

Instituto Superior Técnico

2023/2024

Ensaios em Voo

Trabalho experimental envolvendo processamento e análise de dados

Grupo 2:

Leonor Alves - 102845

Paulo Campos - 103042

Lourenço Gouveia Faria - 103354

Professor Agostinho Rui Alves da Fonseca

junho de 2024

Conteúdo

Conteúdo	1
1 Introdução	1
2 EV_2024.02A: Perguntas 1 - 4	2
2.1 Análise temporal das grandezas medidas	2
2.2 Espectro unilateral de amplitude	2
2.3 Sinais e frequências com picos significativos de amplitude	3
3 EV_2024.02B: Perguntas 5 - 11	5
3.1 Análise temporal dos dados de voo	5
3.2 Análise temporal da aceleração vertical	6
3.3 Extremos relativos das acelerações verticais	6
3.4 Algoritmo para detecção de ciclos de aceleração vertical	7
4 EV_2024.02C: Perguntas 12 - 16	10
4.1 Erros do sistema de navegação (HPE e VPE)	10
4.2 Análise temporal dos erros dos sistema, dos limites de proteção e do número de satélites do sistema de navegação	11
4.3 Parâmetros de Desempenho	12
4.3.1 Exatidão	12
4.3.2 Integridade	12
4.3.3 Disponibilidade	13
4.3.4 Continuidade	13
4.4 Eventos de Integridade e Conclusões	14
5 Conclusão	16
Referências	17
A Anexos - Código	18
A.1 EV_2024.02A	18
A.2 EV_2024.02B	20
A.3 EV_2024.02C	24

1 Introdução

No âmbito da unidade curricular de Ensaaios em Voo desenvolveu-se um trabalho experimental envolvendo o processamento e análise de dados obtidos durante a realização de ensaios em voo. Estes dados são recolhidos com recurso a diferentes equipamentos durante os ensaios de várias aeronaves.

Ao longo deste trabalho vão ser analisados 3 ficheiros de dados com recurso ao *MATLAB*. O ficheiro EV_2024.02A contém dados obtidos em laboratório com recurso a um sistema de instrumentação PCM de Classe I, de acordo com o estipulado no Capítulo 4 das normas RCC IRIG 106-23. O ficheiro EV_2024.02B apresenta dados obtidos em voo numa aeronave DORNIER-DASSAULT ALPHA-JET e o ficheiro EV_2024.02C contém dados recolhidos num voo de calibração de um sistema convencional de navegação.

De realçar, que de modo a analisar os ficheiros no *MATLAB*, estes foram convertidos em *txt*:

- "EV_2024.02A" → "EV_2024_A.txt";
- "EV_2024.02B" → "EV_2024_B.txt";
- "EV_2024.02C" → "EV_2024_C.txt".

2 EV_2024.02A: Perguntas 1 - 4

Como referido anteriormente, o ficheiro EV_2024.02A contém dados obtidos em laboratório e utilizando um sistema de instrumentação PCM de Classe I, de acordo com o estipulado no Capítulo 4 das normas RCC IRIG 106-23 [1]. Desta forma, ao longo desta secção iremos fazer o tratamento das grandezas de tempo (t) e de aceleração (a_1, a_2, a_3, a_4).

2.1 Análise temporal das grandezas medidas

De forma a analisar a evolução temporal das acelerações, desenvolveram-se os gráficos de cada grandeza em função do tempo que se encontram representados em seguida.

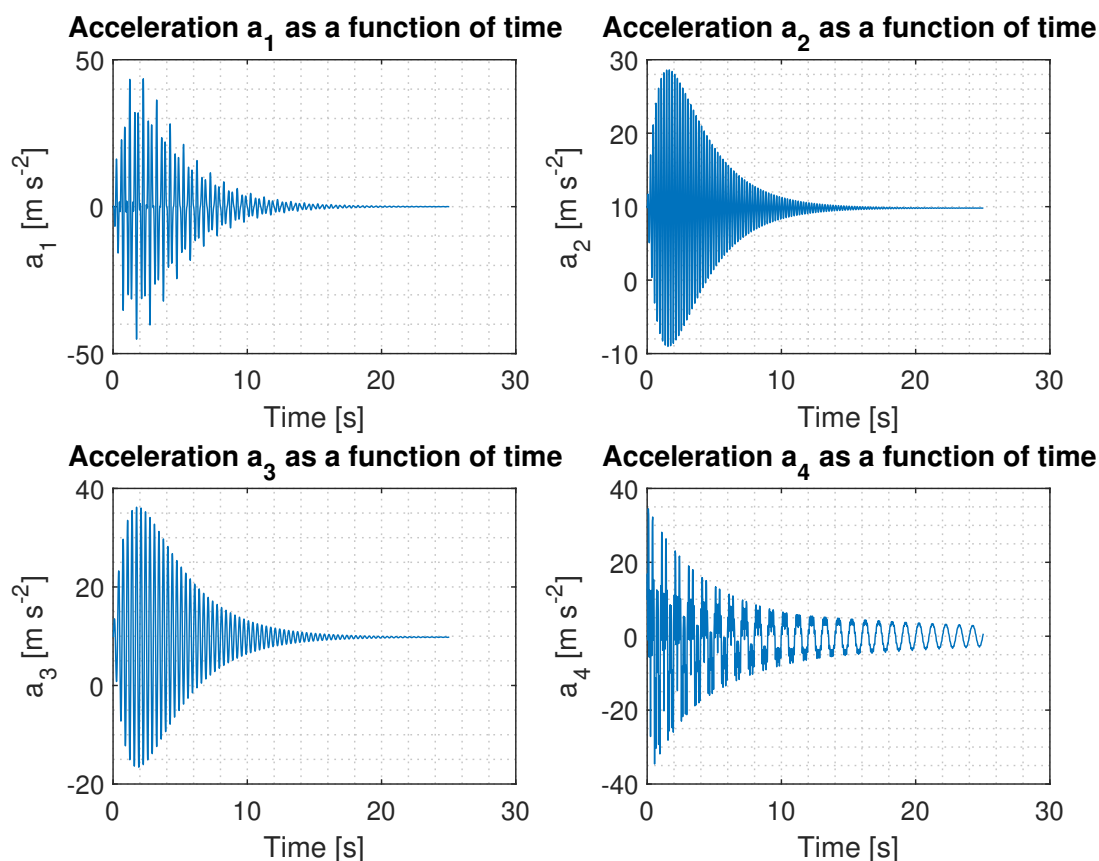


Figura 1: Evolução temporal das acelerações medidas

Através da observação dos gráficos é possível constatar que as acelerações são oscilatórias amortecidas, tendendo para zero as 3 primeiras.

2.2 Espectro unilateral de amplitude

Na figura 2 encontram-se representados os espectros unilaterais de amplitude [2] de cada um dos 4 sinais, nos quais a frequência se encontra em Hz.

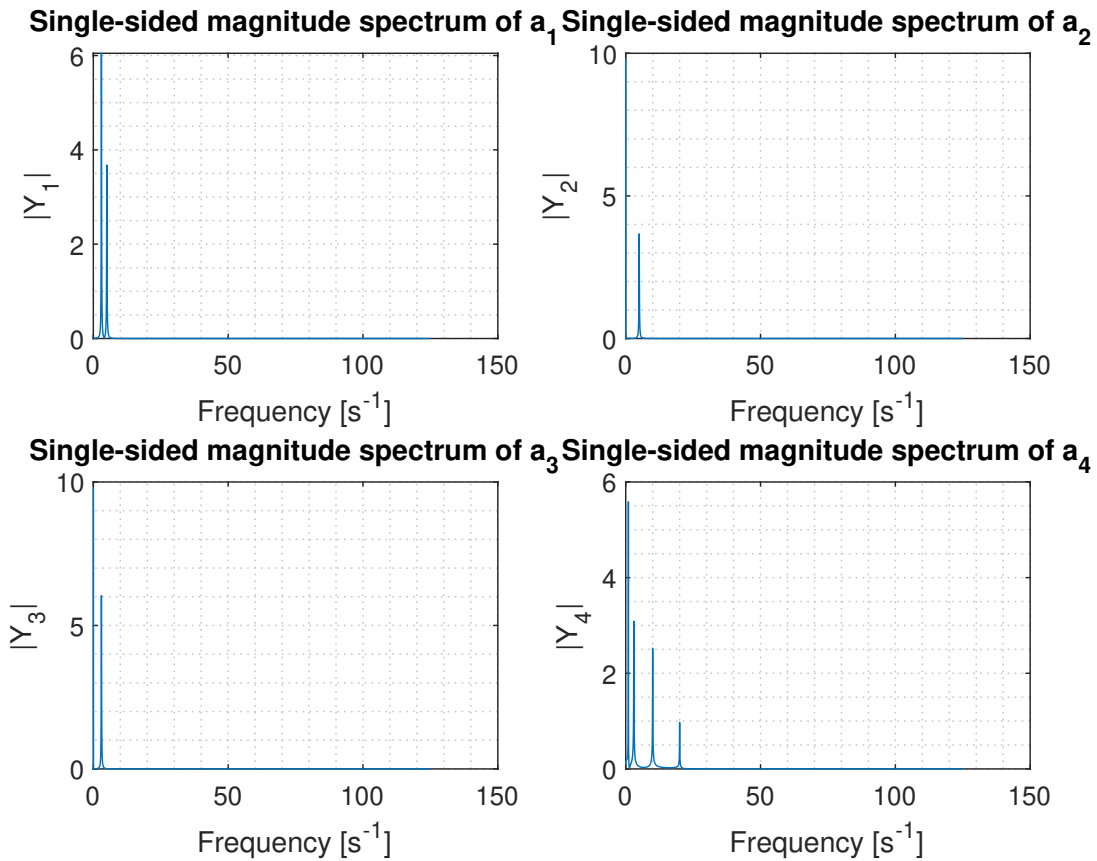


Figura 2: Espectro unilateral de amplitude

2.3 Sinais e frequências com picos significativos de amplitude

Para cada sinal, a_1, a_2, a_3 e a_4 , existem certas frequências às quais correspondem picos de amplitude. Estes valores serão apresentados de acordo com o sinal ao longo das tabelas seguintes.

