

INSTITUTO TECNOLÓGICO DE AERONÁUTICA

LABORATÓRIO DE MVO - 31

DESEMPENHO DE AERONAVES

2º LAB: VOO NO PLANO VERTICAL - RELATÓRIO

PRELIMINAR

Francisco Arthur Bonfim Azevedo

25 de abril de 2019



Sumário

1	Resultados	3
1.1	Item 1	3
1.1.1	A	3
1.1.2	B	3
1.1.3	C	3
1.1.4	D	3
1.1.5	E	3
1.1.6	F	4
1.1.7	G	4
1.2	Item 2	4
1.3	Item 3	5
1.4	Item 4	5
1.5	Item 5	5
1.6	Item 6	5
1.7	Item 7	6
1.8	Item 8	6
1.9	Item 9	6
1.10	Item 10	7
1.11	Item 11	8
1.12	Item 12	8
1.13	Item 13	8
2	Anexos	9

1 Resultados

1.1 Item 1

1.1.1 A

Com

$$L = W \quad (1)$$

$$L = \frac{1}{2} \rho V^2 C_L S \quad (2)$$

$$C_D = C_{D0} + K C_L^2 \quad (3)$$

$$D = \frac{1}{2} \rho V^2 C_D S \quad (4)$$

$$D - T + W \gamma \quad (5)$$

$$P = TV \quad (6)$$

Obtêm-se

$$P = WV\gamma + \frac{1}{2} \rho V^3 C_{D0} + \frac{2KW^2}{\rho VS} \quad (7)$$

Que é a equação da potência requerida para cada velocidade, sendo P a potencia, W o peso, V a velocidade, γ o angulo de trajetória, ρ a densidade local do ar, C_{D0} o arrasto parasita, K o coeficiente de arrasto induzido e S a área de referência.

1.1.2 B

Sendo a razão de subida $V_{Climb} = V\gamma$

$$P = WV_{Climb} + \frac{1}{2} \rho V^3 C_{D0} + \frac{2KW^2}{\rho VS} \quad (8)$$

1.1.3 C

Derivando a equação da potencia requerida em B em relação a velocidade e igualando a zero, obtêm-se a velocidade de potência mínima e essa tal potência, que são

$$V = \sqrt{\frac{2W}{\rho S} \sqrt{\frac{K}{3C_{D0}}}} \quad (9)$$

$$P = WV_{Climb} + \frac{4}{3} \sqrt{\frac{2W^3}{\rho S} \sqrt{3K^3 C_{D0}}} \quad (10)$$

1.1.4 D

Para isso, basta substituir tais valores na equação da potência requerida e resolver a equação do 4º grau resultante. A equação possui 4 raízes, devido o seu grau, no entanto, haverá 2 raízes reais, que possuem significado físico e equivalem as velocidades desejadas, e duas raízes complexas sem significado. Caso se esteja equacionando uma aeronave acima do seu teto de voo, todas as raízes serão complexas.

$$\frac{1}{2} \rho V^4 C_{D0} + (WV_{Climb} - P)V + \frac{2KW^2}{\rho S} = 0 \quad (11)$$

1.1.5 E

Para obter as velocidades máxima e mínima basta, na equação de obtenção da velocidade, considerar a manete no máximo e resolver a equação do quarto grau resultante, que possuirá duas raízes reais que são o máximo e o

mínimo das velocidades.

$$\frac{1}{2}\rho V^4 C_{D0} + (WV_{Climb} - \eta P_{max} \frac{\rho^m}{\rho_s})V + \frac{2KW^2}{\rho S} = 0 \quad (12)$$

1.1.6 F

Para este caso, basta maximizar a potência com o intuito de fazer a aeronave subir o máximo possível. Com isso, obter-se-á a velocidade e a razão de subida como

$$V = \sqrt{\frac{2W}{\rho S} \sqrt{\frac{K}{3C_{D0}}}} \quad (13)$$

$$V_{Climb} = \frac{\eta P_{max}}{W} \frac{\rho^m}{\rho_s} - \frac{4}{3} \sqrt{\frac{2W^3}{\rho S} \sqrt{3K^3 C_{D0}}} \quad (14)$$

1.1.7 G

Para o cálculo do teto de voo, basta forçar que a potência mínima necessária para voo seja a máxima disponível para a aeronave, com isso, obtêm-se a seguinte expressão

$$\eta P_{max} \frac{\rho^m}{\rho_s} = WV_{Climb} + \frac{4}{3} \sqrt{\frac{2W^3}{\rho S} \sqrt{3K^3 C_{D0}}} \quad (15)$$

Daí é possível descobrir a densidade do ar local, da qual tira-se a altitude correspondente ao teto de voo.

