PRJ - 22 - Projeto Conceitual de Aeronave
Lab 06
Atividade Individual

Cap Eng **Ney** Rafael Secco
ney@ita.br

1T Eng João A. **Dantas** de J. Ferreira
dantas@ita.br

Instruções

1. Data de entrega: 31/10/2021, 23:59;
2. Todos devem enviar os relatórios através do Google Classroom - enviar um aquivo compactado (.zip ou .rar) com o relatório e o código escrito - o código deve ter comentários. Coloquem o nome do arquivo compactado como sendo "equipe_nome_número_do_lab.zip", com um nome curto (não coloquem o nome completo aqui). No nome da equipe, coloquem apenas o primeiro nome da equipe (caso tenha mais que um). Dentro do arquivo do relatório, coloquem o nome completo, bem como nome da equipe completo;
3. Deduções por atraso: a cada dia, 2,0 pontos são descontados da nota final;
4. Todos devem entregar um pequeno relatório com os itens marcados em vermelho. Não precisa colocar os resultados dos casos de testes;
5. Podem discutir soluções com outros alunos, mas não podem compartilhar códigos.

1 Desempenho

Esse módulo determina a tração mínima requerida para satisfazer as restrições da missão.

1.1 Inputs

Os inputs para esse módulo são:

- Parâmetro de peso: W_0.
- Parâmetros geométricos da asa: S_w, AR_w, $c_{r,w}$, λ_w, Λ_w, $(t/c)_{r,w}$, $(t/c)_{t,w}$, e b_w.
- Parâmetros geométricos da empenagem horizontal: S_h, $c_{r,h}$, λ_h, $(t/c)_{r,h}$, e $(t/c)_{t,h}$.
- Parâmetros geométricos da empenagem vertical: S_v, $c_{r,v}$, λ_v, $(t/c)_{r,v}$, e $(t/c)_{t,v}$.
- Parâmetros da fuselagem: L_f, e D_f.
- Parâmetros da nacele: L_n, e D_n.
- Parâmetros dos motores: n_{eng}, $n_{eng,w}$, e BPR.
- Parâmetros do flap: $\delta_{flap,TO}$, $\delta_{flap,LD}$, `flap_type`, c_{flap}/c_w, e b_{flap}/b_w.
- Parâmetros do slat: $\delta_{slat,TO}$, $\delta_{slat,LD}$, `slat_type`, c_{slat}/c_w, e b_{slat}/b_w.
- Parâmetro de efeito-solo: h_{ground}.
- Parâmetro do arrasto de excrescência: k_{exc}.
- Parâmetros de decolagem: h_{TO}, e d_{TO}.
- Parâmetros de pouso: h_{LD}, d_{LD}, e $MLW/MTOW$.
- Parâmetros de cruzeiro: h_{cruise}, M_{cruise}, e $M_{f,cruise}$.

1.2 Outputs

Os outputs desse módulo são:

- Parâmetros do motor: T_0, e $\vec{T_0}$.
- Parâmetro de desempenho: $S_{w,lan}$.

1.3 Chamada da função

Para calcular o desempenho da aeronave, cada aluno deve utilizar a função como a seguir:

```
def performance(aircraft,
                T0_flap_def, LD_flap_def,
                T0_slat_def, LD_slat_def,
                h_ground,
                altitude_takeoff, distance_takeoff,
                altitude_landing, distance_landing, MLW_frac,
                altitude_cruise, Mach_cruise):
.
.
.
return T0, T0vec, S_wlan
```

Onde:

$$\vec{T_0} = \left[\begin{array}{ccccccccc} T_{0,TO} & T_{0,cruise} & T_{0,FAR\,25.111} & T_{0,FAR\,25.121a} & T_{0,FAR\,25.121b} & T_{0,FAR\,25.121c} & T_{0,FAR\,25.119} & T_{0,FAR\,25.121d} \end{array} \right] \tag{1}$$

$$T_0 = 1.05 \times \max(\vec{T_0}) \tag{2}$$

e precisamos verificar se nossa asa satisfaz esse requisito: $S_w \geq S_{w,lan}$, onde $S_{w,lan}$ é a mínima área necessária encontrada na análise do pouso (ver Apêndice A)

1.4 Caso de testes

Considere um caso de testes com os inputs previamente utilizados e:

```
aircraft['weights']['W0'] = W0
aircraft['data']['misc']['Mf_cruise'] = Mf_cruise

Mach_cruise = 0.77

T0_flap_def = 0.34906585039887
LD_flap_def = 0.69813170079773
T0_slat_def = 0.0
LD_slat_def = 0.0
altitude_takeoff = 0.0
distance_takeoff = 1520.0
h_ground = 10.668
altitude_landing = 0.0
distance_landing = 1520.0
MLW_frac = 0.84
```

As variáveis `W0` e `Mf_cruise` são outputs do Lab05.