Sinal	a_1	
Frequência (Hz)	2.994	4.990
Amplitude (m/s^2)	6.047	3.679

Tabela 1: Frequência e amplitude dos picos de amplitude do sinal a_1

Sinal	a_2	
Frequência (Hz)	0	4.990
Amplitude (m/s^2)	9.807	3.669

Tabela 2: Frequência e amplitude dos picos de amplitude do sinal a_2

Sinal	a_3	
Frequência (Hz)	0	2.994
Amplitude (m/s^2)	9.807	6.040

Tabela 3: Frequência e amplitude dos picos de amplitude do sinal a_3

Sinal	a_4			
Frequência (Hz)	0.9981	2.994	10.021	20.002
Amplitude (m/s^2)	5.584	3.088	2.472	0.9715

Tabela 4: Frequência e amplitude dos picos de amplitude do sinal a_4

A partir da análise dos gráficos presentes na figura 2, foi possível retirar os valores de frequências e amplitude nos quais se registam picos de amplitude. Estes picos demonstram que o sinal de cada aceleração foi recebido pelo sistema de instrumentação.

No caso da aceleração 1, o sinal foi recebido 2 vezes, às frequências de 2.994Hz e de 4.990Hz, tendo a primeira uma amplitude superior de $6.047m/s^2$. A aceleração 2 também foi detetada 2 vezes, a 0Hz, com a amplitude mais alta de $9.807m/s^2$, e a 4.990Hz. Em relação à aceleração 3, o sinal foi recebido às frequências de 0Hz e 2.994Hz, tendo a primeira frequência uma amplitude de $9.807m/s^2$. Por fim, o sinal da a aceleração 4 foi detetado 4 vezes para as frequências 0.998Hz, 2.994Hz, 10.021Hz e 20.002Hz. Para a frequência mais baixa foi recebido o maior pico de amplitude de $5.584m/s^2$.

Analisando os resultados obtidos, é possível concluir que o maior pico de magnitude se dá a menores frequências, diminuindo à medida que a frequência aumenta. Em relação às restantes frequências não mencionadas, é considerado que a amplitude do sinal é aproximadamente zero, o que significa que não há ruído na transmissão.

3 EV_2024.02B: Perguntas 5 - 11

No segundo ficheiro de dados deste projeto encontram-se dados referentes à aeronave *Dornier-Dassault Alpha-Jet* ao longo de um voo. Dos diversos dados recolhidos, listam-se alguns:

- t - tempo expresso em segundos (s)
- EAS - velocidade ar equivalente (Equivalent Airspeed - EAS) expressa em knots (kn)
- QNE - altitude barométrica QNE expressa em feet (ft)
- a_z - aceleração vertical expressa em metros por segundo ao quadrado (m/s^2)
- $N2_{rh}$ - velocidade de rotação N2 do motor direito, expressa em percentagem do regime de funcionamento nominal
- EGT_{rh} - temperatura dos gases de escape (Exhaust Gas Temperature - EGT) do motor direito, expressa em kelvin (K)
- FF_{rh} - consumo de combustível (Fuel Flow - FF) do motor direito, expresso em quilogramas por hora (kg/h)
- $N2_{lt}$ - velocidade de rotação N2 do motor esquerdo, expressa em percentagem do regime de funcionamento nominal
- EGT_{lt} - temperatura dos gases de escape (Exhaust Gas Temperature - EGT) do motor esquerdo, expressa em kelvin (K)
- FF_{lt} - consumo de combustível (Fuel Flow - FF) do motor esquerdo, expresso em quilogramas por hora (kg/h)

3.1 Análise temporal dos dados de voo

Para possibilitar uma análise da sua variação temporal, construíram-se os 9 gráficos seguintes em função do tempo:

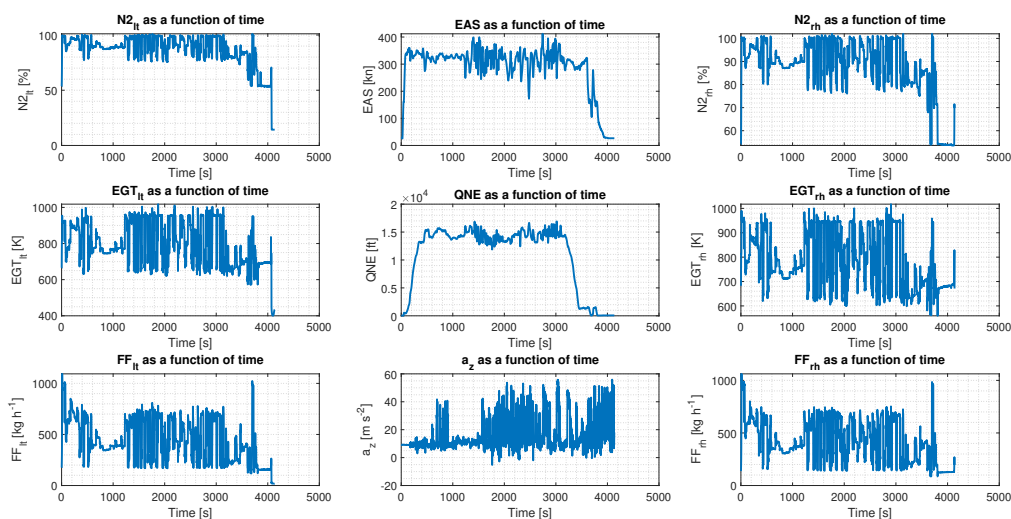


Figura 3: Variação temporal dos dados de voo recolhidos

A coluna da esquerda da figura 4 corresponde a dados do motor esquerdo da aeronave, a coluna do meio a dados de voo genéricos da aeronave e a coluna da direita a dados do motor direito da aeronave.

3.2 Análise temporal da aceleração vertical

O gráfico central inferior corresponde às acelerações verticais a que a aeronave esteve sujeita durante o voo e encontram-se expressas em m/s^2 . De modo a adimensionalizar estes valores, considerou-se a aceleração gravítica como valor de referência $1G = 9.80665 m/s^2$, aceleração da gravidade ao nível médio das águas do mar numa atmosfera padrão ICAO.

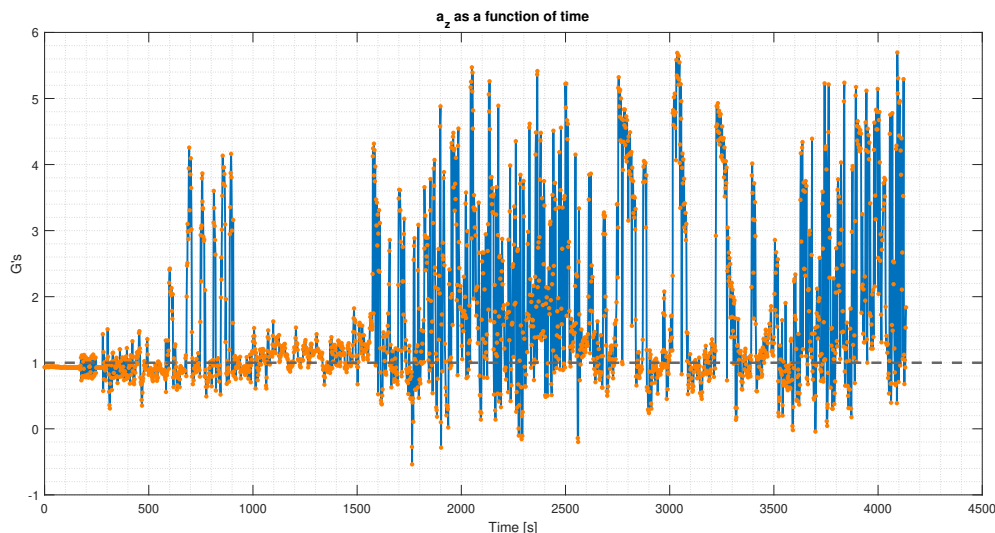


Figura 4: Variação temporal da aceleração vertical em G's

A linha horizontal representada a tracejado representa o valor de referência para 1G e os pontos a laranja as leituras de aceleração efetuadas em cada momento do voo.