1.2 Item 2

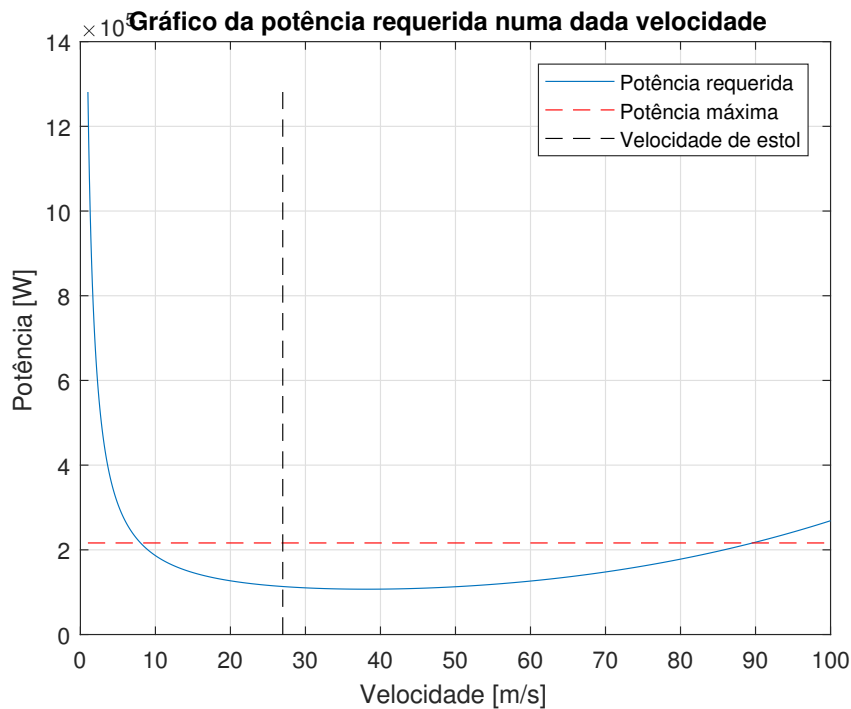


Figura 1: Gráfico da potência requerida em função da velocidade. Nota-se os limites de potência máxima e velocidade de estol.

1.3 Item 3

A velocidade de potência mínima graficamente vale 38.1622 m/s, já através das equações da teoria obtêm-se 38.1197, apresentando um desvio de 0.113 %. Além disso, a potência mínima vale $1.075 \cdot 10^5 \text{ W}$, o ângulo de ataque vale 9.8488° , a deflexão de profundor vale 8.9053° e a posição da manete está em 0.74 ou 74%.

1.4 Item 4

A velocidade máxima teórica obtida foi 68.7721, no entanto, em uma situação real, essa velocidade máxima seria a velocidade de estol nessa altitude. Já a deflexão de profundor é 1.5045° e o ângulo de ataque é 2.9104° considerando essa velocidade máxima teórica.

1.5 Item 5

A velocidade mínima teórica obtida foi 15.3088. Já a deflexão de profundor é 64.4611° e o ângulo de ataque é 61.9323° , que em uma situação real seria limitado pelo ângulo de estol.

