender como cada resultado foi obtido).

Estas são apenas algumas características que o software do programa deve apresentar de modo a garantir que o programa de análise de fiabilidade cumpre os requisitos formais e oferece uma ferramenta fiável para apoiar decisões relativas à segurança de sistemas aeronáuticos.

Neste processo, os CVE assumem um papel crucial para a aprovação do programa.

O documento (Independent Checking Function Assessment, 2011) indica que os CVEs são responsáveis pela "verificação independente das demonstrações de conformidade" (EASA, 2011, p. 4). Esta função independente de verificação é um requisito regulamentar estabelecido no ponto 21A.239(b) do documento *Acceptable Means of Compliance Part 21 Subpart A.239(b)* (EASA, 2003, p. 148), que determina que "o sistema de garantia do projeto deve incluir uma função de verificação independente das demonstrações de conformidade, com base nas quais

a organização submete declarações de conformidade e documentação associada à agência" (EASA, 2011, p. 4).

De forma resumida, os CVEs são engenheiros especializados que exercem uma função de verificação independente das demonstrações de conformidade, conforme estabelecido nos requisitos regulamentares. Esta função é essencial para garantir que o programa desenvolvido cumpre rigorosamente as normas e padrões da indústria aeronáutica, particularmente a SAE ARP4761.

Relativamente às responsabilidades específicas dos CVEs no processo de validação, estas incluem a "aprovação por assinatura de todos os documentos de conformidade, incluindo programas e dados de teste, necessários para a verificação da conformidade com os requisitos aplicáveis" (EASA, 2011, p. 5).

Além disso, os CVEs são responsáveis pela "aprovação do conteúdo técnico (completamento, precisão técnica...), incluindo quaisquer revisões subsequentes, dos manuais aprovados" (EASA, 2011, p. 5).

Um aspeto crucial da função dos CVEs são a sua independência. O processo de validação exige que a organização deve "garantir um nível apropriado de independência, significando que o CVE não deve estar envolvido na criação dos dados de conformidade" (EASA, 2015, p. 1). Esta separação entre quem desenvolve e quem verifica é essencial para manter a integridade do processo de validação e evitar potenciais conflitos de interesse que possam comprometer a qualidade da verificação.

Para o programa de análise de fiabilidade, os CVEs devem avaliar especificamente os aspetos técnicos relacionados com a implementação dos algoritmos de árvore de falhas, os cálculos de importância e sensibilidade, e as análises temporais e verificariam se estes elementos estão corretamente implementados de acordo com a metodologia estabelecida na norma SAE ARP4761, particularmente no seu Apêndice D.

A sua intervenção assegura que cada elemento do programa pode ser rastreado até um requisito específico da norma, garantindo assim a conformidade normativa e integridade da solução.

Uma vez aprovado pelos CVEs, o programa de análise de fiabilidade entra na fase de implementação operacional. Nesta fase, o projeto deverá integrar o programa nos seus processos de engenharia e gestão de segurança, estabelecendo protocolos para a sua utilização nos projetos correntes e futuros.

São desenvolvidos procedimentos internos para a correta aplicação do programa, incluindo a formação dos utilizadores, o estabelecimento de mecanismos de feedback para

identificação de potenciais melhorias e a definição de processos para a gestão de alterações ao programa.

Em síntese o modelo descrito é que se preconiza para verificar e validar o modelo de determinação da fiabilidade na ótica da probabilidade da falha catastrófica.

Capítulo 6 - Conclusões

6.1 Conclusões do estudo

O desenvolvimento deste programa constitui um primeiro passo para a implementação de um sistema estruturado de avaliação da fiabilidade do LUS222, tendo como objetivo fundamental desenvolver um modelo de análise da fiabilidade da aeronave em conformidade com o requisito regulamentar que estabelece uma probabilidade de falha catastrófica não superior a 10^{-9} por hora de voo.

A abordagem adotada permitiu analisar sistemas críticos da aeronave e quantificar a sua fiabilidade ao longo do tempo, fornecendo um quadro conceptual que poderá ser refinado e expandido em iterações futuras.

O programa seguiu as diretrizes da norma SAE ARP4761 para a construção de árvores de falhas, garantindo que a sua estrutura e abordagem estivessem em conformidade com os padrões estabelecidos na indústria aeronáutica. Além disso o programa inclui ferramentas para avaliar a influência de cada componente no sistema global, permitindo identificar elementos críticos cuja falha tem um impacto significativo na segurança da aeronave.

A inclusão de análises temporais veio acrescentar uma nova dimensão à avaliação da fiabilidade, permitindo prever a degradação dos sistemas ao longo do tempo e definir limites de manutenção, como os limites de alerta e críticos.

Esta abordagem possibilita a implementação de estratégias de manutenção *on condition*, aumentando a eficiência das intervenções de manutenção e garantindo que a aeronave permanece dentro do limite de probabilidade de falha catastrófica exigido ao longo de toda a sua vida operacional.

No entanto, como um programa de primeira geração, este estudo enfrentou desafios significativos que devem ser considerados em futuras iterações. As limitações impostas à análise da aeronave, como a exclusão de fatores como fadiga estrutural, sistemas mecânicos e de anti gelo, bem como falhas por erro humano, foram deliberadas para simplificar o modelo e garantir a viabilidade da construção, face à falta de informação técnica do LUS-222 (sistemas, MTBF, arquitetura, entre outros).

Focando no software em si, algumas limitações foram identificadas. Em termos de escalabilidade e complexidade computacional, a análise de fiabilidade baseada em árvores de falhas pode tornar-se computacionalmente exigente à medida que a estrutura se expande. Embora o programa tenha demonstrado eficácia na análise de sistemas isolados e de menor dimensão, será necessário otimizar o seu desempenho para lidar com arquiteturas mais complexas da aeronave que possam influenciar significativamente a probabilidade de falha catastrófica global.

Outra limitação prende-se com a análise de componentes que seguem distribuições de Weibull (com taxa de falha variável). O programa, na sua versão atual, não consegue calcular o MTBF de sistemas que incluam tais componentes, o que impede a realização de análises temporais para esses sistemas. Se um desses componentes pertencer a um sistema crítico, isso compromete a capacidade do programa de analisar a probabilidade de falha catastrófica da aeronave ao longo do tempo, reduzindo a profundidade da informação disponibilizada e potencialmente dificultando a verificação precisa do cumprimento do limite de 10^{-9}.

Adicionalmente, o programa depende fortemente de *inputs* manuais, exigindo que os utilizadores insiram os dados de distribuição dos componentes, assim como toda a estrutura da árvore de falhas diretamente no Excel. Esta abordagem pode tornar o processo mais demorado e menos eficiente, especialmente para análises recorrentes ou sistemas mais complexos.

Apesar destas limitações, o programa desenvolvido representa uma base sólida na avaliação da fiabilidade da aeronave LUS222. A sua capacidade de estruturar e analisar árvores de falhas, identificar componentes críticos e integrar análises temporais confere-lhe um bom potencial. Além disso, a conformidade com normas da indústria e a flexibilidade para incorporar novas funcionalidades fazem com que este seja um ponto de partida promissor para um sistema mais robusto no futuro.

Com melhorias ao nível da automação, escalabilidade e integração com outras ferramentas, este programa poderá tornar-se uma ferramenta útil na gestão da fiabilidade de sistemas da aeronave e na verificação contínua do cumprimento do limite crítico de probabilidade de falha catastrófica de 10^{-9}.

6.2 Sugestões para trabalhos futuros

Tendo em conta as limitações identificadas, existem várias oportunidades para o aperfeiçoamento e expansão deste programa em trabalhos futuros. Uma das principais áreas de desenvolvimento passa pela automatização de certas etapas do processo que poderá melhorar significativamente a usabilidade do programa.

Atualmente, a inserção manual de dados e a construção da árvore de falhas no Excel representam um obstáculo para análises recorrentes e sistemas mais complexos. Assim, o

desenvolvimento de uma interface mais intuitiva, que permita a importação automática de dados e a geração semi-automatizada da árvore de falhas, reduziria o tempo e o esforço necessários para realizar novas análises.

Uma outra proposta relevante para trabalhos futuros seria a incorporação de uma análise de incerteza nos cálculos de fiabilidade. Atualmente, o programa baseia-se em valores pontuais para as probabilidades de falha dos componentes, mas na prática, esses valores podem estar sujeitos a variações devido a diferentes condições operacionais, imprecisões nos dados ou limitações estatísticas na recolha de informações.

A introdução de métodos estatísticos como simulações de Monte Carlo permitiria quantificar o impacto da incerteza nos resultados, fornecendo intervalos de confiança para as métricas de fiabilidade em vez de valores fixos.

Por fim, um avanço natural deste trabalho seria a sua aplicação a um conjunto mais alargado de sistemas da aeronave, incorporando outros subsistemas críticos, como os sistemas hidráulicos, estruturais, entre outros. Essa abordagem possibilitaria uma visão mais holística da fiabilidade global da aeronave, tornando o programa uma ferramenta ainda mais robusta para a gestão da segurança e manutenção do LUS222.

Referências

Aeroflot Russian Airlines. (2022). *Glossary*. Retrieved Mar24, from glossary_eng.pdf (aeroflot.com)

Afsharnia, F. (2017). *Chapter 7 - Failure Rate Analysis*. INTECH. Retrieved Set24

AICEP. (2024, Junho). Aeronáutica, Espaço e Defesa. *O céu pode não ser o limite*, p. 34. Retrieved Jul24

Birolini, A. (2014). *Reliability Engineering*. Springer. Retrieved Dez24

Contini, S., & Matuzas, V. (2010). *Components' Importance Measure for Initiating and Enabling Events in Fault Tree Analysis* . JRC.

Denning, R. (2012). Capítulo 6 - Probabilistic R&M Parameters and Redundancy Calculations. In *Applied R&M Manual for Defence Systems*. Retrieved Jul24

Department of Defense USA. (2020). *Interface Standard System Subsystem Sub-Subsystem Numbering*.

EASA. (2003). *Acceptable Means of Compliance and Guidance Material for the airworthiness and environmental certification of aircraft and related products, parts and appliances, as well as for the certification of design and production organisations*. Retrieved Fev25

EASA. (2011). *Independent Checking Function Assessment*. Retrieved Fev25, from https://www.easa.europa.eu/sites/default/files/dfu/Presentation%206%20-%20Independent%20Checking%20Function%20Assessment.pdf

EASA. (2015). *FAQs: What are the expectations of the Agency in respect to the appointment of CVEs?* Retrieved Fev25

EASA. (2018). *Easy Access Rules for Generic Master Minimum Equipment List (CS-GEN-MMEL)*. Retrieved Jul24

EASA; FAA. (2021). *Maintenance Annex Guidance*.