3.3 Extremos relativos das acelerações verticais

Seguidamente procedeu-se ao armazenamento dos dados de aceleração vertical que correspondem aos extremos relativos, isto é, picos e vales dos valores observados na secção anterior. No total, registam-se 1274 extremos relativos, sendo que uma metade corresponde a 637 picos e a outra a 637 vales.

Abaixo encontra-se uma tabela que serve como excerto dos dados de um ficheiro com estes extremos a ser entregue em conjunto com este relatório:

Tempo [s]	4.002	6.003	8.004	10.005	16.008	18.009	...
Aceleração vertical [G's]	0.937	0.932	0.9417	0.936	0.941	0.934	...

Tabela 5: Excerto de dados dos extremos relativos da aceleração vertical em G's

Como pode ser visto, os valores referentes às acelerações intercalam entre subida e descida até ao fim. Realizou-se também uma representação deste dados:

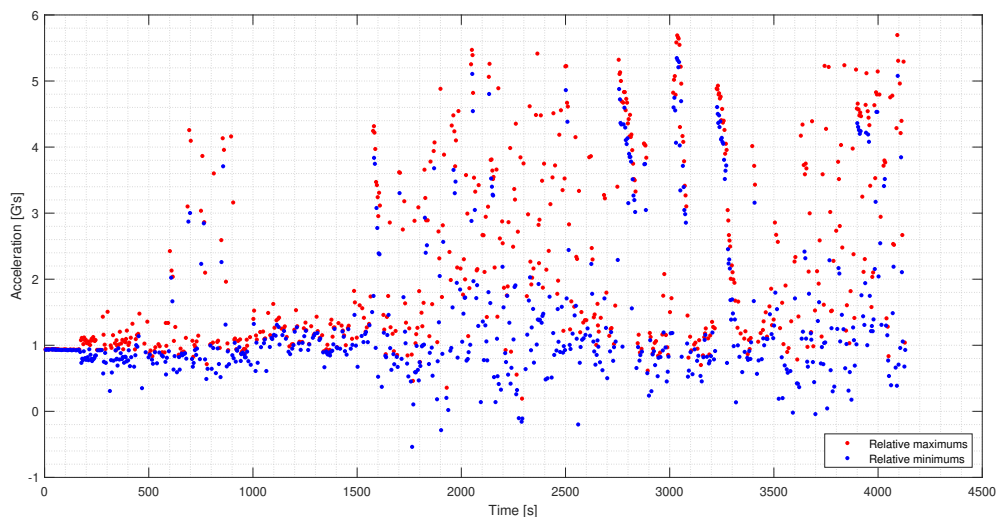


Figura 5: Representação gráfica dos extremos relativos máximos e mínimos da aceleração vertical

Como detalhe curioso, é possível observar que ligando todos os extremos relativos máximos e os extremos relativos mínimos entre si, em nenhuma situação haverá cruzamento das linhas que interligam estes pontos:

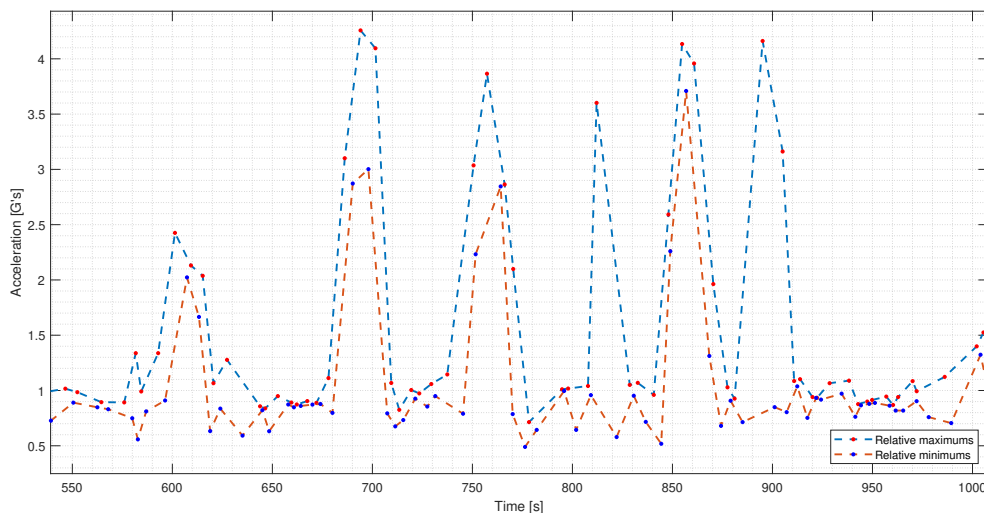


Figura 6: Representação gráfica ampliada dos extremos relativos máximos e mínimos da aceleração vertical

3.4 Algoritmo para detecção de ciclos de aceleração vertical

O objetivo desta secção é desenvolver um algoritmo que permita a contagem de ocorrência de ciclos de aceleração vertical. O processo de contagem de uma ocorrência de um ciclo de valor N_1 é iniciado quando a aceleração vertical ultrapassa o valor N_1 ($a_z > N_1$ se $N_1 > 1G$ ou $a_z < N_1$ se $N_1 < 1G$) e é concluído quando o valor da aceleração vertical ultrapasse o valor N_2 ($a_z < N_2$ se $N_1 > 1G$ ou $a_z > N_2$ se $N_1 < 1G$).

O design do algoritmo foi feito através de um sistema de condições que verificam se o valor da aceleração está de acordo com as *flags* mencionadas no parágrafo anterior. Em caso afirmativo, é acionada uma variável booleana para 1 e que só transita para 0 quando a segunda condição for respeitada. Deste modo, evita-se a perda de informação referente ao início da contagem ciclos que começaram antes de um anterior ter acabado.

```

function flag_status = acc_cicle(N_1, N_2, acc, time)

    flag = false;
    count = 0;

    for iter=1:size(acc,1)
        if (((acc(iter)>N_1) && (N_1>1)) || ((acc(iter)<N_1) && (N_1<1))) && (flag == false))
            count = count + 1;
            flag_status(count, :) = [0, 0, 0, 0];
            flag = true;
            flag_status(count, 1) = time(iter);
            flag_status(count, 2) = acc(iter);
        end
        if (((acc(iter)<N_2) && (N_1>1)) || ((acc(iter)>N_2) && (N_1<1))) && (flag == true))
            flag = false;
            flag_status(count, 3) = time(iter);
            flag_status(count, 4) = acc(iter);
        end
    end
    if count == 0
        flag_status = [];
    elseif flag_status(count,3) == 0
        flag_status(count,:) = [];
    end
end

```

Pretende-se aplicar este algoritmo com as *flags* N_1 e N_2 apresentadas na tabela 6:

Flag	1	2	3	4	5	6	7	8
N_1	2.5	4.0	5.0	6.0	7.0	0.0	-1.5	-2.5
N_2	2.2	3.7	4.7	5.7	6.7	0.3	-1.2	-2.2

Tabela 6: *flags* N_1 e N_2 para início e fim de contagem de ciclos de aceleração vertical

Aplicando o algoritmo para os diversos valores, obtiveram-se os resultados apresentados na tabela abaixo:

Flag	1	2	3	4	5	6	7	8
Nº de ciclos	125	60	18	0	0	7	0	0

Tabela 7: Números de ciclos para cada *flag*

De forma a ilustrar o comportamento da deteção em alguns dos dados, contruiu-se o gráfico seguinte. Nesta porção dos dados, é notório que as duas primeiras *flags* detetam os ciclos da forma que é esperada:

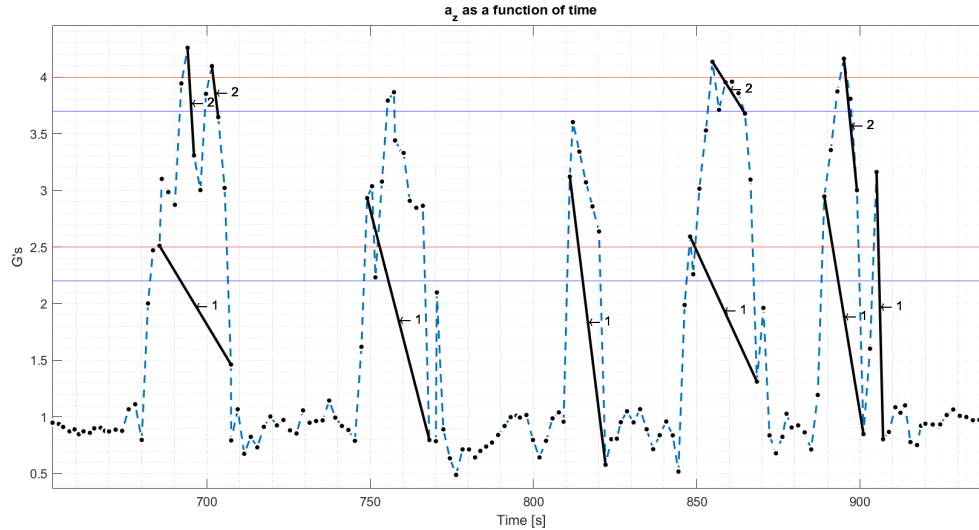


Figura 7: Representação gráfica da aplicação do algoritmo para as *flags* da tabela 6 numa porção dos dados

A vermelho encontra-se representado o início da contagem de cada ciclo correspondente aos valores N_1 e a azul o fim do ciclo correspondente aos valores N_2 .