Você deve obter os seguintes resultados:

```
T0 = 134031.07204778842
T0vec = 127648.64004551277
        106115.01096274279
        101500.22368906968
        107839.76150945664
        120301.55604176046
        85371.85522844378
        61535.84926529819
        109734.68707166845
S_wlan = 74.40712149219357
```

2 Módulo de *Thrust Matching*

Nós tivemos que estimar um valor de tração de decolagem ($T_{0,guess}$) para convergir os valores de MTOW no Lab05. Entretanto, tivemos de usar o valor convergido de MTOW para determinar a tração (T_0) requerida para satisfazer os requisitos de desempenho na Sec. 1, e esse valor não necessariamente irá ser igual ao valor estimado. Portanto, precisamos de um processo iterativo para "casar" esses valores de tração. Iremos implementar esse processo iterativo nesse módulo.

2.1 Inputs

Os inputs para esse módulo são:

- Parâmetros de peso: $W_{0,guess}$, $W_{payload}$, e W_{crew}.
- Parâmetros geométricos da asa: S_w, AR_w, $c_{r,w}$, λ_w, Λ_w, $(t/c)_{r,w}$, $(t/c)_{t,w}$, b_w, $x_{m,w}$, e $c_{m,w}$.
- Parâmetros geométricos da empenagem horizontal: S_h, $c_{r,h}$, λ_h, $(t/c)_{r,h}$, $(t/c)_{t,h}$, $x_{m,h}$, e $c_{m,h}$.
- Parâmetros geométricos da empenagem vertical: S_v, $c_{r,v}$, λ_v, $(t/c)_{r,v}$, $(t/c)_{t,v}$, $x_{m,v}$, e $c_{m,v}$.
- Parâmetros da fuselagem: L_f, e D_f.
- Parâmetros da nacele: L_n, D_n, e x_n.
- Parâmetros dos flaps: $\delta_{flap,TO}$, $\delta_{flap,LD}$, `flap_type`, c_{flap}/c_w, e b_{flap}/b_w.
- Parâmetros dos slats: $\delta_{slat,TO}$, $\delta_{slat,LD}$, `slat_type`, c_{slat}/c_w, e b_{slat}/b_w.
- Parâmetro de efeito-solo: h_{ground}.
- Parâmetro de arrasto de excrescência: k_{exc}.
- Parâmetros dos motores: n_{eng}, $n_{eng,w}$, T_0, e BPR.
- Parâmetros dos trens-de-pouso: x_{nlg} e x_{mlg}.
- Parâmetros do cruzeiro: h_{cruise}, M_{cruise}, e R_{cruise}.
- Parâmetros da espera: E_{loiter}.
- Parâmetros do cruzeiro de alternativa: $h_{altcruise}$, $M_{altcruise}$, e $R_{altcruise}$.
- Parâmetros de decolagem: h_{TO}, e d_{TO}.
- Parâmetros de pouso: h_{LD}, d_{LD}, e $MLW/MTOW$.

2.2 Outputs

Os outputs desse módulo são:

- Parâmetros de peso: W_0, W_e, e W_f.
- Parâmetro de CG: $x_{CG,e}$.
- Parâmetros dos motores: T_0, e $\vec{T}_0$.
- Parâmetro de desempenho: $S_{w,lan}$.