Ebeling, C. E. (2003). *An Introduction to Reliability and Maintainability Engineering*. McGraw-Hill. Retrieved Maio24

European Organisation for the Safety of Air Navigation. (2005). *Fault Tree Analysis (FTA) Guidance Material*. Retrieved Agosto 24

European Union Aviation Safety Agency. (2020). *EASA AMC 25.1309*. Retrieved Jun24

Griguere, H. (1991). *Flight Management System Back-Up Navigation for the A330/A340 Aircraft*. IEEE. Retrieved Out24

IATA. (2023). *Air Passenger Market Analysis*. Retrieved Mar24, from iata.org/en/iata-repository/publications/economic-reports/air-passenger-market-analysis-december-2023/

IEA. (1980-2020). *World Air Passenger Traffic Evolution*. Retrieved Mar24, from https://www.iea.org/data-and-statistics/charts/world-air-passenger-traffic-evolution-1980-2020

International Organization for Standardization. (2011). *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models*. Retrieved Fev25, from https://cdn.standards.iteh.ai/samples/35733/2ca18b477b7845a5b8cae39d6de0c098/ISO-IEC-25010-2011.pdf

Jacobsen, S. A. (1996). *Integrated Logistics Support In Special Operations Aviation - A Case Study of the MH-60K and MH-47E*. Tese, California. Retrieved Jul24, from https://apps.dtic.mil/sti/tr/pdf/ADA311378.pdf

Juran, J. M., & Godfrey, A. B. (1998). *Juran's Quality Handbook*. McGraw-Hill. Retrieved Maio24

Kapur, K. C., & Pecht, M. (2014). *Reliability Engineering*. Wiley. Retrieved Abril24

Kinnison, H. A., & Siddiqui, T. (2012). In *Aviation Maintenance Management*. McGraw-Hill. Retrieved Mar24

Lundteigen, M. A., & Rausand, M. (n.d.). *Chapter 5. Fault Tree Analysis (FTA)*. Retrieved Agosto24, from Norwegian University of Science and Technology - Department of Mechanical and Industrial Engineering: https://www.ntnu.edu/documents/624876/1277046207/SIS+book+-+chapter+05+-+Introduction+to+fault+trees/fa8ba01a-3baf-4bb8-94ed-116bf5bc6b44

Mobley, R. K. (2004). *Maintenance Fundamentals 2nd Edition*. ELSEVIER.

Morgado, M. (2022, 10 12). *Avião português nasce em 2025. Projeto vai custar 100 milhões de euros e criar 300 postos de trabalho*. Retrieved Mar24, from https://24.sapo.pt/atualidade/artigos/aviao-portugues-nasce-em-2025-vai-custar-100-milhoes-de-euros-e-criar-300-postos-de-trabalho

NASA. (1997). *Designing for Maintainability and System Availability*. Retrieved Agosto24

NASA. (2000, Julho). (V. R. Lalli, A. H. Malec, & H. M. Packard, Eds.) *Reliability and Maintainability (RAM) Training*. Retrieved Jul24

Norwegian University of Science and Technology. (n.d.). Appendix D. *Fault tree analysis*, pp. 1-15. Retrieved Agosto 24, from Fault Tree Analysis: https://jvatn.folk.ntnu.no/eLearning/PK6032/pdf/AppD-FTA.pdf

Parasoft. (n.d.). *DO-178C Software Compliance For Aerospace & Defense*.

Pecht, M. (1999). Chapter 8 - Reliability, Maintainability and Availability. In A. P. Sage, & W. B. Rouse, *Handbook of Systems Engineering and Management* (pp. 303-326). Wiley Interscience. Retrieved Jul24

Pike, L. (2010). A Critical Systems Blog, Thoughts on safe and secure embedded computer systems. *10 to the -9*. Retrieved Jun24

Pinto, P. (2022, Outubro 15). *LUS 222: O avião português vai ser produzido em Ponte de Sor*. Retrieved Julho 2024, from pplware: https://pplware.sapo.pt/informacao/lus-222-o-aviao-portugues-vai-ser-produzido-em-ponte-de-sor/

Pita, O. (n.d.). *Introduction to Aircraft Maintenance*. Retrieved from ICAO: https://www.icao.int/APAC/Meetings/2019%20COSCAP%20SEAEASA%20AIR/M20%20Aircraft%20Maintenance%20Programs.pdf

Priberam. (2008-2021). *Fiabilidade*. Retrieved Abril24, from Dicionário Priberam da Língua Portuguesa: https://dicionario.priberam.org/fiabilidade

PTC. (n.d.). Retrieved Setembro 24, from Windchill Risk and Reliability: https://support.ptc.com/help/wrr/r12.0.2.0/en/index.html#page/wrr/welcome.html#

Ribeiro de Oliveira, J. L. (2015). *Developing a reliability program for maintenance and operation.* Dissertação de Mestrado, Instituto Superior Técnico. Retrieved Abril24

Ross, S. M. (2014). In *Introduction to Probability and Statistics for Engineers and Scientists.* Academic Press. Retrieved Abril24

SAE International. (1996). *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*. Retrieved Fev25

Sigman, K. (2009). *IEOR 6711: Introduction to Renewal Theory.* Retrieved Nov24, from https://www.columbia.edu/~ks20/stochastic-I/stochastic-I-RRT.pdf

Standford University. (n.d.). *What is a Compliance Matrix?* Retrieved Fev25, from https://uco.stanford.edu/sites/g/files/sbiybj20651/files/media/file/what_is_a_compliance_matrix_0.pdf

Vaskic, L., & Paetzold, K. (2019). *A Critical Review of The Integrated Logistics Support Suit for Aerospace and Defense Programmes.* Retrieved Dez24

Verma, A. (2016). Chapter 2 Basic Reliability Mathematics. In *Reliability and Safety Engineering* (pp. 19-73). Londres: Springer. Retrieved Jul24

Vesely, W. E., Goldberg, F. F., Roberts, N. H., & Haasl, D. F. (1981). *Fault Tree Handbook.* U.S. Nuclear Regulatory Commission. Retrieved Agosto 24

Vieira, D. R., Rebaiaia, M. L., & Chain, M. C. (2016). *The Application of Realiability Methods for Aircraft Design Project Management*, pp. 967-992. Retrieved Mar24

Vincent, J. (2021). *System Safety Objectives: One in a billion? - Rotorcraft and VTOL Symposium 2021.* Retrieved Jun24, from YouTube: https://www.youtube.com/watch?v=zHnxncOdjEc&t=1082s

Weibull.com. (2001, 9 7). *Reliability Basics.* Retrieved Mar24, from https://www.ukm.my/kamal3/rro/math%20Reliability%20Function.pdf

Apêndice A

Este apêndice contém todos os códigos em *python* utilizados para o programa de análise de fiabilidade.

Ficheiro 'Programa Análise Sensibilidade.py':

```
import subprocess

def executar_script(script_path):
    try:
        result = subprocess.run([script_path], check=True,
capture_output=True, text=True)
        print(f"Execução de {script_path} concluída com sucesso.")
        print("Saída:")
        print(result.stdout)
    except subprocess.CalledProcessError as e:
        print(f"Erro ao executar {script_path}.")
        print("Saída de erro:")
        print(e.stderr)

if __name__ == "__main__":
    scripts_para_executar = [
        "Minimal Cutset Sistemas.exe",
        "Analise Sensibilidade Sistemas.exe",
        "MCS Aeronave.exe",
        "Analise Sensibilidade Aeronave.exe"
    ]
    for script in scripts_para_executar:
        executar_script(script)
```

Ficheiro 'Minimal Cutset Sistemas.py':

```
import xlwings as xw
from sympy import symbols, simplify_logic
from sympy.logic.boolalg import to_dnf
import re

def extract_variables(expression):
    pattern = r'[A-Z]\d*'
    return sorted(set(re.findall(pattern, expression)))

def boolean_operators(expression):
    expression = expression.replace('+', '|')
    expression = expression.replace('*', '&')
    return expression

def algebraic_operators(expression):
    expression = str(expression)
    expression = expression.replace('|', '+')
    expression = expression.replace('&', '*')
    expression = expression.replace('(', '').replace(')', '')
    return expression
```

```
    def simplify_boolean_expression(expression):
        if not expression:
            return ""
        try:
            expression = expression.replace(" ", "")
            expression = boolean_operators(expression)

            vars_in_expr = extract_variables(expression)
            sym_dict  =  {var:  symbols(var,  boolean=True)  for  var  in
vars_in_expr}

            expr_with_symbols = expression
            for var in vars_in_expr:
                expr_with_symbols      =      expr_with_symbols.replace(var,
f"sym_dict['{var}']")

            expr_obj = eval(expr_with_symbols, {"sym_dict": sym_dict})

            simplified_expr = to_dnf(expr_obj, simplify=True)

            result = algebraic_operators(simplified_expr)

            print(f"{result}")
            return result

        except Exception as e:
            print(f"Erro: {str(e)}")
            return f"Erro: {str(e)}"

    def process_excel_file(file_path):
        try:
            app = xw.App(visible=False)
            wb = app.books.open(file_path)
            folha = wb.sheets['Minimal Cut Set']
            start_row = 7
            end_row = 40

            for row in range(start_row, end_row + 1):
                expressao_booleana = folha.range(f'C{row}').value
                if expressao_booleana:
                    resultado                                                =
simplify_boolean_expression(expressao_booleana)
                    folha.range(f'D{row}').value = resultado

            wb.save()
            wb.close()

        except Exception as e:
            print(f"Erro: {str(e)}")

        finally:
            app.quit()

    if __name__ == "__main__":
        file_path = 'Análise-Fiabilidade.xlsm'
        process_excel_file(file_path)
        print("Expressões simplificadas com sucesso.")
```