Aplicaram-se as mesmas *flags* para os dados referentes à subsecção 3.3 e os resultados foram iguais no que toca à quantização de ciclos:

Flag	1	2	3	4	5	6	7	8
Nº de ciclos	125	60	18	0	0	7	0	0

Tabela 8: Números de ciclos para cada *flag* aplicado aos extremos relativos

Olhando para uma representação gráfica, é de notar que agora há mais ciclos a terminar no mesmo instante e o início e fim de cada ciclo naturalmente ocorre mais tardiamente ou no mesmo instante que nos dados de aceleração usuais:

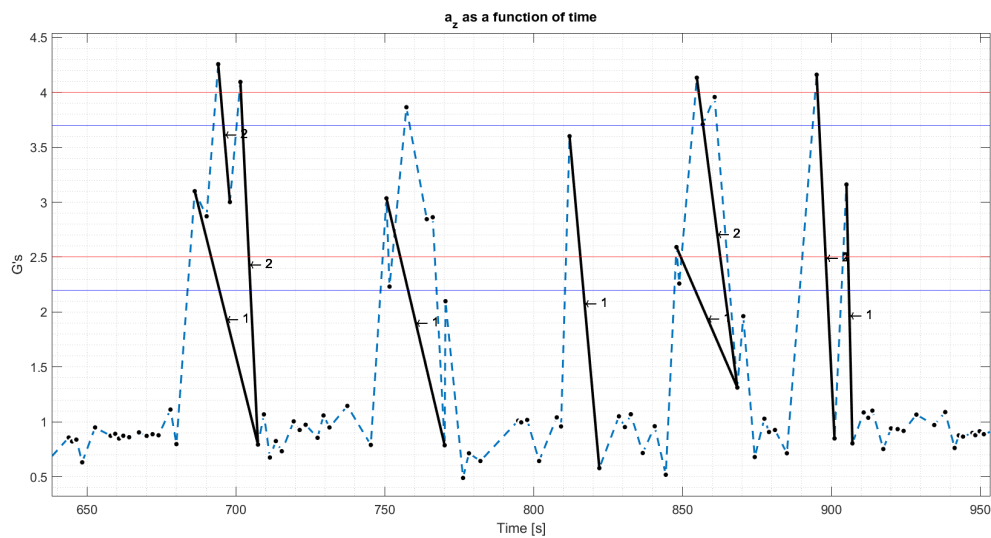


Figura 8: Representação gráfica da aplicação do algoritmo para as *flags* da tabela 6 numa porção dos dados dos extremos relativos

4 EV_2024.02C: Perguntas 12 - 16

O ficheiro EV_2024.02C contém os dados recolhidos durante um voo de calibração de um sistema convencional de navegação, com o objetivo de avaliar o desempenho do sistema de navegação *Euro-pean Geostationary Navigation Overlay Service* (EGNOS). O ficheiro inclui dados de tempo (RX_TOM e RX_WEEK), número de satélites (NSV_LOCK e NSV_USED), níveis de proteção (NS_HPL e NS_VPL), latitude (NS_LAT e REF_LAT), longitude (NS_LON e REF_LON) e altitude (NS_ALT e REF_ALT), tanto para a solução de navegação EGNOS como para a posição de referência ("verdadeira") da aeronave.

A avaliação do EGNOS será realizada de acordo com os requisitos do *International Civil Aviation Organization* (ICAO), especificados no documento *Standards and Recommended Practices* (SARPs), para três modos de operação: CAT-I¹, APV-II² e APV-I³, para um sistema *Satellite Based Augmentation System* (SBAS):. Deste modos, os parâmetros de desempenho a serem avaliados são:

- **Exatidão:** Erros de posição horizontal (HPE) e vertical (VPE);
- **Integridade:** Níveis de proteção horizontal (HPL) e vertical (VPL), e sua relação com os erros de navegação;
- **Disponibilidade:** Percentagem de tempo em que o sistema permite um determinado modo de operação;
- **Continuidade:** Contagem de períodos ininterruptos de disponibilidade do sistema.

A análise destes dados visa determinar os erros do sistema EGNOS; representar graficamente os erros, os limites de proteção e o número de satélites utilizados; calcular os parâmetros de desempenho; identificar eventos de integridade; e, finalmente, comentar os resultados obtidos. Esta avaliação, contribui para uma compreensão abrangente do desempenho do EGNOS em cenários reais e a conformidade com os requisitos do ICAO.

4.1 Erros do sistema de navegação (HPE e VPE)

O objetivo desta secção é calcular os erros do sistema de navegação EGNOS quer para as posições horizontais - *Horizontal Position Error* (HPE) - quer para as verticais - *Vertical Position Error* (VPE) - para posterior análise e comparação com outras grandezas, de modo a retirar conclusões sobre o sistema de navegação.

O erros de posição são definidos como as distâncias absolutas (em metros) entre a posição de referência ("verdadeira") - obtidos pelo programa *TOTAL TRIMBLE CONTROL* - e a posição medida pelo sistema EGNOS, no plano horizontal e vertical, HPE e VPE, respetivamente. A posição geográfica é medida no sistema padrão de coordenadas WGS84, que considera a Terra como um elipsoide com componentes de latitude, longitude e altitude.

De modo a obter o HPE, em metros, foram utilizadas duas abordagens:

- **Função *Haversine*:** Esta abordagem consiste na implementação da fórmula de Haversine, utilizada para calcular a distância entre dois pontos numa esfera, dadas as suas latitudes e longitudes. É relativamente simples de implementar e computacionalmente eficiente. No entanto, assume que a Terra é uma esfera perfeita, o que não é totalmente preciso. Isto pode levar a pequenas imprecisões no cálculo da distância. [3]

Uso da função *Haversine*:

```
distance_m = haversine(lat, lon, lat_ref, lon_ref) ;
```

- **Função *geodetic2enu*:** Outra abordagem envolve a conversão das coordenadas WGS84 (latitude, longitude, altitude) em coordenadas locais *East-North-Up* (ENU). Usando a função *geodetic2enu* do *MATLAB*, que utiliza o elipsoide WGS84 para realizar essa conversão, podemos

¹CAT-I: precision approach CATegory I

²APV-II: Approach Procedure with Vertical guidance II

³APV-I: Approach Procedure with Vertical guidance I

determinar as componentes: Este (x_East), Norte (y_North) e Cima (z_Up). O HNE é então calculado como a raiz quadrada da soma dos quadrados das componentes x_East e y_North. Essa abordagem permite uma consideração mais precisa do erro horizontal. [4]

Uso da função *geodetic2en*:

```
[x_East,y_North,zUp] = geodetic2enu(lat, lon, ...
    alt, lat_ref, lon_ref, alt_ref, ...
    wgs84Ellipsoid);
HPE = sqrt(x_East.^2+y_North.^2);
distance_m = haversine(lat, lon, lat_ref, lon_ref) ;
```

Comparando as diferenças das abordagens obtidas, através do vetor *dif_h* do código no anexo A.3, apesar das diferenças pouco significativas (ordem dos mm), optou-se por utilizar a abordagem da função *geodetic2enu*, garantindo maior rigor, uma vez que eficiência computacional não é prioritária.

De modo a obter o VPE, em metros, a abordagem é mais direta. Na realidade, uma vez obtido o valor para a *zUp* apartir da função *geodetic2enu*, basta fazer o módulo do mesmo para obter a diferença entre a altitude de referência e a altitude medida pelo EGNOS.

4.2 Análise temporal dos erros dos sistema, dos limites de proteção e do número de satélites do sistema de navegação

Uma vez obtidos os valores HPE e VPE é possível, representar graficamente os erros e os limites de proteção horizontais (fig. 9) e verticais (fig. 10) em função do tempo (RX_TOM).

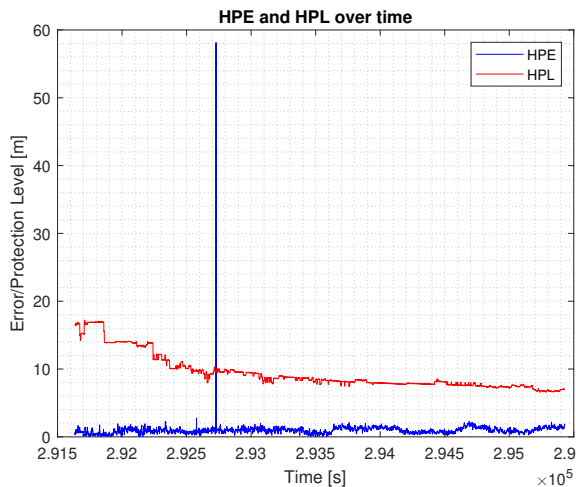


Figura 9: HPE e HPL em função do tempo

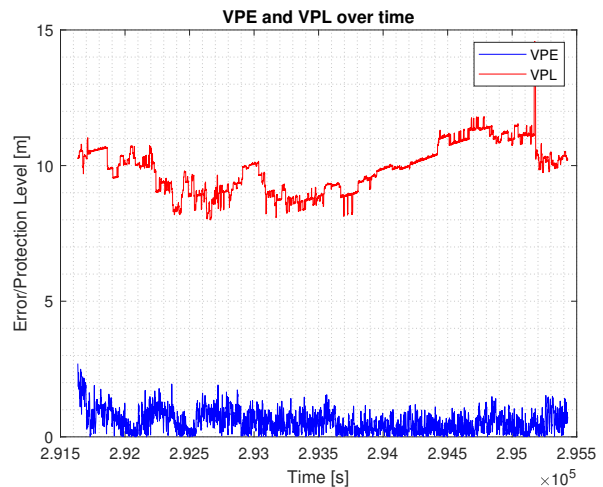


Figura 10: VPE e VPL em função do tempo

Através da observação do gráfico HPE/HPL em função do tempo é facilmente identificável um ponto que se destaca dos outros, por volta do instante $t \approx 2.927 \times 10^5$ s, este será analisado na secção Eventos de Integridade e Conclusões.