2.3 Chamada da função

Para fazer a iteração e definir a tração, cada aluno deve utilizar a função como a seguir:

```
def thrust_matching(aircraft, W0_guess, T0_guess
                    T0_flap_def, LD_flap_def,
                    T0_slat_def, LD_slat_def,
                    h_ground,
                    altitude_cruise, Mach_cruise, range_cruise,
                    loiter_time,
                    altitude_altcruise, Mach_altcruise, range_altcruise,
                    altitude_takeoff, distance_takeoff,
                    altitude_landing, distance_landing, MLW_frac):
.
.
.
return W0, We, Wf, xcg_e, T0, T0vec, S_wlan
```

2.4 Caso de testes

Considere um caso de testes com os inputs já usados e:

```
W0_guess = 422712.90000000002328
T0_guess = 125600

altitude_cruise = 11000
Mach_cruise = 0.77
range_cruise = 2390000

loiter_time = 2700

altitude_altcruise = 4572
Mach_altcruise = 0.4
range_altcruise = 370000
```

Você deve receber os seguintes resultados:

```
W0 = 446026.6632100688
We = 241731.35205124083
Wf = 104311.79115882801
xcg_e = 17.372821482996248
T0 = 137867.81704572498
T0vec = 131302.68290069044
        107275.7230723365
        102939.8203866342
        109472.86351257091
        122008.3577771767
        86580.62736057091
        62439.589267753916
        111379.79844219559
S_wlan = 75.46458840992018
```

2.5 Exercícios

Pegue o caso de testes e varie a área de asa (S_w) enquanto mantém os outros parâmetros (independentes) fixos. Guarde o vetor de valores de tração necessários ($\vec{T_0}$) para cada valor de área de asa analisado. Gere um plot de $\vec{T_0}$ como função de S_w (área de asa) para o intervalo $80.0 < S_w < 140.0$. Note que esse plot deve ter 8 curvas, correspondentes aos 8 requisitos de desempenho descritos na Sec. 1. Se lembre de adicionar legendas para identificar essas curvas. Quais requisitos de desempenho limitam a escolha da tração dentro desse intervalo de áreas de asa?

A Construção do Código

A.1 Análise da decolagem

Inicialmente precisamos calcular as propriedades atmosféricas na condição de decolagem:

```
T,p,rho,mi = atmosphere(altitude_takeoff, 288.15)
```

Então calcule a razão σ entre a densidade do ar atual e a densidade do ar no nível do mar (que é 1.225kg/m^3):

$$\sigma = \frac{\rho}{1.225} \tag{3}$$

A seguir, use o módulo de aerodinâmica para calcular o máximo coeficiente de sustentação nas condições de decolagem ($C_{Lmax,TO}$). Use as seguintes configurações para esse caso:

```
Mach = 0.2
altitude = altitude_takeoff
n_engines_failed = 0
flap_def = TO_flap_def
slat_def = TO_flap_def
lg_down = 1
h_ground_takeoff = h_ground
Weight = W0
```

ATENÇÃO: lembre de não sobrescrever o valor de `h_ground` definido como um input para o módulo de desempenho quando atribuir o valor para o módulo de aerodinâmica.

ATENÇÃO: guarde o valor de $C_{Lmax,TO}$ que você obtém aqui para uso posterior na análise da subida (Sec. A.4).

Agira calcule a razão tração-por-peso de acordo com Roskam (Eq. 3.8 from Part I).

$$\frac{T_0}{W_0} = \frac{0.2387}{\sigma \cdot C_{Lmax,TO} \cdot d_{TO}} \cdot \frac{W_0}{S_w} \tag{4}$$

Essa equação é adaptada para unidades do SI.

Agora calcule a tração necessária para a decolagem:

$$T_{0,TO} = \left(\frac{T_0}{W_0}\right) \cdot W_0 \tag{5}$$

A.2 Análise do pouso

O procedimento de pouso não tem uma grande dependência na tração disponível. Nós só precisamos verificar se temos uma área de asa adequada para essa fase.

Comece calculando as propriedades atmosféricas na condição de pouso:

```
T,p,rho,mi = atmosphere(altitude_landing, 288.15)
```

A seguir use o módulo de aerodinâmica para calcular o coeficiente de sustentação máximo nas condições de pouso ($C_{Lmax,LD}$). Use as seguintes configurações para esse caso:

```
Mach = 0.2
altitude = altitude_landing
n_engines_failed = 0
flap_def = LD_flap_def
slat_def = LD_flap_def
lg_down = 1
h_ground_landing = h_ground
Weight = W0*MLW_frac
```

ATENÇÃO: guarde o valor de $C_{Lmax,LD}$ que você obtém aqui para uso posterior na análise da subida (Sec. A.4).