Ficheiro 'Analise Sensibilidade Sistemas.py':

```
import subprocess, sys
import pandas as pd
import sympy as sp
import numpy as np
import re
import win32com.client as win32
import os
import math
import traceback

def extract_variables(expression):
    pattern = r'[A-Z]\d*'
    return sorted(set(re.findall(pattern, expression)))

def expression_parser(expression):
    vars_in_expr = extract_variables(expression)
    expr_with_symbols = expression
    for var in vars_in_expr:
        expr_with_symbols = expr_with_symbols.replace(var,
f"symbols('{var}')")
    expr_obj = eval(expr_with_symbols, {"symbols": sp.symbols})
    return vars_in_expr, expr_obj

def calculate_system_probability(expression, event_prob_dict, sym_dict):
    numeric_expr = expression
    for event, prob in event_prob_dict.items():
        numeric_expr = numeric_expr.subs(sym_dict[event], prob)
    return float(numeric_expr.evalf())

def calculate_component_measures(expression, event_prob_dict, sym_dict,
event):
    base_prob = calculate_system_probability(expression,
event_prob_dict, sym_dict)

    rrw_dict = event_prob_dict.copy()
    rrw_dict[event] = 0
    prob_with_zero = calculate_system_probability(expression, rrw_dict,
sym_dict)

    raw_dict = event_prob_dict.copy()
    raw_dict[event] = 1
    prob_with_one = calculate_system_probability(expression, raw_dict,
sym_dict)

    rrw = float('inf') if prob_with_zero == 0 else base_prob /
prob_with_zero
    raw = prob_with_one / base_prob

    return rrw, raw

def calculate_absolute_impact(base_prob, varied_prob):
    return abs(varied_prob - base_prob) / base_prob

def calculate_system_varied_probability(expression, event_prob_dict,
sym_dict, target_event, varied_prob):
```

```
        varied_dict = event_prob_dict.copy()
        varied_dict[target_event] = varied_prob
        return     calculate_system_probability(expression,     varied_dict,
sym_dict)

    if getattr(sys, 'frozen', False):
        exe_path = os.path.dirname(sys.executable)
    else:
        exe_path = os.path.dirname(os.path.abspath(__file__))

    excel = win32.Dispatch("Excel.Application")
    excel.Visible = False

    excel_path = os.path.join(exe_path, "./Análise-Fiabilidade.xlsm")

    try:
        workbook = excel.Workbooks.Open(excel_path)
        sheet_name = "Análise Sensibilidade"
        sheet = workbook.Sheets(sheet_name)

        for table in sheet.ListObjects:
            print(f"Table Name: {table.Name}")
            rows = []
            data_range = table.DataBodyRange
            for row in data_range.Rows:
                row_data = [cell.Value for cell in row.Cells]
                rows.append(row_data)

            rows = np.array(rows)

            expression = rows[0, 0]
            system_events = rows[:, 1]
            probabilities = rows[:, 2]
            minimal_cutsets = rows[:, 7]

            prob_minus_50 = rows[:, 12]
            prob_minus_25 = rows[:, 14]
            prob_plus_25 = rows[:, 16]
            prob_plus_50 = rows[:, 18]

            event_prob_dict = {}
            for event, prob in zip(system_events, probabilities):
                if  event  and  event  !=  ''  and  event  is  not  None  and  not
pd.isna(event):
                    if prob and not pd.isna(prob):
                        event_prob_dict[event] = float(prob)

            event_list = list(event_prob_dict.keys())

            if event_list:
                vars_in_expr, expr_obj = expression_parser(expression)
                sym_dict = {var: sp.symbols(var) for var in vars_in_expr}

                derivate_list = []
                numeric_results = []

                base_system_prob   =   calculate_system_probability(expr_obj,
event_prob_dict, sym_dict)
```

```
                for i, event in enumerate(event_list):
                    try:
                        derivative = sp.diff(expr_obj, sym_dict[event])
                        derivate_list.append(derivative)

                        numeric_derivative = derivative
                        for e, prob in event_prob_dict.items():
                            numeric_derivative                                  =
numeric_derivative.subs(sym_dict[e], prob)

                        numeric_result = float(numeric_derivative.evalf())
                        numeric_results.append(numeric_result)

                        rrw,                        raw                         =
calculate_component_measures(expr_obj,event_prob_dict,sym_dict,event)

                        if rrw == float('inf'):
                            data_range.Cells(i + 1, 11).Value = "∞"
                        else:
                            data_range.Cells(i + 1, 11).Value = rrw

                        data_range.Cells(i + 1, 12).Value = raw

                        print(f"Componente {event} - RRW: {rrw}, RAW: {raw}")


                        if not pd.isna(prob_minus_50[i]):
                            varied_prob                                         =
calculate_system_varied_probability(expr_obj,
                                event_prob_dict,
                                sym_dict,
                                event,
                                float(prob_minus_50[i])
                            )
                            impact_minus_50                                     =
calculate_absolute_impact(base_system_prob, varied_prob)
                            data_range.Cells(i    +    1,    14).Value    =
impact_minus_50

                        if not pd.isna(prob_minus_25[i]):
                            varied_prob                                         =
calculate_system_varied_probability(
                                expr_obj,
                                event_prob_dict,
                                sym_dict,
                                event,
                                float(prob_minus_25[i])
                            )
                            impact_minus_25                                     =
calculate_absolute_impact(base_system_prob, varied_prob)
                            data_range.Cells(i    +    1,    16).Value    =
impact_minus_25

                        if not pd.isna(prob_plus_25[i]):
                            varied_prob                                         =
calculate_system_varied_probability(
                                expr_obj,
                                event_prob_dict,
                                sym_dict,
```

```
                                event,
                                float(prob_plus_25[i])
                            )
                            impact_plus_25                                      =
calculate_absolute_impact(base_system_prob, varied_prob)
                            data_range.Cells(i    +    1,    18).Value    =
impact_plus_25

                        if not pd.isna(prob_plus_50[i]):
                            varied_prob                                         =
calculate_system_varied_probability(
                                expr_obj,
                                event_prob_dict,
                                sym_dict,
                                event,
                                float(prob_plus_50[i])
                            )
                            impact_plus_50                                      =
calculate_absolute_impact(base_system_prob, varied_prob)
                            data_range.Cells(i    +    1,    20).Value    =
impact_plus_50


                    except Exception as e:
                        print(f"Erro   a   processar   componente   {event}:
{str(e)}")
                        data_range.Cells(i + 1, 11).Value = "Erro"
                        data_range.Cells(i + 1, 12).Value = "Erro"


                for i, cutset in enumerate(minimal_cutsets):
                    if cutset and cutset != '' and cutset is not None and not
pd.isna(cutset):
                        try:

                            vars_in_cutset,          cutset_expr          =
expression_parser(cutset)


                            cutset_prob = cutset_expr
                            for event, prob in event_prob_dict.items():
                                if event in vars_in_cutset:
                                    cutset_prob                             =
cutset_prob.subs(sym_dict[event], prob)

                            cutset_result = float(cutset_prob.evalf())
                            data_range.Cells(i + 1, 9).Value = cutset_result

                        except Exception as e:
                            print(f"Erro a processar MCS {cutset}: {str(e)}")
                            print(traceback.format_exc())
                            data_range.Cells(i + 1, 9).Value = "Erro"


                for i, (derivative, numeric) in enumerate(zip(derivate_list,
numeric_results)):
                    data_range.Cells(i + 1, 4).Value = str(derivative)
                    data_range.Cells(i + 1, 5).Value = numeric
```

```
            else:
                print("Sem Eventos!")

        workbook.Save()

    finally:
        workbook.Close(SaveChanges=True)
        excel.Quit()
```

Ficheiro 'MCS Aeronave.py'

```
import xlwings as xw
from sympy import symbols
from sympy.logic.boolalg import to_dnf
import re

def extract_variables(expression):
    pattern = r'[A-Z]\d*'
    return sorted(set(re.findall(pattern, expression)))

def boolean_operators(expression):
    expression = expression.replace('+', '|')
    expression = expression.replace('*', '&')
    return expression

def algebraic_operators(expression):
    expression = str(expression)
    expression = expression.replace('|', '+')
    expression = expression.replace('&', '*')
    expression = expression.replace('(', '').replace(')', '')
    return expression

def check_absorption(term1, term2):

    vars1 = set(extract_variables(term1))
    vars2 = set(extract_variables(term2))

    if vars1.issubset(vars2):
        return True

    for var in vars1:
        single_var_set = {var}

        if single_var_set.issubset(vars2):
            return True

    return False

def apply_complete_absorption(expression):

    terms = expression.split('+')
    terms = [term.strip() for term in terms if term.strip()]

    terms = list(dict.fromkeys(terms))

    final_terms = []
```

```
    terms.sort(key=lambda x: len(extract_variables(x)))

    for i, term1 in enumerate(terms):
        should_add = True

        for existing_term in final_terms:
            if check_absorption(existing_term, term1):
                should_add = False
                break
            elif check_absorption(term1, existing_term):
                final_terms.remove(existing_term)

        if should_add:
            final_terms.append(term1)

    return '+'.join(final_terms)

def split_and_simplify(expression, max_vars=8):

    try:
        expression = boolean_operators(expression)

        terms = expression.split('|')

        groups = []
        current_group = []
        current_vars = set()

        for term in terms:
            term_vars = set(extract_variables(term))
            if    len(current_vars.union(term_vars))    >    max_vars    and
current_group:
                group_expr = '|'.join(current_group)
                vars_in_group = {var: symbols(var, boolean=True) for var in
current_vars}
                simplified_group = to_dnf(eval(group_expr, {}, vars_in_group),
simplify=True)
                groups.append(str(simplified_group))

                current_group = [term]
                current_vars = term_vars
            else:
                current_group.append(term)
                current_vars.update(term_vars)

        if current_group:
            group_expr = '|'.join(current_group)
            vars_in_group  =  {var:  symbols(var,  boolean=True)  for  var  in
current_vars}
            simplified_group  =  to_dnf(eval(group_expr,  {},  vars_in_group),
simplify=True)
            groups.append(str(simplified_group))

        combined = '|'.join(groups)

        result = algebraic_operators(combined)
        result = apply_complete_absorption(result)

        return result
```

```
    except Exception as e:
        print(f"Erro: {str(e)}")
        return expression

def simplify_boolean_expression(expression):
    if not expression:
        return ""
    try:
        expression = expression.replace(" ", "")
        return split_and_simplify(expression)
    except Exception as e:
        print(f"Erro: {str(e)}")
        return f"Erro: {str(e)}"

def excel(file_path):
    try:
        app = xw.App(visible=False)
        wb = app.books.open(file_path)
        folha = wb.sheets['Análise Sensibilidade Aeronave']

        expressao_booleana = folha.range('B25').value
        if expressao_booleana:
            resultado = simplify_boolean_expression(expressao_booleana)
            print(f"{str(resultado)}")
            folha.range('B28').value = resultado

        wb.save()
        wb.close()

    except Exception as e:
        print(f"Erro: {str(e)}")

    finally:
        app.quit()

if __name__ == "__main__":
    file_path = 'Análise-Fiabilidade.xlsm'
    excel(file_path)
    print("Expressão simplificada com sucesso.")
```