Além disso, também foi representado o número de satélites utilizados (NSV_USED) em função do tempo:

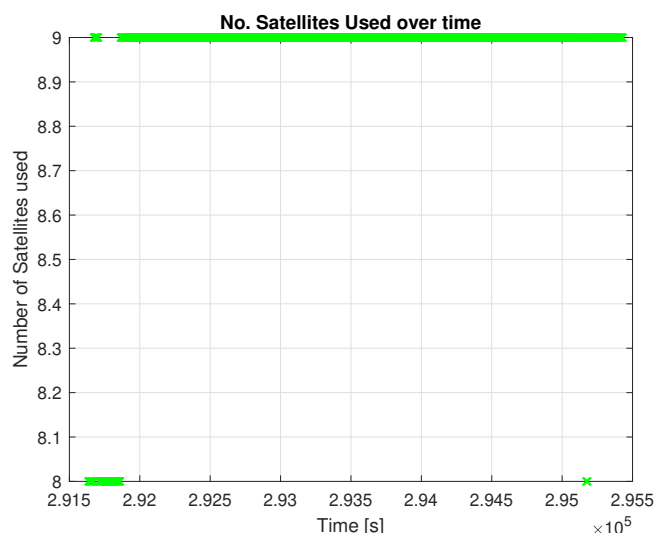


Figura 11: Número de satélites utilizados em função do tempo

A partir dos gráficos apresentados é ainda que possível concluir que: não houve uma grande flutuação do satélites utilizados; e também não se verificou uma acentuada relação entre o número de satélites usados e os erros de posição.

Os gráficos apresentados foram obtidos a partir do código *MATLAB* do anexo A.3.

4.3 Parâmetros de Desempenho

Os parâmetros de desempenho do sistema de navegação EGNOS são avaliados com base nos critérios estabelecidos pela ICAO. Estes parâmetros incluem exatidão, integridade, disponibilidade e continuidade.

De notar, que todos os dados apresentados foram obtidos do código *MATLAB* no anexo A.3.

4.3.1 Exatidão

A exatidão é quantificada pelo HPE e pelo VPE. Na verdade, o percentil 95 destes erros é comparado com os limites superiores definidos para diferentes modos de operação (ver tabela 9).

Modo de Operação	HPE (95%)	VPE (95%)
APV-I	16 m	20 m
APV-II	16 m	8 m
CAT-I	16 m	5 m

Tabela 9: Limites a ser verificados por cada modo de operação para o parâmetro exatidão

Assim, para os dados estudados foram obtidos os seguintes valores:

- HPE (95%) = 1.6292 m ;
- VPE (95%) = 1.2606 m ;

Uma vez que os valores obtidos (HPE (95%) e VPE(95%)) são inferiores aos limites apresentados, é possível então concluir que, pelo parâmetro exatidão, o sistema pode operar em qualquer modo de operação.

4.3.2 Integridade

A integridade é medida através do Nível de Proteção Horizontal (HPL) e Nível de Proteção Vertical (VPL), comparados com o Nível de Alerta Horizontal (HAL) e Nível de Alerta Vertical (VAL). Assim,

os Percentis 99 dos níveis de proteção das amostras obtidas devem verificar os limites superiores da tabela 10.

De realçar, que existe a possibilidade de ocorrência de eventos de integridade, melhor explicitados na secção Eventos de Integridade e Conclusões.

Modo de Operação	HAL	VAL
APV-I	40 m	50 m
APV-II	40 m	20 m
CAT-I	40 m	12 m

Tabela 10: Limites a ser verificados para cada modo de operação para o parâmetro integridade

Deste modo, para os Percentis 99, os seguintes valores foram obtidos:

- HPL (99%) = 16.920 m ;
- VPL (99%) = 11.701 m ;

Sendo os valores dos Percentis 99 dos níveis de proteção (HPL (99%) e VPL(99%)) inferiores aos níveis de alerta apresentados na tabela, é possível afirmar que, pelo parâmetro integridade, o sistema pode operar em qualquer modo de operação.

4.3.3 Disponibilidade

A disponibilidade corresponde à percentagem de tempo durante o período de análise em que o sistema de navegação suporta um modo específico de operação, baseado nos níveis de proteção estarem dentro dos limites de alerta. De acordo com o ICAO, o sistema deve manter mais de 99% de disponibilidade para que um modo seja considerado operacional.

Assim, através do código no anexo A.3 foram obtidos os seguintes valores para a disponibilidade de cada modo:

- APV-I: 100%;
- APV-II: 100%;
- CAT-I: 99.7104%;

Assim, uma vez que a disponibilidade foi superior a 99% para todos os modos, é possível afirmar que todos modos poderão vir a ser adotados como modo de operação do sistema de navegação em satélite EGNOS em questão, pelo parâmetro disponibilidade.

No gráfico 12, é possível visualizar a permissividade de cada modo(os níveis de proteção estarem dentro dos limites de alerta) ao longo do tempo, de modo a ter uma representação visual da disponibilidade.

4.3.4 Continuidade

A continuidade avalia o desempenho ininterrupto do sistema, contabilizando os intervalos de manutenção da disponibilidade para cada um dos modo. O número de eventos de continuidade é contabilizado para garantir uma operação consistente do sistema. ⁴

⁴De notar, que no gráfico 12, *Availability* é usado como sinónimo de permissividade de cada modo (em cada instante), e não, como sinónimo de disponibilidade

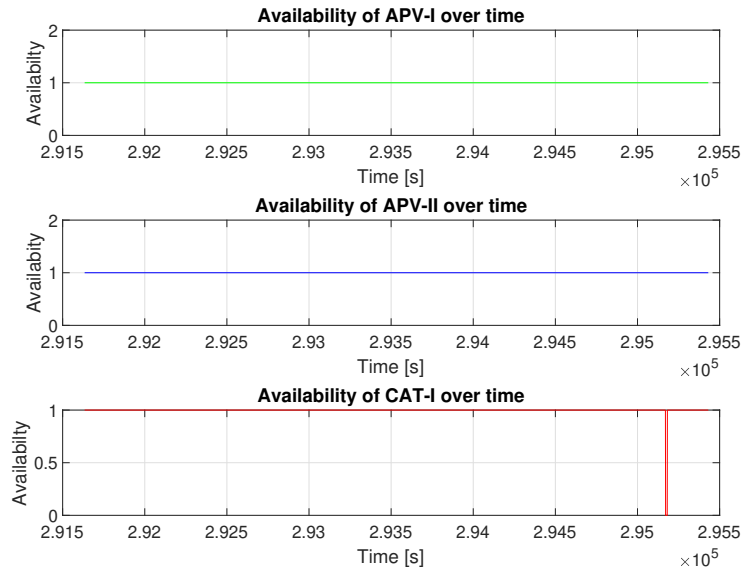


Figura 12: Visualização da Disponibilidade e da Continuidade

O gráfico acima apresenta uma representação visual dos resultados obtidos relativos à continuidade, de seguida apresentados.

No código *MATLAB*, os eventos/intervalos de continuidade foram detetados e contabilizados pelas das transições [modo indisponível] → [modo disponível] (ou seja $0 \rightarrow 1$), através da seguinte linha de código:

```
continuity_CAT_I = sum(diff([0; aux_CAT_I ; 0]) == 1);
```

Os resultados obtidos foram:

- APV-I: 1 evento de continuidade;
- APV-II: 1 evento de continuidade;
- CAT-1: 2 eventos de continuidade;

Estes resultados são, portanto, coerentes com os obtidos para a disponibilidade, uma vez que, por exemplo, para uma disponibilidade 100% será de esperar apenas um evento de continuidade, enquanto que para uma disponibilidade $< 100\%$, será de esperar no mínimo 2 (exceto se a "falha" na disponibilidade ocorrer nos extremos).

4.4 Eventos de Integridade e Conclusões

A identificação de eventos de integridade é uma etapa crítica na avaliação do desempenho do sistema de navegação, como o EGNOS. Eventos de integridade ocorrem se, em algum momento, o nível de proteção for inferior ao erro correspondente, requerendo a identificação e análise destes eventos.