O parâmetro `MLW_frac` é a razão entre o MLW[1] e o MTOW[2]. O peso de pouso pode ser limitado por restrições de desempenho e estruturais. Nós não usamos as frações de combustível para estimar essa razão já que a aeronave não necessariamente precisará voar todo o cruzeiro antes de pousar. Como alternativa, podemos usar as tendências históricas da Fig. 1 para obter os valores para essa razão.

A distância de pouso pode ser historicamente correlacionada com a velocidade de aproximação da aeronave (V_a), de acordo com a seguinte relação (Roskam Part I Eq. 3.16):

$$V_a = 1.701 \cdot \sqrt{d_{LD}} \tag{6}$$

[1] *Maximum Landing Weight*, Máximo Peso de Pouso
[2] *Maximum Takeoff Weight*, Máximo Peso de Decolagem

Table 3.3 Typical Values For Landing Weight to Take-off Weight Ratio

Airplane Type		W_L/W_{TO} Minimum	Average	Maximum
1.	Homebuilts	0.96	1.0	1.0
2.	Single Engine Propeller Driven	0.95	0.997	1.0
3.	Twin Engine Propeller Driven	0.88	0.99	1.0
4.	Agricultural	0.7	0.94	1.0
5.	Business Jets	0.69	0.88	0.96
6.	Regional TBP	0.92	0.98	1.0
7.	Transport Jets	0.65	0.84	1.0
8.	Military Trainers	0.87	0.99	1.1
9.	Fighters (jets)	0.78	insufficient	1.0
	(tbp's)	0.57	data	1.0
10.	Mil. Patrol, Bomb and Transports (jets)	0.68	0.76	0.83
	(tbp's)	0.77	0.84	1.0
11.	Flying Boats, Amphibious and Float Airplanes (land)	0.79	insufficient	0.95
	(water)	0.98	data	1.0
12.	Supersonic Cruise Airplanes	0.63	0.75	0.88

Figura 1: Valores históricos para MLW (Roskam Part I Tab. 3.3).

Essa equação é adaptada para unidades do SI.

Os requisitos FAR 25 estabelecem que a velocidade de aproximação deve ser, pelo menos, 1.3 vezes a velocidade de stall (V_s). Portanto, a velocidade de stall pode ser obtida com (Roskam Part I Eq. 3.15):

$$V_s = \frac{V_a}{1.3} \tag{7}$$

Agora podemos calcular a área de asa necessária para alcançar essa velocidade de stall com:

$$S_{w,lan} = \frac{2 \cdot W_0 \cdot MLW/MTOW}{\rho \cdot V_s^2 \cdot C_{Lmax,LD}} \tag{8}$$

Precisamos verificar se nossa asa satisfaz esse requisito: $S_w \geq S_{w,lan}$.

A.3 Análise do cruzeiro

Precisamos verificar se temos tração suficiente para obter a velocidade de cruzeiro desejada. Para as próximas derivações, utilizaremos o peso da aeronave no começo da fase de cruzeiro. Em outras palavras: $W_cruise = W_0 \cdot M_{f,cruise}$.

Comece calculando as propriedades atmosféricas na condição de cruzeiro:

```
T,p,rho,mi = atmosphere(altitude_cruise, 288.15)
```

Assim que obtivermos a temperatura do ar (T), podemos determinar a velocidade do som (a) com:

$$a_{cruise} = \sqrt{\gamma \cdot R \cdot T} \tag{9}$$

na qual $\gamma = 1.4$ e $R = 287$ J/kg/K para o ar.

Então a velocidade de cruzeiro da aeronave (V_{cruise}) é:

$$V_{cruise} = M_{cruise} \cdot a_{cruise} \tag{10}$$

Podemos estimar o peso da aeronave no começo do cruzeiro com:

$$W_{cruise} = M_f \cdot W_0 \tag{11}$$

Em seguida, use o módulo de aerodinâmica para calcular os parâmetros da polar de arrasto nas condições de cruzeiro($C_{D0,cruise}$ e K_{cruise}). Use as seguintes configurações para esse caso:

```
Mach = Mach_cruise
altitude = altitude_cruise
n_engines_failed = 0
flap_def = 0.0
slat_def = 0.0
lg_down = 0
h_ground_cruise = 0
Weight = W_cruise
```