Ficheiro 'Analise Sensibilidade Aeronave.py':

```
import subprocess, sys
import pandas as pd
import sympy as sp
import numpy as np
import re
import win32com.client as win32
import os
import math

def extract_variables(expression):
    pattern = r'[A-Z]\d+'
    return sorted(set(re.findall(pattern, expression)))

def expression_parser(expression):
    vars_in_expr = extract_variables(expression)
    expr_with_symbols = expression
    for var in vars_in_expr:
        expr_with_symbols = expr_with_symbols.replace(var,
f"symbols('{var}')")
    expr_obj = eval(expr_with_symbols, {"symbols": sp.symbols})
    return vars_in_expr, expr_obj

def calculate_system_probability(expression, event_prob_dict, sym_dict):
    numeric_expr = expression
    for event, prob in event_prob_dict.items():
        numeric_expr = numeric_expr.subs(sym_dict[event], prob)
    return float(numeric_expr.evalf())

def calculate_component_measures(expression, event_prob_dict, sym_dict,
event):
    base_prob = calculate_system_probability(expression,
event_prob_dict, sym_dict)

    rrw_dict = event_prob_dict.copy()
    rrw_dict[event] = 0
    prob_with_zero = calculate_system_probability(expression, rrw_dict,
sym_dict)

    raw_dict = event_prob_dict.copy()
    raw_dict[event] = 1
    prob_with_one = calculate_system_probability(expression, raw_dict,
sym_dict)

    rrw = float('inf') if prob_with_zero == 0 else base_prob /
prob_with_zero
    raw = prob_with_one / base_prob

    return rrw, raw

def calculate_absolute_impact(base_prob, varied_prob):
    return abs(varied_prob - base_prob) / base_prob

def calculate_system_varied_probability(expression, event_prob_dict,
sym_dict, target_event, varied_prob):
    varied_dict = event_prob_dict.copy()
    varied_dict[target_event] = varied_prob
```

```
        return     calculate_system_probability(expression,     varied_dict,
sym_dict)

    if getattr(sys, 'frozen', False):
        exe_path = os.path.dirname(sys.executable)
    else:
        exe_path = os.path.dirname(os.path.abspath(__file__))

    excel = win32.Dispatch("Excel.Application")
    excel.Visible = False

    excel_path = os.path.join(exe_path, "./Análise-Fiabilidade.xlsm")

    try:
        workbook = excel.Workbooks.Open(excel_path)
        sheet_name = "Análise Sensibilidade Aeronave"
        sheet = workbook.Sheets(sheet_name)

        if sheet.ListObjects.Count > 0:
            table = sheet.ListObjects(1)
            print(f"Processing Table - Table Name: {table.Name}")
            rows = []
            data_range = table.DataBodyRange
            for row in data_range.Rows:
                row_data = [cell.Value for cell in row.Cells]
                rows.append(row_data)

            rows = np.array(rows)

            expression = rows[0, 0]
            system_events = rows[:, 1]
            probabilities = rows[:, 2]
            minimal_cutsets = rows[:, 6]


            prob_minus_50 = rows[:, 11]
            prob_minus_25 = rows[:, 13]
            prob_plus_25 = rows[:, 15]
            prob_plus_50 = rows[:, 17]

            event_prob_dict = {}
            for event, prob in zip(system_events, probabilities):
                if  event  and  event  !=  ''  and  event  is  not  None  and  not
pd.isna(event):
                    if prob and not pd.isna(prob):
                        event_prob_dict[event] = float(prob)

            event_list = list(event_prob_dict.keys())

            if event_list:
                vars_in_expr, expr_obj = expression_parser(expression)
                sym_dict = {var: sp.symbols(var) for var in vars_in_expr}


                base_system_prob   =   calculate_system_probability(expr_obj,
event_prob_dict, sym_dict)

                derivate_list = []
                numeric_results = []
```

```
                for i, event in enumerate(event_list):
                    try:
                        derivative = sp.diff(expr_obj, sym_dict[event])
                        derivate_list.append(derivative)

                        numeric_derivative = derivative
                        for e, prob in event_prob_dict.items():
                            numeric_derivative                                  =
numeric_derivative.subs(sym_dict[e], prob)

                        numeric_result = float(numeric_derivative.evalf())
                        numeric_results.append(numeric_result)


                        rrw, raw = calculate_component_measures(
                            expr_obj,
                            event_prob_dict,
                            sym_dict,
                            event
                        )


                        if rrw == float('inf'):
                            data_range.Cells(i + 1, 10).Value = "∞"
                        else:
                            data_range.Cells(i + 1, 10).Value = rrw


                        data_range.Cells(i + 1, 11).Value = raw


                        if not pd.isna(prob_minus_50[i]):
                            varied_prob                                         =
calculate_system_varied_probability(
                                expr_obj,
                                event_prob_dict,
                                sym_dict,
                                event,
                                float(prob_minus_50[i])
                            )
                            impact_minus_50                                     =
calculate_absolute_impact(base_system_prob, varied_prob)
                            data_range.Cells(i    +    1,    13).Value    =
impact_minus_50


                        if not pd.isna(prob_minus_25[i]):
                            varied_prob                                         =
calculate_system_varied_probability(
                                expr_obj,
                                event_prob_dict,
                                sym_dict,
                                event,
                                float(prob_minus_25[i])
                            )
                            impact_minus_25                                     =
calculate_absolute_impact(base_system_prob, varied_prob)
```

```
                            data_range.Cells(i    +    1,    15).Value    =
impact_minus_25


                        if not pd.isna(prob_plus_25[i]):
                            varied_prob                                     =
calculate_system_varied_probability(
                                expr_obj,
                                event_prob_dict,
                                sym_dict,
                                event,
                                float(prob_plus_25[i])
                            )
                            impact_plus_25                                  =
calculate_absolute_impact(base_system_prob, varied_prob)
                            data_range.Cells(i    +    1,    17).Value    =
impact_plus_25


                        if not pd.isna(prob_plus_50[i]):
                            varied_prob                                     =
calculate_system_varied_probability(
                                expr_obj,
                                event_prob_dict,
                                sym_dict,
                                event,
                                float(prob_plus_50[i])
                            )
                            impact_plus_50                                  =
calculate_absolute_impact(base_system_prob, varied_prob)
                            data_range.Cells(i    +    1,    19).Value    =
impact_plus_50

                        print(f"Componente {event} - RRW: {rrw}, RAW: {raw}")

                    except Exception as e:
                        print(f"Error   a   processar   componente   {event}:
{str(e)}")
                        data_range.Cells(i + 1, 10).Value = "Erro"
                        data_range.Cells(i + 1, 11).Value = "Erro"


                for i, cutset in enumerate(minimal_cutsets):
                    if cutset and cutset != '' and cutset is not None and not
pd.isna(cutset):
                        try:
                            vars_in_cutset,          cutset_expr          =
expression_parser(cutset)
                            numeric_cutset = cutset_expr
                            for event, prob in event_prob_dict.items():
                                if event in vars_in_cutset:
                                    numeric_cutset                          =
numeric_cutset.subs(sym_dict[event], prob)
                            cutset_result = float(numeric_cutset.evalf())
                            data_range.Cells(i + 1, 8).Value = cutset_result
                        except Exception as e:
                            print(f"Error   a   processar   MCS   {cutset}:
{str(e)}")
                            data_range.Cells(i + 1, 8).Value = "Erro"
```

```
                for i, (derivative, numeric) in enumerate(zip(derivate_list,
numeric_results)):
                    data_range.Cells(i + 1, 4).Value = str(derivative)
                    data_range.Cells(i + 1, 5).Value = numeric

            else:
                print("Sem eventos!")

        workbook.Save()

    finally:
        workbook.Close(SaveChanges=True)
        excel.Quit()
```