1.6 Item 6

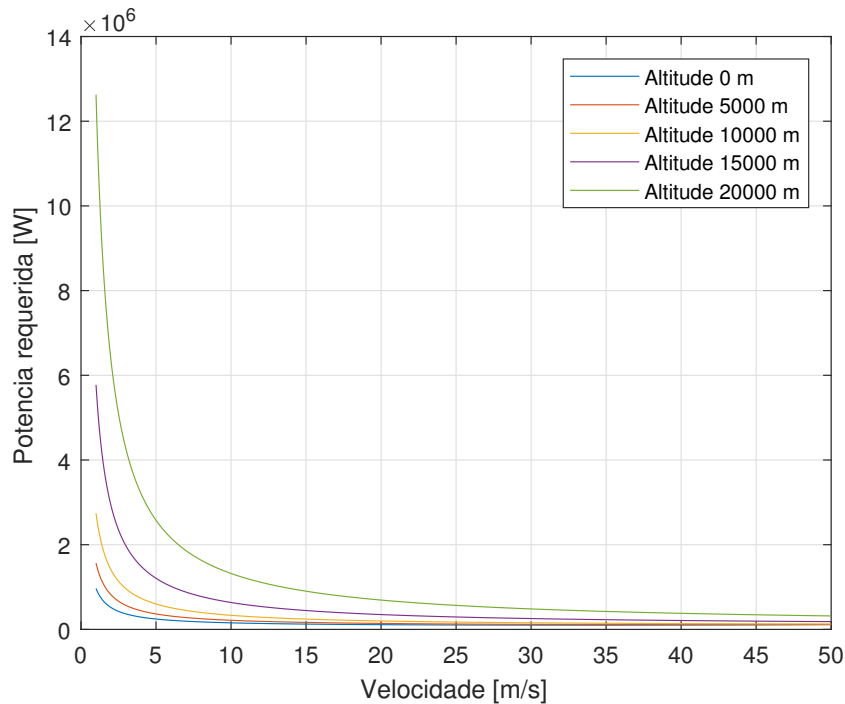


Figura 2: Gráfico da potência requerida para vários valores de altitude.

Nota-se que a potência requerida tende a aumentar com o aumento da altitude

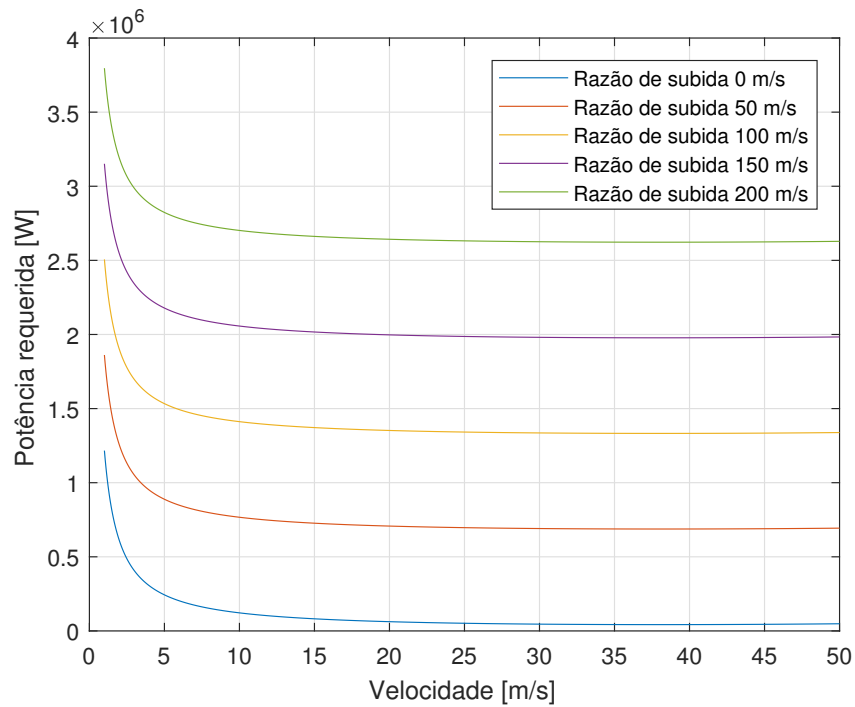


Figura 3: Gráfico da potência requerida para vários valores de razão de subida.

Nota-se que a potência requerida tende a aumentar com o aumento da razão de subida.

1.7 Item 7

Aplicando-se as equações de taxa máxima de subida, ela ocorrerá quando a velocidade for $38.1197m/s$. Essa taxa será de $7.9151m/s$. O coeficiente de sustentação será de 1.2019, o ângulo de ataque será 9.8488° , a deflexão de profundor será 8.9053°

1.8 Item 8

O teto de voo será $6455m$.