Podemos usar a condição $L = W_{cruise}$ para calcular o coeficiente de sustentação de cruzeiro:

$$C_{L,cruise} = \frac{2 \cdot W_{cruise}}{\rho \cdot S_w \cdot V_{cruise}^2} \tag{12}$$

O coeficiente de arrasto correspondente pode ser calculado com a polar de arrasto:

$$C_{D,cruise} = C_{D0,cruise} + K_{cruise} \cdot C_{L,cruise}^2 \tag{13}$$

Podemos usar a condição $T_{cruise} = D$ para encontrar a tração necessária em cruzeiro:

$$T_{cruise} = \frac{\rho V_{cruise}^2}{2} \cdot S_w \cdot C_{D,cruise} \tag{14}$$

Entretanto, precisamos reverter essa configuração de tração de volta para condições de decolagem, para termos uma base de comparação entre os múltiplos requisitos. O fator de correção da tração é (Scholz):

$$k_T = (0.0013 \cdot BPR - 0.0397) \cdot \frac{h_{cruise}}{1000} - 0.0248 \cdot BPR + 0.7125 \tag{15}$$

A tração correspondente de decolagem demandada pela condição de cruzeiro é:

$$T_{0,cruise} = \frac{T_{cruise}}{k_T} \tag{16}$$

A.4 Análise da subida

Já que teremos que analisar múltiplas condições de subida, é recomendado que seja implementada uma função auxiliar para calcular a tração requerida para uma restrição genérica de subida. Defina a função `climb_analysis` que recebe os seguintes inputs:

- gradiente de subida: γ_{climb}
- fator da velocidade de stall: k_s
- altitude de subida: h_{climb}
- estimativa de máximo coeficiente de sustentação: $C_{Lmax,guess}$
- posição do trem de pouso: `lg_down`
- distância ao solo: $h_{ground,climb}$
- deflexão do flap: δ_{flap}
- deflexão do slat: δ_{slat}
- número de motores falhados: $n_{eng,f}$
- fração de peso com respeito ao MTOW: M_f

- fator de correção de tração: k_T
- MTOW: W_0
- área da asa: S_w

O único output dessa função é:

- tração necessária na condição de subida: $T_{0,climb}$

O primeiro conjunto de passos irá usar o $C_{Lmax,guess}$ para estimar um número de Mach de subida. Esse número de Mach vai ser usar para o cálculo da polar de arrasto para que tenhamos um valor de $C_{Lmax,climb}$ atualizado.

Depois de receber os inputs da função, determine as propriedades atmosféricas na condição de subida:

```
T,p,rho,mi = atmosphere(altitude_climb, 288.15)
```

Assim que tivermos a densidade do ar (ρ), podemos determinar a velocidade de stall da aeronave com:

$$V_s = \sqrt{\frac{2 \cdot W_0 \cdot M_f}{\rho \cdot S_w \cdot C_{Lmax,guess}}} \tag{17}$$

Agora usamos o fator de velocidade de stall para calcular a velocidade de subida da aeronave (V_{climb}):

$$V_{climb} = k_s \cdot V_s \tag{18}$$

Podemos usar a temperatura do ar (T) para encontrar a velocidade do som:

$$a_{climb} = \sqrt{\gamma \cdot R \cdot T} \tag{19}$$

Então o número de Mach de subida é:

$$M_{climb} = \frac{V_{climb}}{a_{climb}} \tag{20}$$

A seguir, use o módulo de aerodinâmica para calcular os parâmetros da polar de arrasto nas condições de subida ($C_{D0,climb}$, K_{climb}, e $C_{Lmax,climb}$). Use as seguintes configurações para esse caso:

```
Mach = Mach_climb
altitude = altitude_climb
n_engines_failed = <de acordo com o input da função>
flap_def = <de acordo com o input da função>
max_flap_def = LD_flap_def
slat_def = <de acordo com o input da função>
max_slat_def = LD_slat_def
lg_down = <de acordo com o input da função>
h_ground = h_ground_climb
Weight = W0*Mf
```