Ficheiro 'MTBF.py':

```
    import xlwings as xw
    import sympy as sp
    import re
    t = sp.Symbol('t')

    def find_systems(sheet):

        systems_data = []
        range_data             =             sheet.range('A1:H'             +
str(sheet.cells.last_cell.row)).value

        for i in range(len(range_data)):
            if   isinstance(range_data[i][1],   str)   and   "Sistema"   in
range_data[i][1]:
                specific_name = range_data[i][1]
                cutset = range_data[i + 1][7] if i + 1 < len(range_data) else
None
                events = []
                for j in range(i + 1, len(range_data)):
                    if    not    range_data[j][2]    or    "Sistema"    in
str(range_data[j][1]):
                        break
                    event = {
                        'id': range_data[j][2],
                        'component': range_data[j][3],
                        'mtbf': range_data[j][4],
                        'beta': range_data[j][5],
                        'theta': range_data[j][6]
                    }
                    events.append(event)

                systems_data.append({
                    'name': specific_name,
                    'cutset': cutset,
                    'events': events
                })
        return systems_data

    def read_expression(expr, reliability_funcs):
```

```
        def find_operator(expr):
            depth = 0
            for i, char in enumerate(expr):
                if char == '(':
                    depth += 1
                elif char == ')':
                    depth -= 1
                elif depth == 0 and (char == '&' or char == '|'):
                    return char, i
            return None, -1

        def remove_parenteses_redundantes(expr):

            while expr.startswith('(') and expr.endswith(')'):
                depth = 0
                for i, char in enumerate(expr):
                    if char == '(':
                        depth += 1
                    elif char == ')':
                        depth -= 1

                    if depth == 0 and i < len(expr) - 1:
                        return expr
                expr = expr[1:-1]
            return expr

        def evaluate(expr):
            expr = str(expr).strip()
            expr = remove_parenteses_redundantes(expr)
            operator, op_index = find_operator(expr)

            if operator is None:
                if expr in reliability_funcs:
                    result = reliability_funcs[expr]
                    print(f"Componente básico {expr}: {result}")
                    return result
                try:
                    result = sp.sympify(expr)
                    print(f"Convertido para símbolo: {result}")
                    return result
                except Exception as e:
                    raise ValueError(f"Erro ao processar componente: {expr}.
Erro: {e}")

            if operator == '&':

                left_expr = expr[:op_index].strip()
                right_expr = expr[op_index + 1:].strip()

                print(f"Componentes em Paralelo")
                print(f"Componente esquerdo: {left_expr}")
                print(f"Componente direito: {right_expr}")

                left_reliability = evaluate(left_expr)
                right_reliability = evaluate(right_expr)

                reliability = sp.simplify(1 - (1 - left_reliability) * (1 -
right_reliability))
```

```
                print(f"Fiabilidade em paralelo calculada: {reliability}")
                return reliability

            if operator == '|':

                left_expr = expr[:op_index].strip()
                right_expr = expr[op_index + 1:].strip()

                print(f"Componentes em Série")
                print(f"Componente esquerdo: {left_expr}")
                print(f"Componente direito: {right_expr}")

                left_reliability = evaluate(left_expr)
                right_reliability = evaluate(right_expr)

                reliability      =      sp.simplify(left_reliability      *
right_reliability)

                print(f"Fiabilidade em série calculada: {reliability}")
                return reliability

        return evaluate(expr)

    def reliability_event_functions(events):
        reliability_funcs = {}

        for event in events:
            if event['mtbf']:
                reliability_funcs[event['id']] = sp.exp(-t / event['mtbf'])

            elif event['beta'] and event['theta']:
                reliability_funcs[event['id']] = sp.exp(-(t / event['theta'])
** event['beta'])

        return reliability_funcs

    def system_reliability_functions(system_data):

        reliability_funcs                                                =
reliability_event_functions(system_data['events'])

        try:
            cutset_expr = system_data['cutset'].replace(' ', '')
            processed_expr = read_expression(cutset_expr, reliability_funcs)
            system_reliability = sp.simplify(processed_expr)
            return system_reliability
        except Exception as e:
            print(f"Erro ao processar expressão: {e}")
            return None

    def system_reliability_values(system_reliability, t_values):

        reliability_values = []
        for t_val in t_values:
            try:
                reliability_at_t = float(system_reliability.subs(t, t_val))
                reliability_values.append(reliability_at_t)
            except Exception as e:
                reliability_values.append(None)
```

```
        return reliability_values

    def main():
        file_path = "Análise-Fiabilidade.xlsm"
        app = xw.App(visible=False)
        wb = app.books.open(file_path)

        temporal_sheet = wb.sheets["Interface MTBF"]
        mtbf_sheet = wb.sheets["Cálculo MTBF"]

        try:
            systems_data = find_systems(temporal_sheet)
            if not systems_data:
                print("Nenhum sistema encontrado para análise.")
                return

            t_values          =          mtbf_sheet.range('B6:B'          +
str(mtbf_sheet.cells.last_cell.row)).value
            t_values = [float(val) for val in t_values if val is not None and
isinstance(val, (int, float))]

            system_output_columns = {}
            for i, system in enumerate(systems_data):
                col_index = 3 + i * 3
                if col_index <= 26:
                    output_col = chr(ord('A') + col_index - 1)
                else:
                    primeira_letra = chr(ord('A') + (col_index - 1) // 26 -
1)
                    segunda_letra = chr(ord('A') + (col_index - 1) % 26)
                    output_col = primeira_letra + segunda_letra
                system_output_columns[system['name']] = output_col

            for system in systems_data:
                system_reliability_expr                                      =
system_reliability_functions(system)

                if system_reliability_expr is not None:
                    output_col  =  system_output_columns.get(system['name'],
'C')
                    reliability_values                                       =
system_reliability_values(system_reliability_expr, t_values)
                    output_range    =    f"{output_col}6:{output_col}{5    +
len(reliability_values)}"
                    mtbf_sheet.range(output_range).value = [[val] for val in
reliability_values]
                    print("Analisar próximo sistema:")

        except Exception as e:
            print(f"Ocorreu um erro: {e}")

        finally:
            wb.save()
            wb.close()
            app.quit()

    if __name__ == "__main__":
        main()
```

Ficheiro 'Análise Temporal.py':

```
import xlwings as xw
import numpy as np
import matplotlib
matplotlib.use('Agg')

import matplotlib.pyplot as plt

def reliability_exponential(t, mtbf):
    return np.exp(-t / mtbf)

def reliability_weibull(t, beta, theta):
    if beta is None or theta is None or theta == 0:
        return np.zeros_like(t)
    else:
        return np.exp(-(t / theta) ** beta)

def main():
    file_path = "Análise-Fiabilidade.xlsm"
    app = xw.App(visible=False)
    wb = None

    try:
        wb = app.books.open(file_path)
        sheet = wb.sheets["Análise Temporal"]

        componentes = sheet.range("B4:B106").value
        mtbf_values = sheet.range("C4:C106").value
        beta_values = sheet.range("D4:D106").value
        theta_values = sheet.range("E4:E106").value
        limite_alerta = sheet.range("J4").value
        limite_critico = sheet.range("J5").value

        dados_falha = []
        for comp, mtbf, beta, theta in zip(componentes, mtbf_values,
beta_values, theta_values):
            if comp:
                if mtbf:
                    dados_falha.append({
                        "Componente/Sistema": comp,
                        "Distribuição": "Exponencial",
                        "MTBF": mtbf
                    })
                elif beta and theta:
                    dados_falha.append({
                        "Componente/Sistema": comp,
                        "Distribuição": "Weibull",
                        "Beta": beta,
                        "Theta": theta
                    })

        t = np.linspace(0, 500000, 1000)
        tempos_limite = {}

        plt.figure(figsize=(15, 10))
        plt.grid(True, linestyle='--', alpha=0.7)

        cmap = matplotlib.colormaps.get_cmap('tab10')
```

```
            colors = cmap(np.linspace(0, 1, len(dados_falha)))
            line_styles = ['-', '--', '-.', 'solid', 'dotted']

            for idx, (dado, color) in enumerate(zip(dados_falha, colors)):
                line_style = line_styles[idx % len(line_styles)]
                comp = dado["Componente/Sistema"]

                if dado["Distribuição"] == "Exponencial":
                    R = reliability_exponential(t, dado["MTBF"])
                    label = f"{comp} (Exp)"
                    t_alerta = -dado["MTBF"] * np.log(limite_alerta)
                    t_critico = -dado["MTBF"] * np.log(limite_critico)
                else:
                    R = reliability_weibull(t, dado["Beta"], dado["Theta"])
                    label = f"{comp} (Weibull)"
                    try:
                        t_alerta = dado["Theta"] * (-np.log(limite_alerta))
** (1 / dado["Beta"])
                        t_critico = dado["Theta"] * (-np.log(limite_critico))
** (1 / dado["Beta"])
                    except (TypeError, ZeroDivisionError):
                        t_alerta = t_critico = float('inf')

                plt.plot(t, R, label=label, color=color, alpha=0.8,
linewidth=2, linestyle=line_style)

                if t_alerta != float('inf') and t_critico != float('inf'):
                    sheet.range(f"F{componentes.index(comp) + 4}").value =
t_alerta
                    sheet.range(f"G{componentes.index(comp) + 4}").value =
t_critico
                else:
                    sheet.range(f"F{componentes.index(comp) + 4}").value =
None
                    sheet.range(f"G{componentes.index(comp) + 4}").value =
None

                tempos_limite[comp] = {"alerta": t_alerta, "critico":
t_critico}

            plt.axhline(y=limite_alerta, color='yellow', linestyle='--',
alpha=0.5,
                        label=f'Limite de Alerta ({limite_alerta:.0%})')
            plt.axhline(y=limite_critico, color='red', linestyle='--',
alpha=0.5,
                        label=f'Limite Crítico ({limite_critico:.0%})')

            plt.xlabel('Tempo (horas)', fontsize=12)
            plt.ylabel('Fiabilidade', fontsize=12)
            plt.legend(loc='center right', fontsize=10)

            plt.tight_layout()
            plt.savefig('Análise Temporal.png', bbox_inches='tight',
dpi=300)

            wb.save()

        except Exception as e:
            print(f"Erro durante a execução: {e}")
```

```
        finally:
            if wb:
                wb.close()
            app.quit()

            print("\nTempos Limites de cada Componente/Sistema:")
            for comp, times in tempos_limite.items():
                print(f"\n{comp}:")
                print(f"  Tempo até limite de alerta: {times['alerta']:.2f}
horas")
                print(f"   Tempo até limite crítico: {times['critico']:.2f}
horas")

    if __name__ == "__main__":
        main()
```