Analisando os dados fornecidos, com o código presente no anexo A.3, foi possível detetar os seguintes eventos de integridade (e as suas características relevantes):

Índice	Tempo (s)	HPE (m)	HPL (m)
1095	292728	58.144	9.594

Tabela 11: Eventos de Integridade Horizontal

No período analisado, não foram encontrados eventos de integridade vertical.

O evento de integridade detetado é, também, facilmente identificável através do gráfico 9, correspondendo este ao único ponto que verifica $HPE > HPL$. Este resultado sugere que, apesar de um incidente isolado, o sistema manteve a integridade horizontal e vertical dentro dos limites aceitáveis durante a maioria ou todo o tempo de operação.

Em suma, as medições dos erros horizontais e verticais, em conjunto com a verificação dos níveis de proteção, permitiram mostrar que o sistema cumpre os requisitos estabelecidos pela ICAO para os modos de operação APV-I, APV-II e CAT-I. Além disso, os eventos de integridade identificados foram mínimos, tendo, o sistema demonstrado alta disponibilidade e números de eventos de continuidade próximos do 1 e diferentes de 0. Estes resultados confirmam que o sistema é confiável, garantindo a segurança e precisão necessárias.

5 Conclusão

A familiarização com o processamento e análise de dados, gerados por sistemas de instrumentação utilizados em ensaios em voo é fundamental para que a interpretação dos resultados seja feita corretamente. A realização deste trabalho experimental permitiu, assim, compreender não só a sua importância, mas também perceber como são feitos.

Primeiramente, foram analisados dados obtidos em laboratório e utilizando um sistema de instrumentação PCM de Classe I, que permitiu não só visualizar a variação temporal das acelerações medidas como também determinar os seus espectros unilaterais de amplitude. Foram, ainda, analisados os seus picos de amplitude.

Em seguida, foi analisada a variação temporal das grandezas obtidas em voo numa aeronave DORNIER-DASSAULT ALPHA-JET, tendo também sido obtido os extremos relativos das acelerações verticais. Por fim, desenvolveu-se um algoritmo capaz de contar a ocorrência de ciclos de aceleração vertical.

Na última parte do trabalho, foram analisados os dados recolhidos num voo de calibração de um sistema convencional de navegação. Foi possível confirmar o bom desempenho do sistema de navegação EGNOS, que cumpriu os requisitos da ICAO, para os 3 modos de operação.

Desta forma, é possível concluir que os resultados obtidos ao longo de todo o trabalho foram ao encontro do que era previsto, permitindo-nos cumprir o objetivo de nos familiarizarmos com o processamento e análise de dados, gerados por sistemas de instrumentação utilizados em ensaios em voo. Ao mesmo tempo, que nos permitiu desenvolver *soft-skills*, como a cooperação e o trabalho de equipa.

Referências

- [1] Telemetry Group. (2023). *IRIG Standard 106-23: Telemetry Standards*. Secretariat Range Commanders Council. Retrieved from https://www.irig106.org/docs/106-23/106-23_Cover_TOC_Changes_Preface.pdf.
- [2] MathWorks. (n.d.). *FFT*. MATLAB & Simulink. Retrieved June 11, 2024, from <https://www.mathworks.com/help/matlab/ref/fft.html>.
- [3] Gade, Kenneth (2010). *A Non-singular Horizontal Position Representation*. Journal of Navigation. 63 (3): 395–417. Bibcode:2010JNav...63..395G. doi:10.1017/S0373463309990415. ISSN 0373-4633.
- [4] MathWorks. (n.d.). *wgs84Ellipsoid*. MATLAB & Simulink. Retrieved June 9, 2024, from <https://www.mathworks.com/help/map/ref/wgs84ellipsoid.html>.

A Anexos - Código

A.1 EV_2024.02A

Análise temporal das grandezas medidas

```
clear; close all; clc;

% Read data
data_1 = readtable("EV_2024_A.txt");

mean_a1 = mean(data_1.a1);
mean_a2 = mean(data_1.a2);
mean_a3 = mean(data_1.a3);
mean_a4 = mean(data_1.a4);

% Signal parameters
delta_time = zeros(length(data_1.t)- 1, 1);
for step=1:length(data_1.t)-1
    delta_time(step) = data_1.t(step+1) - data_1.t(step);
end

frequency = 1./delta_time;
frequency = [frequency; frequency(1)];

Fs = frequency(1);    % Sampling frequency
T = 1/Fs;             % Sampling period
L = length(data_1.t); % Length of signal
t = (0:L-1)*T;        % Time vector

frequencies = Fs/L*(0:L-1);

f = Fs/L*(0:(L/2));

% Plot data
figure;
subplot(2,2,1);
plot(data_1.t, data_1.a1);
grid minor;
title('Acceleration a_1 as a function of time');
xlabel('Time [s]');
ylabel('a_1 [m s-2]');
box on;
subplot(2,2,2);
plot(data_1.t, data_1.a2);
grid minor;
title('Acceleration a_2 as a function of time');
xlabel('Time [s]');
ylabel('a_2 [m s-2]');
box on;
subplot(2,2,3);
plot(data_1.t, data_1.a3);
grid minor;
title('Acceleration a_3 as a function of time');
xlabel('Time [s]');
ylabel('a_3 [m s-2]');
box on;
subplot(2,2,4);
plot(data_1.t, data_1.a4);
grid minor;
title('Acceleration a_4 as a function of time');
xlabel('Time [s]');
ylabel('a_4 [m s-2]');
box on;
```

Espetro unilateral de amplitude

```
% Single-sided magnitude spectrum
Y_1 = fft(data_1.a1);
Y_2 = fft(data_1.a2);
Y_3 = fft(data_1.a3);
Y_4 = fft(data_1.a4);

P2_1 = abs(Y_1/L);
P1_1 = P2_1(1:L/2+1);
P1_1(2:end-1) = 2*P1_1(2:end-1);

P2_2 = abs(Y_2/L);
P1_2 = P2_2(1:L/2+1);
P1_2(2:end-1) = 2*P1_2(2:end-1);

P2_3 = abs(Y_3/L);
P1_3 = P2_3(1:L/2+1);
P1_3(2:end-1) = 2*P1_3(2:end-1);

P2_4 = abs(Y_4/L);
P1_4 = P2_4(1:L/2+1);
P1_4(2:end-1) = 2*P1_4(2:end-1);

figure;
subplot(2,2,1);
plot(f, P1_1);
grid minor;
title('Single-sided magnitude spectrum of a_1');
xlabel('Frequency [s^{-1}]');
ylabel('|Y_1|')
box on;
subplot(2,2,2);
plot(f, P1_2);
grid minor;
title('Single-sided magnitude spectrum of a_2');
xlabel('Frequency [s^{-1}]');
ylabel('|Y_2|')
box on;
subplot(2,2,3);
plot(f, P1_3);
grid minor;
title('Single-sided magnitude spectrum of a_3');
xlabel('Frequency [s^{-1}]');
ylabel('|Y_3|')
box on;
subplot(2,2,4);
plot(f, P1_4);
grid minor;
title('Single-sided magnitude spectrum of a_4');
xlabel('Frequency [s^{-1}]');
ylabel('|Y_4|')
box on;
```

A.2 EV_2024.02B

Ler tabela de valores

```
% Read data
data_2 = readtable("EV_2024_B.txt");
```

Análise temporal dos dados de voo

```
figure;
subplot(3,3,3);
plot(data_2.t, data_2.N2_rh, 'LineWidth', 1.5);
grid minor;
title('N2_{rh} as a function of time');
xlabel('Time [s]');
ylabel('N2_{rh} [%]')
box on;
axis auto;
subplot(3,3,6);
plot(data_2.t, data_2.EGT_rh, 'LineWidth', 1.5);
grid minor;
title('EGT_{rh} as a function of time');
xlabel('Time [s]');
ylabel('EGT_{rh} [K]')
box on;
axis auto;
subplot(3,3,9);
plot(data_2.t, data_2.FF_rh, 'LineWidth', 1.5);
grid minor;
title('FF_{rh} as a function of time');
xlabel('Time [s]');
ylabel('FF_{rh} [kg h^{-1}]')
box on;
axis auto;
subplot(3,3,2);
plot(data_2.t, data_2.EAS, 'LineWidth', 1.5);
grid minor;
title('EAS as a function of time');
xlabel('Time [s]');
ylabel('EAS [kn]')
box on;
axis auto;
subplot(3,3,5);
plot(data_2.t, data_2.QNE, 'LineWidth', 1.5);
grid minor;
title('QNE as a function of time');
xlabel('Time [s]');
ylabel('QNE [ft]')
box on;
axis auto;
subplot(3,3,8);
plot(data_2.t, data_2.a_z, 'LineWidth', 1.5);
grid minor;
title('a_{z} as a function of time');
xlabel('Time [s]');
ylabel('a_{z} [m s^{-2}]')
box on;
axis auto;
subplot(3,3,1);
plot(data_2.t, data_2.N2_lt, 'LineWidth', 1.5);
grid minor;
title('N2_{lt} as a function of time');
xlabel('Time [s]');
ylabel('N2_{lt} [%]')
box on;
axis auto;
subplot(3,3,4);
plot(data_2.t, data_2.EGT_lt, 'LineWidth', 1.5);
grid minor;
```

```

title('EGT_{lt} as a function of time');
xlabel('Time [s]');
ylabel('EGT_{lt} [K]')
box on;
axis auto;
subplot(3,3,7);
plot(data_2.t, data_2.FF_lt, 'LineWidth', 1.5);
grid minor;
title('FF_{lt} as a function of time');
xlabel('Time [s]');
ylabel('FF_{lt} [kg h^{-1}]')
box on;
axis auto;