Agora que temos um valor atualizado para $C_{Lmax,climb}$, a relação entre a velocidade de subida e velocidade de stall ($V_{climb} = k_s \cdot V_s$) nos permite determinar o C_L de subida:

$$C_{L,climb} = \frac{C_{Lmax,climb}}{k_s^2} \tag{21}$$

O coeficiente de arrasto correspondente pode ser calculado com a polar de arrasto:

$$C_{D,climb} = C_{D0,climb} + K_{climb} \cdot C_{L,climb}^2 \tag{22}$$

Então determine a razão tração-por-peso para essa condição de subida:

$$\left(\frac{T_0}{W_{climb}}\right) = \frac{n_{eng}}{n_{eng} - n_{eng,f}} \cdot \left(\gamma_{climb} + \frac{C_{D,climb}}{C_{L,climb}}\right) \tag{23}$$

Note que essa equação irá ajustar a tração necessária de acordo com o número de motores falhados. Agora podemos, finalmente, calcular a tração necessária com:

$$T_{0,climb} = \left(\frac{T_0}{W_{climb}}\right) \cdot \frac{W_0 \cdot M_f}{k_T} \tag{24}$$

Essa equação já aplica o fator de correção da tração (k_T). Use $k_T = 0.94$ quando o requisito pedir por tração máxima continua ao invés de máxima tração de decolagem.

Você terá que chamar essa função de análise da subida para os seis casos abaixo. Note que o gradiente de subida muda de acordo com o número de motores da aeronave (n_{eng}):

1. **FAR 25.111**
   - $\gamma_{climb} = 0.012$ (2 motores), 0.015 (3 motores), or 0.017 (4 motores)
   - $k_s = 1.2$
   - $h_{climb} = h_{takeoff}$
   - $C_{Lmax,guess} = C_{Lmax,TO}$
   - `lg_down` = 0 (trem de pouso retraído)
   - $h_{ground,climb} = h_{ground}$ (use o input do módulo de desempenho, conforme descrito na Sec. 1.1)
   - $\delta_{flap} = \delta_{flap,TO}$ (flaps de decolagem)
   - $\delta_{slat} = \delta_{slat,TO}$ (slats de decolagem)
   - $n_{eng,f} = 1$ (condição OEI[3])
   - $M_f = 1.0$ (peso de decolagem)
   - $k_T = 1.0$ (tração de decolagem)
2. **FAR 25.121a**
   - $\gamma_{climb} = 0.000$ (2 motores), 0.003 (3 motores), or 0.005 (4 motores)
   - $k_s = 1.1$
   - $h_{climb} = h_{takeoff}$
   - $C_{Lmax,guess} = C_{Lmax,TO}$
   - `lg_down` = 1 (trem de pouso estendido)
   - $h_{ground,climb} = h_{ground}$ (use o input do módulo de desempenho, conforme descrito na Sec. 1.1)
   - $\delta_{flap} = \delta_{flap,TO}$ (flaps de decolagem)
   - $\delta_{slat} = \delta_{slat,TO}$ (slats de decolagem)
   - $n_{eng,f} = 1$ (condição OEI)
   - $M_f = 1.0$ (peso de decolagem)
   - $k_T = 1.0$ (tração de decolagem)
3. **FAR 25.121b**
   - $\gamma_{climb} = 0.024$ (2 motores), 0.027 (3 motores), or 0.030 (4 motores)
   - $k_s = 1.2$
   - $h_{climb} = h_{takeoff}$
   - $C_{Lmax,guess} = C_{Lmax,TO}$
   - `lg_down` = 0 (trem de pouso retraído)
   - $h_{ground,climb} = 0$ (sem efeito solo)
   - $\delta_{flap} = \delta_{flap,TO}$ (flaps de decolagem)
   - $\delta_{slat} = \delta_{slat,TO}$ (slats de decolagem)
   - $n_{eng,f} = 1$ (condição OEI)
   - $M_f = 1.0$ (peso de decolagem)
   - $k_T = 1.0$ (tração de decolagem)
4. **FAR 25.121c**
   - $\gamma_{climb} = 0.012$ (2 motores), 0.015 (3 motores), or 0.017 (4 motores)
   - $k_s = 1.25$