Ficheiro 'Análise Temporal Aeronave.py':

```
    import xlwings as xw
    import sympy as sp
    import numpy as np
    import matplotlib
    matplotlib.use('Agg')
    import matplotlib.pyplot as plt
    from scipy.interpolate import interp1d

    t = sp.Symbol('t')

    def read_expression(expr, reliability_funcs):

        def evaluate(expr):
            expr = str(expr).strip()

            operator = '|'
            if operator in expr:
                parts = expr.split(operator)
                print(f"Componentes em série: {parts}")

                reliabilities = [evaluate(part.strip()) for part in parts]
                result = sp.S.One
                for reliability in reliabilities:
                    result *= reliability

                print(f"Fiabilidade em série calculada: {result}")
                return sp.simplify(result)

            if expr in reliability_funcs:
                result = reliability_funcs[expr]
                print(f"Componente básico {expr}: {result}")
                return result

            try:
                result = sp.sympify(expr)
                print(f"Erro: Componente {expr} não encontrado")
                return result
```

```
            except Exception as e:
                raise  ValueError(f"Erro  ao  processar  componente:  {expr}.
Erro: {e}")

        return evaluate(expr)

    def get_events_data(sheet):

        events = []
        row = 4

        while True:
            event_id = sheet.range(f'B{row}').value
            mtbf = sheet.range(f'D{row}').value

            if not event_id:
                break

            if event_id and mtbf:
                try:
                    mtbf_float = float(mtbf)
                    events.append({'id': str(event_id), 'mtbf': mtbf_float})
                    print(f"Evento {event_id} : MTBF = {mtbf_float}")
                except ValueError:
                    print(f" MTBF inválido: Evento {event_id}")

            row += 1

        return events

    def reliability_event_functions(events):

        reliability_funcs = {}
        for event in events:
            if event['mtbf']:
                reliability_funcs[str(event['id'])]     =     sp.exp(-t     /
event['mtbf'])
                print(f"Fiabilidade calculada para {event['id']} com MTBF =
{event['mtbf']}")
        return reliability_funcs


    def system_reliability(sheet):

        cutset_expr = sheet.range('E4').value.replace(' ', '')
        print(f"Expressão do cut set: {cutset_expr}")

        events = get_events_data(sheet)

        reliability_funcs = reliability_event_functions(events)
        return read_expression(cutset_expr, reliability_funcs)

    def    calculate_failure_probabilities(system_reliability,    max_time,
num_points=1000):

        t_values = np.linspace(0, max_time, num_points)
        failure_values = []

        for t_val in t_values:
```

```
            try:
                reliability_at_t = float(system_reliability.subs(t, t_val))
                failure_at_t = 1 - reliability_at_t
                failure_values.append(failure_at_t)
            except Exception as e:
                failure_values.append(None)

        return t_values, failure_values

    def       find_critical_times_numerical(t_values,       failure_values,
threshold=1e-9):

        valid_points = [(t, f) for t, f in zip(t_values, failure_values) if
f is not None]
        if not valid_points:
            return None, None

        t_values_clean, failure_values_clean = zip(*valid_points)

        failure_interp   =   interp1d(failure_values_clean,   t_values_clean,
bounds_error=False, fill_value=np.inf)

        time_threshold = failure_interp(threshold)
        if np.isinf(time_threshold):
            print(f"Não foi encontrado tempo para F(t) > {threshold}")
            time_threshold = None
        else:
            print(f"Tempo  para  F(t)  >  {threshold}:  {time_threshold:.2f}
horas")

        # Find time for P_failure = 1
        time_failure_one = failure_interp(0.999)
        if np.isinf(time_failure_one):
            print(f"Não foi encontrado tempo para F(t) = 1")
            time_failure_one = None
        else:
            print(f"Tempo para F(t) ≈ 1: {time_failure_one:.2f} horas")

        return time_threshold, time_failure_one

    def  plot_failure_probability(t_values,  failure_values,  critical_times,
output_path):

        plt.figure(figsize=(12, 8))

        # Plot failure probability curve
        plt.plot(t_values, failure_values,
                 label="Probabilidade de Falha Catastrófica da Aeronave",
                 color="black",
                 linewidth=2)

        time_threshold, time_failure_one = critical_times
        if time_threshold is not None:
            plt.axvline(x=time_threshold,   color='black',   linestyle='-.',
alpha=0.5,
                        label=f'Tempo      para      F(t)      >      10^-9
({time_threshold:.0f}h)')
        if time_failure_one is not None:
```

```
            plt.axvline(x=time_failure_one,  color='black',  linestyle='--',
alpha=0.5,
                        label=f'Tempo       para       F(t)       ≈       1
({time_failure_one:.0f}h)')

        plt.xlabel("Horas de Voo Acumuladas", fontsize=12)
        plt.ylabel("Probabilidade de Falha", fontsize=12)
        plt.grid(True, linestyle='--', alpha=0.7)
        plt.legend(fontsize=10)
        plt.ylim(0, 1.1)

        plt.savefig(output_path, dpi=300, bbox_inches='tight')
        plt.close()


    def main():
        file_path = "Análise-Fiabilidade.xlsm"
        app = xw.App(visible=False)
        wb = app.books.open(file_path)

        try:
            sheet = wb.sheets["Análise Temporal Aeronave"]

            system_reliability_expr = system_reliability(sheet)

            max_time = 200000
            t_values, failure_values = calculate_failure_probabilities(
                system_reliability_expr,
                max_time
            )

            critical_times     =     find_critical_times_numerical(t_values,
failure_values)

            sheet.range("F4").value = critical_times[0]
            sheet.range("G4").value = critical_times[1]


            output_path = "Probabilidade Falha Catastrófica vs Tempo.png"
            plot_failure_probability(t_values,               failure_values,
critical_times, output_path)

            print(f"Gráfico salvo em: {output_path}")

        except Exception as e:
            print(f"Erro: {e}")

        finally:
            wb.save()
            wb.close()
            app.quit()

    if __name__ == "__main__":
        main()
```

Apêndice B

Resultados de importância e sensibilidade para cada sistema analisado.

Sistema A (34-60 FLIGHT MANAGEMENT COMPUTING):

Upper Bound Approximation = $1 - (1 - A1 \cdot A2 \cdot A3 \cdot A4 \cdot A5 \cdot A6)$

Tabela 19 - Parâmetros de importância para o sistema A

Fonte: (autor, 2025)

Eventos	BIR	CRIT	FV	RRW	RAW
A1	0	1	1	∞	25000
A2	0	1	1	∞	25000
A3	0	1	1	∞	25000
A4	0	1	1	∞	25000
A5	0	1	1	∞	30000
A6	0	1	1	∞	30000

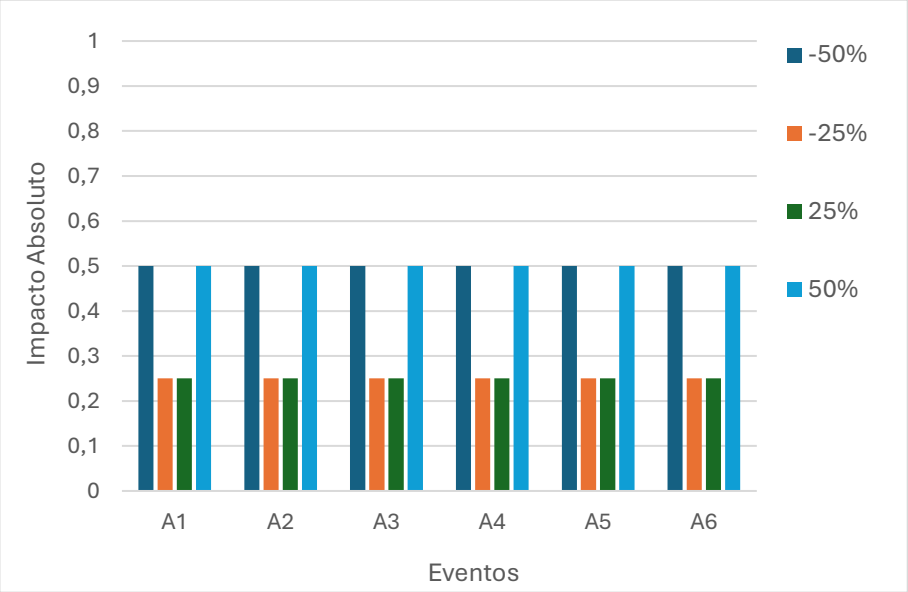


Figura 55 - Impacto absoluto dos eventos do sistema A

Fonte: (autor, 2025)

Sistema B (34 NAVIGATION):

Upper Bound Approximation = $1 - (1 - \mathrm{B1} \cdot \mathrm{B2}) \cdot (1 - \mathrm{B3} \cdot \mathrm{B4})$

Tabela 20 - Parâmetros de importância para o sistema B

Fonte: (autor, 2025)

Eventos	BIR	CRIT	FV	RRW	RAW
B1	$5 \cdot 10^{-5}$	0.692	0.692	3.25	13846.7
B2	$5 \cdot 10^{-5}$	0.692	0.692	3.25	13846.7
B3	$3.3 \cdot 10^{-5}$	0.307	0.307	1.44	9231.7
B4	$3.3 \cdot 10^{-5}$	0.307	0.307	1.44	9231.7

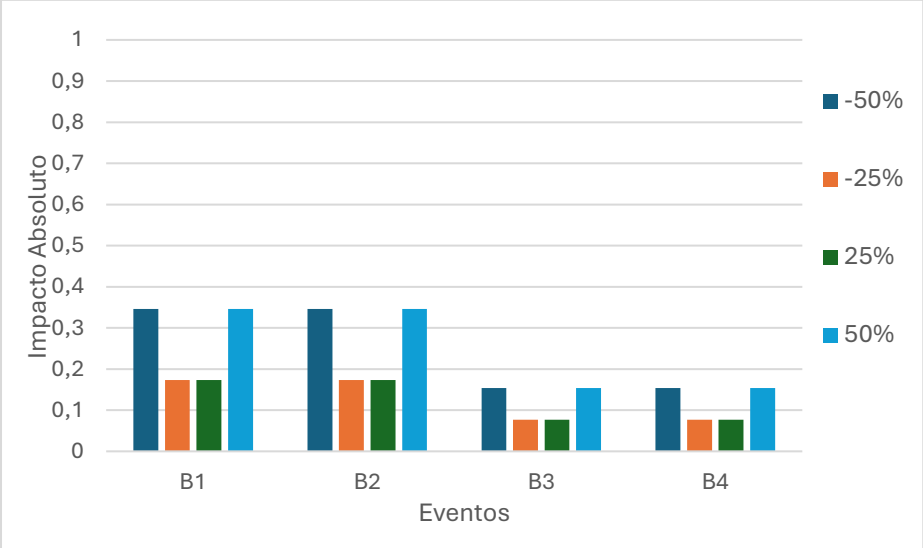


Figura 56 - Impacto absoluto dos eventos do sistema B

Fonte: (autor, 2025)

Sistema C (72.1 ENGINE):

Upper Bound Approximation = $1 - (1 - \mathrm{C1} \cdot \mathrm{C2}) \cdot (1 - \mathrm{C3} \cdot \mathrm{C4})$

Tabela 21 - Parâmetros de importância para o sistema C

Fonte: (autor, 2025)

Eventos	BIR	CRIT	FV	RRW	RAW
C1	$10 \cdot 10^{-6}$	0.17	0.17	1.20	16970
C2	$10 \cdot 10^{-6}$	0.17	0.17	1.20	16970
C3	$2.21 \cdot 10^{-5}$	0.83	0.83	5.89	37536
C4	$2.21 \cdot 10^{-5}$	0.83	0.83	5.89	37536