```

Análise temporal da aceleração vertical

```

acc = data_2.a_z/9.80665;

figure;
plot(data_2.t, acc, '-o', 'LineWidth', 1.5, 'MarkerSize', 1.8, 'MarkerEdgeColor', [1 0.5 0],
↵ 'MarkerFaceColor', [0 0 0]);
yline(1, '--', 'LineWidth', 2);
%yline(2, 'LineWidth', 1.2);
%yline(1.6, 'LineWidth', 1.2);
grid minor;
title('a_{z} as a function of time');
xlabel('Time [s]');
ylabel("G's")

```

Extremos relativos das acelerações verticais

```

[acc_max_peaks, acc_max_locs] = findpeaks(acc);

[acc_min_peaks, acc_min_locs] = findpeaks(-acc);

acc_min_peaks = -acc_min_peaks;
acc_max_locs = data_2.t(acc_max_locs);
acc_min_locs = data_2.t(acc_min_locs);

figure; hold on;
plot(acc_max_locs, acc_max_peaks, "--o", 'LineWidth', 1.5, 'MarkerSize', 1.8, 'MarkerEdgeColor', [1 0 0],
↵ 'MarkerFaceColor', [0 0 0]);
plot(acc_min_locs, acc_min_peaks, "--o", 'LineWidth', 1.5, 'MarkerSize', 1.8, 'MarkerEdgeColor', [0 0 1],
↵ 'MarkerFaceColor', [0 0 0]);
xlabel('Time [s]');
ylabel("Acceleration [G's]");
legend('Relative maximums', 'Relative minimums', 'Location', 'southeast');
grid minor;
box on;

max = cat(2, acc_max_locs, acc_max_peaks);
min = cat(2, acc_min_locs, acc_min_peaks);
%extremes = table('VariableNames', {'Time [s]', 'Aceleration [m/s^2]}');
extremes = sortrows(cat(1, max, min));

save("relative_extremes", "extremes");
writematrix(extremes, 'relative_extremes.csv');

```

Algoritmo para detecção de ciclos de aceleração vertical

```

N_1 = [2.5 4.0 5.0 6.0 7.0 0.0 -1.5 -2.5];
N_2 = [2.2 3.7 4.7 5.7 6.7 0.3 -1.2 -2.2];

```

```

flag_status_1 = acc_cicle(N_1(1), N_2(1), acc, data_2.t);
flag_status_2 = acc_cicle(N_1(2), N_2(2), acc, data_2.t);
flag_status_3 = acc_cicle(N_1(3), N_2(3), acc, data_2.t);
flag_status_4 = acc_cicle(N_1(4), N_2(4), acc, data_2.t);
flag_status_5 = acc_cicle(N_1(5), N_2(5), acc, data_2.t);
flag_status_6 = acc_cicle(N_1(6), N_2(6), acc, data_2.t);
flag_status_7 = acc_cicle(N_1(7), N_2(7), acc, data_2.t);
flag_status_8 = acc_cicle(N_1(8), N_2(8), acc, data_2.t);

%drawArrow = @(x,y) quiver(x(1),y(1),x(2)-x(1),y(2)-y(1),0);

figure;hold on;
plot(data_2.t, acc, "--o", 'LineWidth', 1.5, 'MarkerSize', 5, 'MarkerEdgeColor', [1 1 1],
↳ 'MarkerFaceColor', [0 0 0]);
for flag=1:length(N_1)
    for iter=1:size(flag_status_1,1)
        plot([flag_status_1(iter, 1) flag_status_1(iter, 3)], [flag_status_1(iter, 2) flag_status_1(iter,
↳ 4)], 'LineWidth', 2, 'Color', [0 0 0]);
        text((flag_status_1(iter, 1)+flag_status_1(iter, 3))/2, (flag_status_1(iter,
↳ 2)+flag_status_1(iter, 4))/2, '\leftarrow 1');
    end
end
for flag=1:length(N_1)
    for iter=1:size(flag_status_2,1)
        plot([flag_status_2(iter, 1) flag_status_2(iter, 3)], [flag_status_2(iter, 2) flag_status_2(iter,
↳ 4)], 'LineWidth', 2, 'Color', [0 0 0]);
        text((flag_status_2(iter, 1)+flag_status_2(iter, 3))/2, (flag_status_2(iter,
↳ 2)+flag_status_2(iter, 4))/2, '\leftarrow 2');
    end
end
for flag=1:length(N_1)
    for iter=1:size(flag_status_3,1)
        plot([flag_status_3(iter, 1) flag_status_3(iter, 3)], [flag_status_3(iter, 2) flag_status_3(iter,
↳ 4)], 'LineWidth', 2, 'Color', [0 0 0]);
        text((flag_status_3(iter, 1)+flag_status_3(iter, 3))/2, (flag_status_3(iter,
↳ 2)+flag_status_3(iter, 4))/2, '\leftarrow 3');
    end
end
for flag=1:length(N_1)
    for iter=1:size(flag_status_6,1)
        plot([flag_status_6(iter, 1) flag_status_6(iter, 3)], [flag_status_6(iter, 2) flag_status_6(iter,
↳ 4)], 'LineWidth', 2, 'Color', [0 0 0]);
        text((flag_status_6(iter, 1)+flag_status_6(iter, 3))/2, (flag_status_6(iter,
↳ 2)+flag_status_6(iter, 4))/2, '\leftarrow 6');
    end
end
grid minor;
for i=1:length(N_1)
    yline(N_1(i), 'red');
end
for i=1:length(N_2)
    yline(N_2(i), 'blue');
end
box on;
title('a_{z} as a function of time');
xlabel('Time [s]');
ylabel("Acceleration G's");

flag_status_1_ext = acc_cicle(N_1(1), N_2(1), extremes(:,2), extremes(:,1));
flag_status_2_ext = acc_cicle(N_1(2), N_2(2), extremes(:,2), extremes(:,1));
flag_status_3_ext = acc_cicle(N_1(3), N_2(3), extremes(:,2), extremes(:,1));
flag_status_4_ext = acc_cicle(N_1(4), N_2(4), extremes(:,2), extremes(:,1));
flag_status_5_ext = acc_cicle(N_1(5), N_2(5), extremes(:,2), extremes(:,1));
flag_status_6_ext = acc_cicle(N_1(6), N_2(6), extremes(:,2), extremes(:,1));
flag_status_7_ext = acc_cicle(N_1(7), N_2(7), extremes(:,2), extremes(:,1));
flag_status_8_ext = acc_cicle(N_1(8), N_2(8), extremes(:,2), extremes(:,1));

figure;hold on;
plot(extremes(:,1), extremes(:,2), "--o", 'LineWidth', 1.5, 'MarkerSize', 5, 'MarkerEdgeColor', [1 1 1],
↳ 'MarkerFaceColor', [0 0 0]);
for flag=1:length(N_1)
    for iter=1:size(flag_status_1_ext,1)

```

```

    plot([flag_status_1_ext(iter, 1) flag_status_1_ext(iter, 3)], [flag_status_1_ext(iter, 2)
    ↪ flag_status_1_ext(iter, 4)], 'LineWidth', 2, 'Color', [0 0 0]);
    text((flag_status_1_ext(iter, 1)+flag_status_1_ext(iter, 3))/2, (flag_status_1_ext(iter,
    ↪ 2)+flag_status_1_ext(iter, 4))/2, '\leftarrow 1');
end
end
for flag=1:length(N_1)
    for iter=1:size(flag_status_2_ext,1)
        plot([flag_status_2_ext(iter, 1) flag_status_2_ext(iter, 3)], [flag_status_2_ext(iter, 2)
        ↪ flag_status_2_ext(iter, 4)], 'LineWidth', 2, 'Color', [0 0 0]);
        text((flag_status_2_ext(iter, 1)+flag_status_2_ext(iter, 3))/2, (flag_status_2_ext(iter,
        ↪ 2)+flag_status_2_ext(iter, 4))/2, '\leftarrow 2');
    end
end
for flag=1:length(N_1)
    for iter=1:size(flag_status_3_ext,1)
        plot([flag_status_3_ext(iter, 1) flag_status_3_ext(iter, 3)], [flag_status_3_ext(iter, 2)
        ↪ flag_status_3_ext(iter, 4)], 'LineWidth', 2, 'Color', [0 0 0]);
        text((flag_status_3_ext(iter, 1)+flag_status_3_ext(iter, 3))/2, (flag_status_3_ext(iter,
        ↪ 2)+flag_status_3_ext(iter, 4))/2, '\leftarrow 3');
    end
end
for flag=1:length(N_1)
    for iter=1:size(flag_status_6_ext,1)
        plot([flag_status_6_ext(iter, 1) flag_status_6_ext(iter, 3)], [flag_status_6_ext(iter, 2)
        ↪ flag_status_6_ext(iter, 4)], 'LineWidth', 2, 'Color', [0 0 0]);
        text((flag_status_6_ext(iter, 1)+flag_status_6_ext(iter, 3))/2, (flag_status_6_ext(iter,
        ↪ 2)+flag_status_6_ext(iter, 4))/2, '\leftarrow 6');
    end
end
grid minor;
for i=1:length(N_1)
    yline(N_1(i), 'red');
end
for i=1:length(N_2)
    yline(N_2(i), 'blue');
end
box on;
title('a_{z} as a function of time');
xlabel('Time [s]');
ylabel("G's");