[3] *One Engine Inoperative*, um motor inoperante

- $h_{climb} = h_{takeoff}$
- $C_{Lmax,guess} = C_{Lmax,TO}$
- `lg_down` $= 0$ (trem de pouso retraído)
- $h_{ground,climb} = 0$ (sem efeito solo)
- $\delta_{flap} = 0.0$ (flaps retraídos)
- $\delta_{slat} = 0.0$ (slats retraídos)
- $n_{eng,f} = 1$ (condição OEI)
- $M_f = 1.0$ (peso de decolagem)
- $k_T = 0.94$ (tração máxima contínua)

5. **FAR 25.119**

- $\gamma_{climb} = 0.032$
- $k_s = 1.30$
- $h_{climb} = h_{landing}$
- $C_{Lmax,guess} = C_{Lmax,LD}$
- `lg_down` $= 1$ (trem de pouso estendido)
- $h_{ground,climb} = 0$ (sem efeito solo)
- $\delta_{flap} = \delta_{flap,LD}$ (flaps de pouso)
- $\delta_{slat} = \delta_{slat,LD}$ (slats de pouso)
- $n_{eng,f} = 0$ (condição AEO[4])
- $M_f = MLW/MTOW$ (peso de pouso)
- $k_T = 1.0$ (tração de decolagem)

6. **FAR 25.121d**

- $\gamma_{climb} = 0.021$ (2 motores), 0.024 (3 motores), or 0.027 (4 motores)
- $k_s = 1.40$
- $h_{climb} = h_{landing}$
- $C_{Lmax,guess} = C_{Lmax,LD}$
- `lg_down` $= 1$ (trem de pouso estendido)
- $h_{ground,climb} = 0$ (sem efeito solo)
- $\delta_{flap} = 0.8 * \delta_{flap,LD}$ (flaps de aproximação)
- $\delta_{slat} = 0.8 * \delta_{slat,LD}$ (slats de aproximação)
- $n_{eng,f} = 1$ (condição OEI)
- $M_f = MLW/MTOW$ (peso de pouso)
- $k_T = 1.0$ (tração de decolagem)

A.5 Juntando os outputs

Nós analisamos diversos critérios de desempenho. Podemos juntar todos os valores de tração necessária em um único *array*, que é um dos outputs do módulo de desempenho:

$$\vec{T_0} = \left[\begin{array}{ccccccccc} T_{0,TO} & T_{0,cruise} & T_{0,FAR\,25.111} & T_{0,FAR\,25.121a} & T_{0,FAR\,25.121b} & T_{0,FAR\,25.121c} & T_{0,FAR\,25.119} & T_{0,FAR\,25.121d} \end{array} \right] \tag{25}$$

Outro output do módulo de desempenho é o valor de tração requerido pela fase mais crítica:

$$T_0 = 1.05 \times \max(\vec{T_0}) \tag{26}$$

O fator de 1.05 é usado para ser mais conservador com o projeto.
O terceiro e último output é a área de asa requerida pela restrição de pouso ($S_{w,lan}$), calculada na Sec. A.2.

Algorithm 1 Iteração da tração de decolagem

$\Delta = 1000$ ▷ Defina uma variável auxiliar para controlar o processo iterativo
while $\Delta > 100$ **do**
    Use o módulo de MTOW (Lab05) para computar W_0, W_f, W_e, e $x_{CG,e}$ usando $W_{0,guess}$ e $T_{0,guess}$
    Use o módulo de desempenho para calcular T_0, $\vec{T}_0$, e $S_{w,lan}$ usando W_0
    $\Delta = T_0 - T_{0,guess}$ ▷ Calcule a diferença entre a estimativa e o valor calculado
    $T_{0,guess} = T_0$ ▷ Defina o valor calculado como a nova estimativa
    $W_{0,guess} = W_0$ ▷ Também atualize a estimativa de MTOW para acelerar a próxima iteração

A.6 Iteração da tração de decolagem

A tração de decolagem (T_0) deve ser determinada iterativamente dadas as relações implícitas com o MTOW. O Algoritmo 1 abaixo descreve esse procedimento iterativo:

O processo deve convergir após algumas iterações. Os valores de W_0, W_f, W_e, $x_{CG,e}$, T_0, $\vec{T}_0$, e $S_{w,lan}$ depois do código sair do `while` são os valores convergidos que devem compor os outputs do módulo atual.

[4] *All Engines Operative*, todos os motores operantes