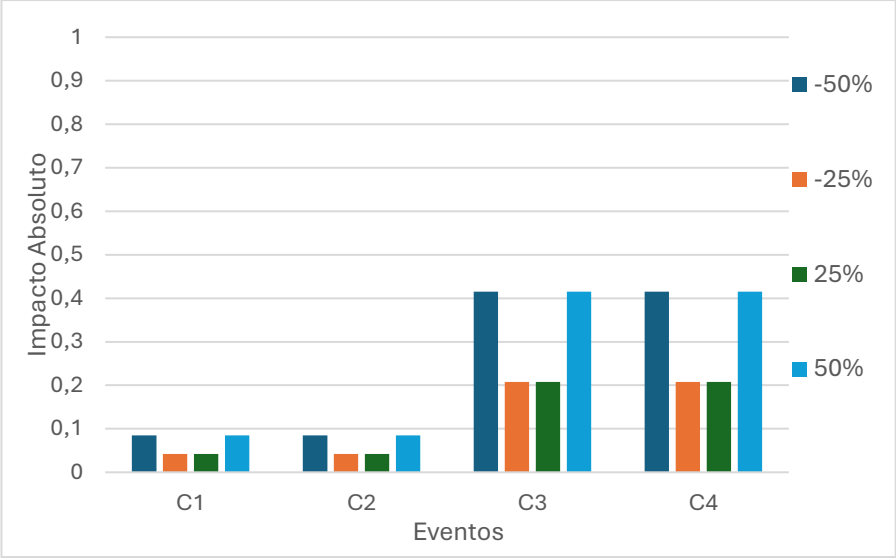


Figura 57 - Impacto absoluto dos eventos do sistema C

Fonte: (autor, 2025)

Sistema D (24 ELECTRICAL POWER):

Upper Bound Approximation = $1 - (1 - \mathrm{D1}) \cdot (1 - \mathrm{D2} \cdot \mathrm{D3})$

Tabela 22 - Parâmetros de importância para o sistema D

Fonte: (autor, 2025)

Eventos	BIR	CRIT	FV	RRW	RAW
C	1	0.849	0.849	6.626	∞
D1	$1.02 \cdot 10^{-5}$	0.151	0.151	1.178	14747
D2	$1.02 \cdot 10^{-5}$	0.151	0.151	1.178	14747

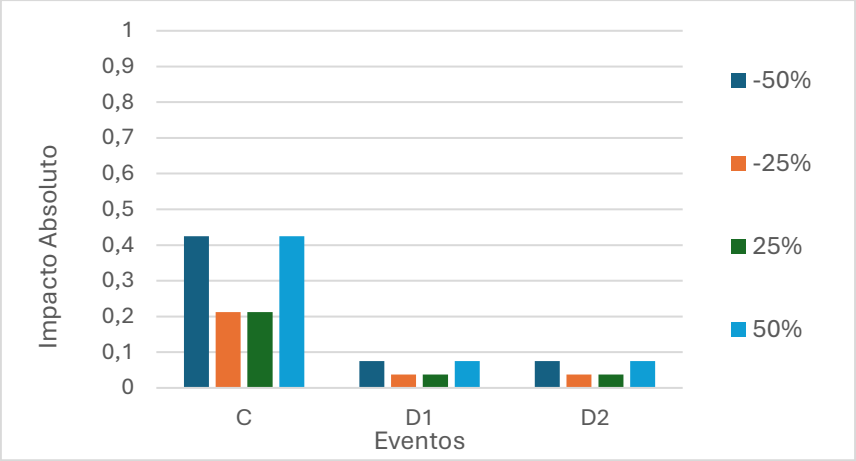


Figura 58 - Impacto absoluto dos eventos do sistema D

Fonte: (autor, 2025)

Sistema E (FMS):

Upper Bound Approximation = $1 - (1 - \mathrm{E1}) \cdot (1 - \mathrm{E2})$

Tabela 23 - Parâmetros de importância para o sistema E

Fonte: (autor, 2025)

Eventos	BIR	CRIT	FV	RRW	RAW
A	1	0	0	1	∞
D	1	1	1	∞	∞

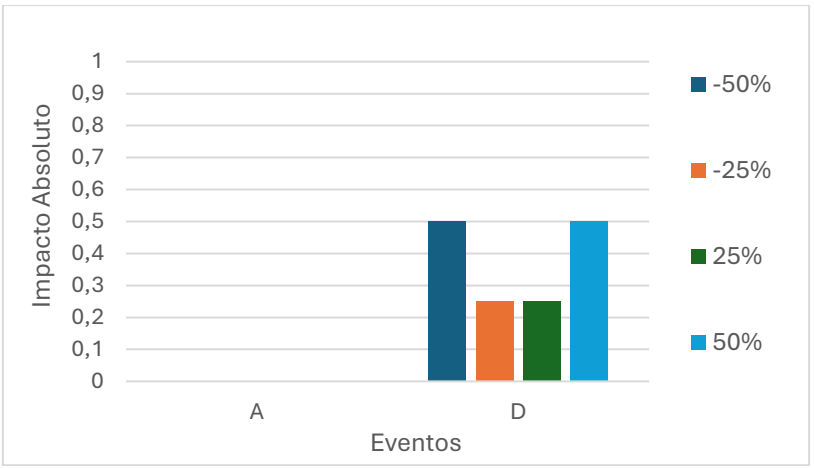


Figura 59 - Impacto absoluto dos eventos do sistema E

Fonte: (autor, 2025)

Sistema F (AVIONICOS):

Upper Bound Approximation = $1 - (1 - \mathrm{F1}) \cdot (1 - \mathrm{F2})$

Tabela 24 - Parâmetros de importância para o sistema F

Fonte: (autor, 2025)

Eventos	BIR	CRIT	FV	RRW	RAW
B	1	0.84	0.84	6.2	∞
D	1	0.16	0.16	1.2	∞

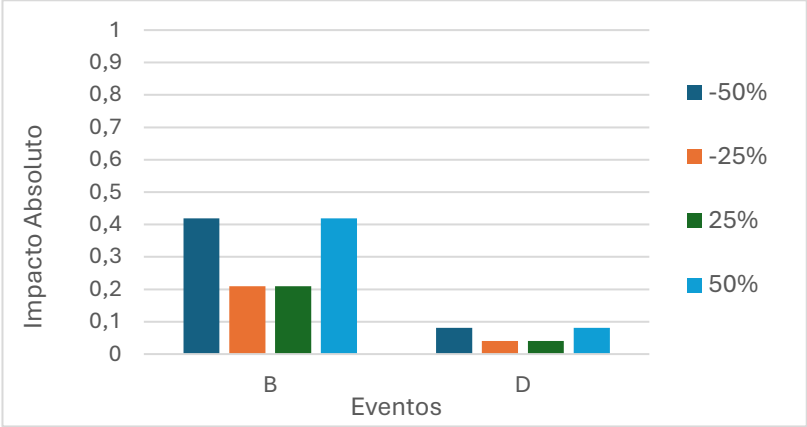


Figura 60 - Impacto absoluto dos eventos do sistema F

Fonte: (autor, 2025)

Sistema G (SUPERFICIES SUSTENTADORAS):

Upper Bound Approximation = $1 - (1 - \mathrm{G1}) \cdot (1 - \mathrm{G2})$

Tabela 25 - Parâmetros de importância para o sistema G

Fonte: (autor, 2025)

Eventos	BIR	CRIT	FV	RRW	RAW
D	1	0.54	0.54	2.2	∞
C	1	0.46	0.46	1.85	∞

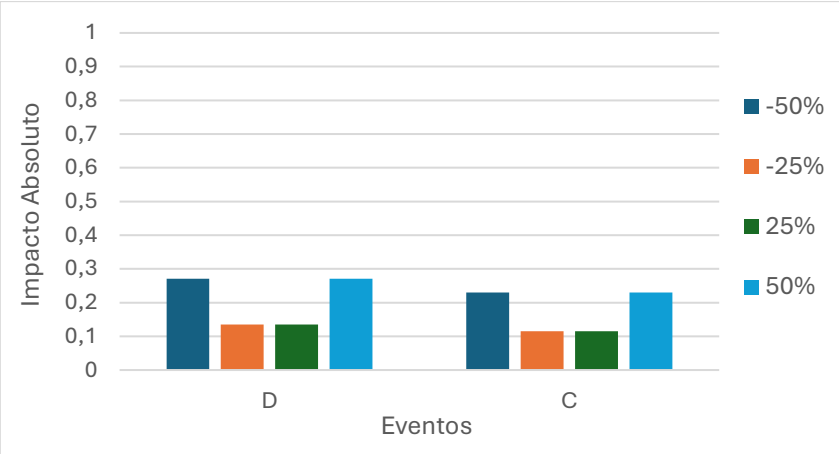


Figura 61 - Impacto absoluto dos eventos do sistema G

Fonte: (autor, 2025)

Apêndice C

Análise BIR=RRW=CRIT=33.3%

Sistema A (34-60 FLIGHT MANAGEMENT COMPUTING):

Figura 62 - Valores de importância dos eventos do sistema A

Fonte: (autor, 2025)

Sistema B (34 NAVIGATION):

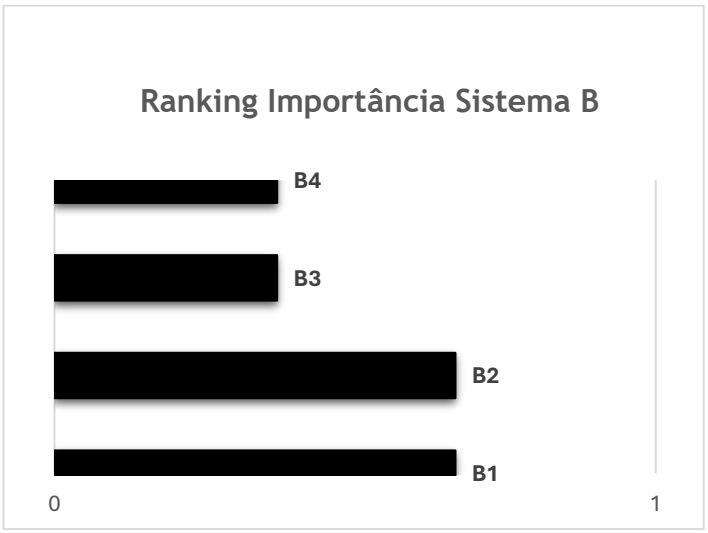


Figura 63 - Valores de importância dos eventos do sistema B

Fonte: (autor, 2025)

Sistema C (72.1 ENGINE):