```

A.3 EV_2024.02C

Ler dados e Erros do sistema de navegação (HPE e VPE)

```
clear; close all; clc;

% Read data
data_3 = readtable("EV_2024_C.txt");

% Haversine function - Distance between two coordinates in the system WGS84
% distance -> m, coordinates -> degrees
function distance = haversine(lat1, lon1, lat2, lon2)
    R = 6371000;
    lat1 = deg2rad(lat1);
    lon1 = deg2rad(lon1);
    lat2 = deg2rad(lat2);
    lon2 = deg2rad(lon2);
    distance = 2 * R .* asin(sqrt((1-cos(lat2-lat1)+ ...
        cos(lat1).*cos(lat2).*(1-cos(lon2-lon1)))/2));
end

%12.
% Calculate HPE
HPE_haversine = haversine(data_3.NS_LAT, data_3.NS_LON, ...
    data_3.REF_LAT, data_3.REF_LON);
[x_East,y_North,zUp] = geodetic2enu(data_3.NS_LAT, data_3.NS_LON, ...
    data_3.NS_ALT, data_3.REF_LAT, data_3.REF_LON,data_3.REF_ALT, ...
    wgs84Ellipsoid);
HPE = sqrt(x_East.^2+y_North.^2);
dif_h = HPE_haversine-HPE; %compare the difference between the two methods

% Calculate VPE
VPE = abs(zUp);
```

Análise temporal dos erros dos sistema, dos limites de proteção e do número de satélites do sistema de navegação

```
%13.

% Variables for the plots
time = data_3.RX_TOM;
HPL = data_3.NS_HPL;
VPL = data_3.NS_VPL;
Sat_USED = data_3.NSV_USED;

% Plot of HPE & HPL over time
figure;
hold on;
plot(time, HPE, 'b');
plot(time, HPL, 'r');
title('HPE and HPL over time');
xlabel('Time [s]');
ylabel('Error/Protection Level [m]');
legend('HPE', 'HPL');
grid minor;
box on;
hold off;

% Plot of VPE & VPL over time
figure;
hold on;
plot(time, VPE, 'b');
plot(time, VPL, 'r');
title('VPE and VPL over time');
xlabel('Time [s]');
ylabel('Error/Protection Level [m]');
legend('VPE', 'VPL');
```

```

grid minor;
box on;
hold off;

% Plot of No. satellites used over time
figure;
plot(time, Sat_USED, 'gx');
title('No. Satellites Used over time');
xlabel('Time [s]');
ylabel('Number of Satellites used');
grid on;

```

Parâmetros de Desempenho

```

%14.
% Performance limits for accuracy and integrity [HPE(95%), VPE(95%), HAL, VAL]
limits_APV_I = [16, 20, 40, 50];
limits_APV_II = [16, 8, 40, 20];
limits_CAT_I = [16, 5, 40, 12];

% Compute Performance Parameters
% Accuracy
HPE_95 = prctile(HPE, 95);
VPE_95 = prctile(VPE, 95);
accuracy_APV_I = [HPE_95 <= limits_APV_I(1), VPE_95 <= limits_APV_I(2)];
accuracy_APV_II = [HPE_95 <= limits_APV_II(1), VPE_95 <= limits_APV_II(2)];
accuracy_CAT_I = [HPE_95 <= limits_CAT_I(1), VPE_95 <= limits_CAT_I(2)];
accuracy = [accuracy_APV_I, accuracy_APV_II, accuracy_CAT_I];

% Integrity
HPL_99 = prctile(HPL, 99);
VPL_99 = prctile(VPL, 99);
integrity_APV_I = [HPL_99 <= limits_APV_I(3), VPL_99 <= limits_APV_I(4)];
integrity_APV_II = [HPL_99 <= limits_APV_II(3), VPL_99 <= limits_APV_II(4)];
integrity_CAT_I = [HPL_99 <= limits_CAT_I(3), VPL_99 <= limits_CAT_I(4)];
integrity = [integrity_APV_I, integrity_APV_II, integrity_CAT_I];

% Aux variable for availability and continuity - verify if the operation mode is
% allowed
aux_APV_I = (HPL <= limits_APV_I(3)) & (VPL <= limits_APV_I(4));
aux_APV_II = (HPL <= limits_APV_II(3)) & (VPL <= limits_APV_II(4));
aux_CAT_I = (HPL <= limits_CAT_I(3)) & (VPL <= limits_CAT_I(4));

% Availability
availability_APV_I = sum(aux_APV_I) / numel(time) * 100;
availability_APV_II = sum(aux_APV_II) / numel(time) * 100;
availability_CAT_I = sum(aux_CAT_I) / numel(time) * 100;
availability = [availability_APV_I, availability_APV_II, availability_CAT_I];

% Continuity
continuity_APV_I = sum(diff([0; aux_APV_I ; 0]) == 1); %-> counts the transitions
continuity_APV_II = sum(diff([0; aux_APV_II ; 0]) == 1);
continuity_CAT_I = sum(diff([0; aux_CAT_I ; 0]) == 1);
continuity = [continuity_APV_I, continuity_APV_II, continuity_CAT_I];

%Continuity Intervals Visualization
% APV_I
figure;
subplot(3, 1, 1);
plot(time, aux_APV_I, 'g');
title('Availability of APV-I over time');
xlabel('Time [s]');
ylabel('Availability');
grid on;
box on;

% APV_II
subplot(3, 1, 2);

```

```

plot(time, aux_APV_II, 'b');
title('Availability of APV-II over time');
xlabel('Time [s]');
ylabel('Availability');
grid on;
box on;

% CAT_I
subplot(3, 1, 3);
plot(time, aux_CAT_I, 'r');
title('Availability of CAT-I over time');
xlabel('Time [s]');
ylabel('Availability');
grid on;
box on;

% Results
disp('(1-True; 0-False)')

%Accuracy
fprintf('\n\nHPE_95 = %f m\n',HPE_95)
fprintf('VPE_95 = %f m\n',VPE_95)
disp('Accuracy (APV-I (H & V), APV-II (H & V), CAT-I (H & V))');
disp(accuracy);

%Integrity
fprintf('\nHPL_99 = %f m\n',HPL_99)
fprintf('VPL_99 = %f m\n',VPL_99)
disp('Integrity (APV-I (H & V), APV-II (H & V), CAT-I (H & V))');
disp(integrity);

%Availability and Continuity
disp('Availability (APV-I, APV-II, CAT-I): (%)');
disp(availability);

disp('Continuity (APV-I, APV-II, CAT-I): ');
disp(continuity);

```

OUTPUT:

```

(1-True; 0-False)

HPE_95 = 1.629153 m
VPE_95 = 1.260600 m
Accuracy (APV-I (H & V), APV-II (H & V), CAT-I (H & V)):
    1    1    1    1    1    1

HPL_99 = 16.920000 m
VPL_99 = 11.700840 m
Integrity (APV-I (H & V), APV-II (H & V), CAT-I (H & V)):
    1    1    1    1    1    1

Availability (APV-I, APV-II, CAT-I): (%)
    100.0000    100.0000    99.7104

Continuity (APV-I, APV-II, CAT-I):
    1        1        2

```

Eventos de Integridade

```

%15.
% Compute Integrity Events

```

```

iev_H = find(HPL < HPE);
iev_V = find(VPL < VPE);
integrity_events_H = [iev_H, time(iev_H), HPE(iev_H), HPL(iev_H)];
integrity_events_V = [iev_V, time(iev_V), HPE(iev_V), HPL(iev_V)];

% Display Integrity Events
disp('Integrity Events:');
if ~isempty(iev_H)
    fprintf('\n%d horizontal events found. \n', numel(iev_H));
    for i = 1:length(iev_H)
        n = iev_H(i);
        fprintf('Index: %d; Time = %d s; HPE = %f m; HPL = %f m \n', n, time(n), HPE(n), HPL(n));
    end
else
    fprintf('No horizontal integrity events found.\n');
end

disp('');
if ~isempty(iev_V)
    fprintf('\n %d vertical events found. \n', numel(iev_V));
    for i=1:length(iev_V)
        n = iev_V(i);
        fprintf('Index: %d; Time = %d s; HPE = %f m; HPL = %f m; \n', n, time(n), HPE(n),
            ↪ HPL(n));
    end
else
    fprintf('\nNo vertical integrity events found.\n');
end

```

OUTPUT:

Integrity Events:

1 horizontal events found.

Index: 1095; Time = 292728 s; HPE = 58.143714 m; HPL = 9.594000 m

No vertical integrity events found.