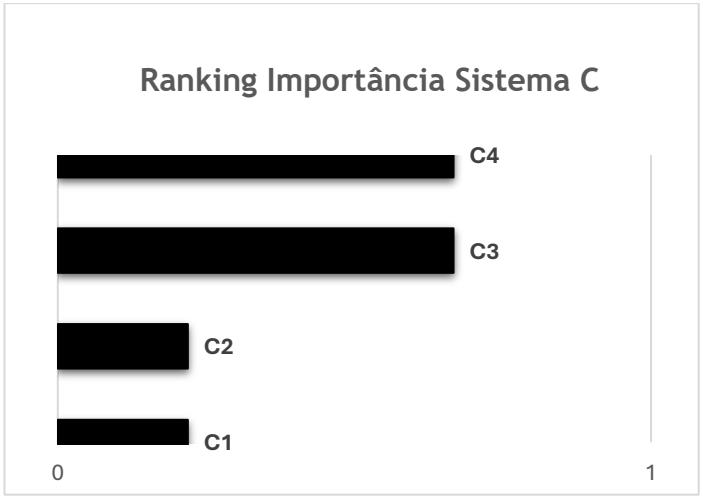


Figura 64 - Valores de importância dos eventos do sistema C

Fonte: (autor, 2025)

Sistema D (24 ELECTRICAL POWER):

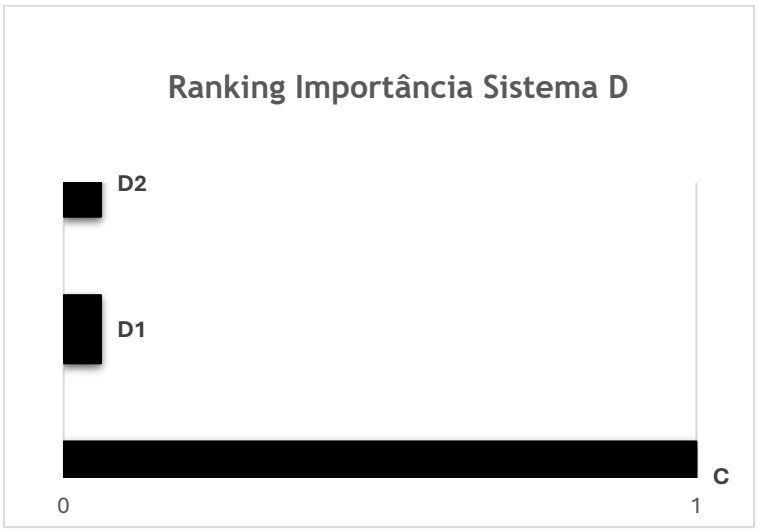


Figura 65 - Valores de importância dos eventos do sistema D

Fonte: (autor, 2025)

Sistema E (FMS):

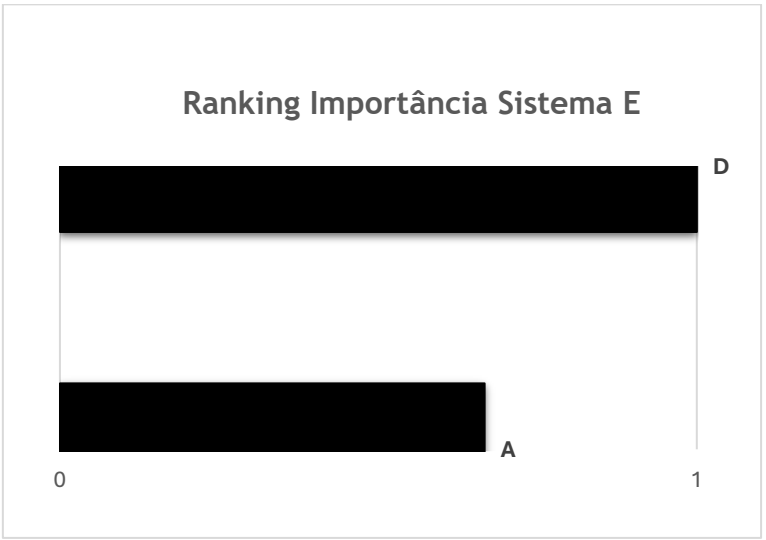


Figura 66 - Valores de importância dos eventos do sistema E

Fonte: (autor, 2025)

Sistema F (AVIONICOS):

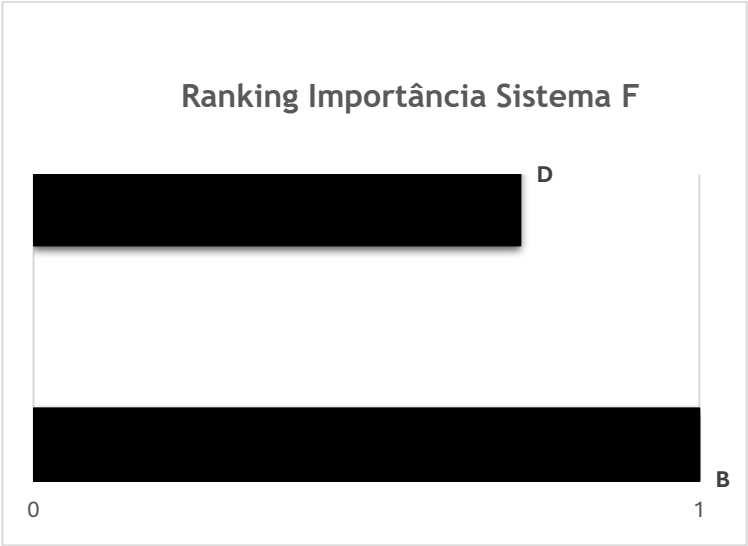


Figura 67 - Valores de importância dos eventos do sistema F

Fonte: (autor, 2025)

Sistema G (SUPERFICIES SUSTENTADORAS):

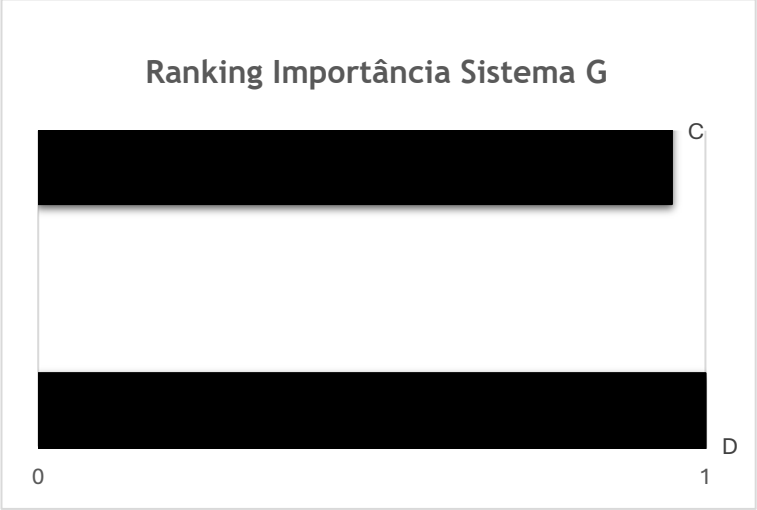


Figura 68 - Valores de importância dos eventos do sistema G

Fonte: (autor, 2025)

Apêndice D

Considerando um componente que segue uma distribuição exponencial o valor de MTBF do componente é equivalente ao tempo médio esperado sem falhas (ET) dado pela seguinte equação (Birolini, 2014, p. 6):

$$MTBF = ET = \int_0^\infty R(t)dt = \int_0^\infty t \cdot f(t)\, dt$$

Ao submeter o componente a um critério de *hard time*, onde ações de manutenção preventiva são realizadas a cada T horas de operação, de acordo com a teoria de renovação (Sigman, 2009, p. 15), o valor de ET é dado por:

$$E(T) = \int_0^T t \cdot f(t)\, dt \; + \; T \cdot R(T)$$

Onde $\int_0^T t \cdot f(t)\, dt$ representa a contribuição esperada do tempo de falha (t) quando esta ocorre antes de T. É o valor médio ponderado pelo tempo em que as falhas ocorrem dentro do intervalo [0, T].

O termo $T \cdot \; R(T)$ representa o contributo do limite T para os casos em que o componente não falha antes de T. Como $R(T) = 1 - F(T)$, este é o complemento da probabilidade cumulativa F(T), ou seja, a probabilidade de não falha até T. Em termos de fiabilidade, isto reflete o contributo do tempo máximo T nos casos em que o sistema permanece funcional até T.

Ao resolver E(T) é obtida a seguinte equação:

$$E(T) = \int_0^T R(t)dt = MTBF_0 \cdot \big(1 - R(T)\big) \tag{43}$$

Onde $MTBF_0$ representa o valor original de MTBF do componente.

Neste contexto, E(T) representa o valor esperado da distribuição truncada, indicando a vida útil do componente que foi "interrompida" pela implementação da estratégia de manutenção preventiva. Esta interrupção resulta numa redução do valor do MTBF do componente.

E(T) e MTBF são conceptualmente semelhantes, pois ambos representam médias: E(T) mede o tempo médio até à primeira intervenção (seja por falha ou manutenção preventiva),

enquanto o MTBF mede o tempo médio entre intervenções numa perspetiva de longo prazo. Dada esta similaridade conceptual, considera-se que E(T) = $MTBF_1$.

Teoricamente, esta igualdade sugere que o novo MTBF do componente seria inferior ao original, visto que as ações de manutenção reduzem o tempo efetivo de operação do componente. No entanto, esta interpretação não traduz adequadamente o impacto positivo da manutenção preventiva na fiabilidade global do sistema. Para traduzir corretamente este impacto no programa de análise de fiabilidade, utiliza-se o inverso da equação $E(T) = \int_0^T R(t)dt = MTBF_0 \cdot \big(1 - R(T)\big)(43)$.

Esta abordagem inverte essencialmente a equação: em vez de quantificar o tempo de vida "perdido", procura-se determinar o tempo de vida efetivamente ganho através da estratégia de manutenção preventiva. Por exemplo, se a manutenção preventiva reduzir em 50% as falhas potenciais do componente, isto implica que cada componente necessitará de duas vezes mais substituições comparativamente ao cenário sem manutenção preventiva. Consequentemente, o novo valor de MTBF será dado pelo MTBF original dividido pela proporção de redução das falhas potenciais, então:

$$MTBF_1 = \frac{MTBF_0}{1 - R(T)}$$

Onde $MTBF_1$ representa o valor efetivo de MTBF considerando o impacto da manutenção preventiva e $MTBF_0$ representa o valor original de MTBF do componente.