1.9 Item 9

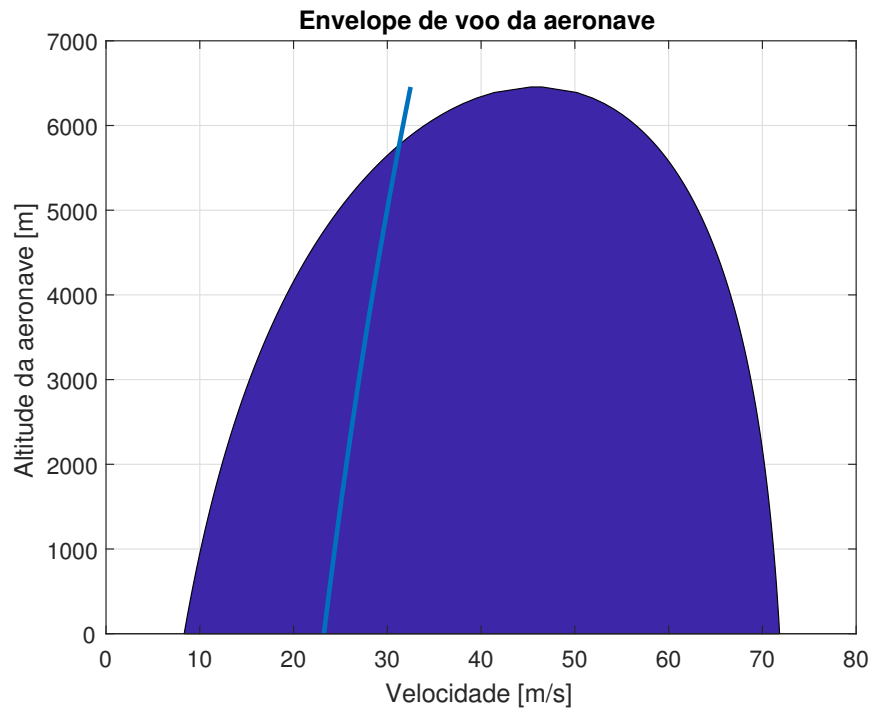


Figura 4: Envelope de voo para condição de razão de subida 5 m/s. Em azul está a curva representando a velocidade de estol.

1.10 Item 10

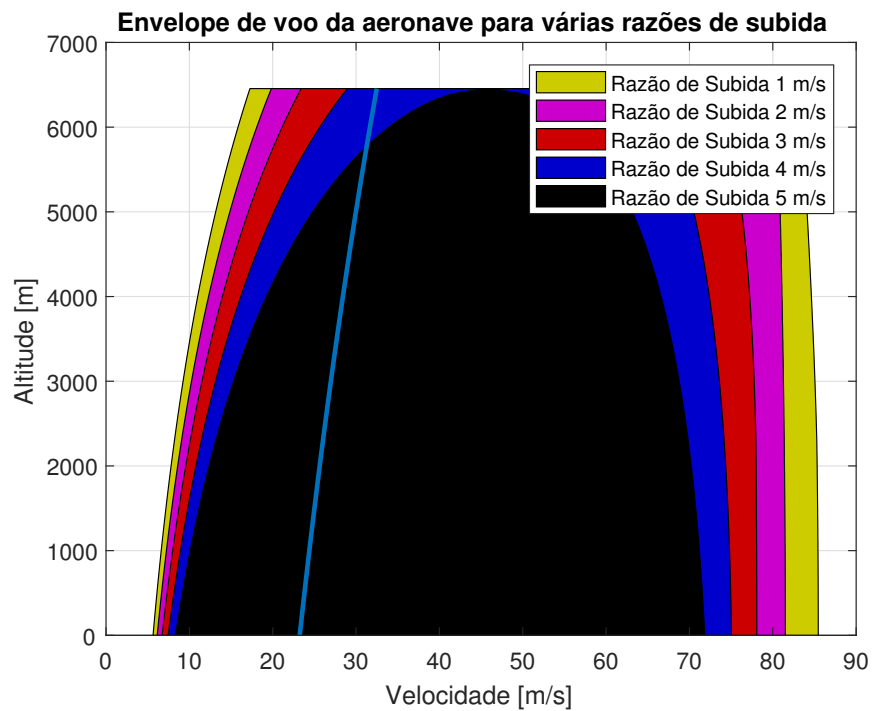


Figura 5: Envelope de voo para várias condições de razão de subida.

Nota-se que, quanto maior a razão de subida, menor será a região do envelope de voo.

1.11 Item 11

As curvas apresentam o mesmo valor de tração no ponto de velocidade mínima para aquela dada altitude. Se a aeronave recebesse uma rajada de proa a velocidade relativa dela com o escoamento aumentaria, deslocando o ponto do gráfico para a direita, requerendo uma tração menor que a disponível, sendo possível manter a aeronave em voo nessa condição. Já se recebesse de popa, sua velocidade relativa do escoamento diminuiria e a tração requerida seria menor que a disponível, não sendo possível manter a aeronave em voo nessa condição.

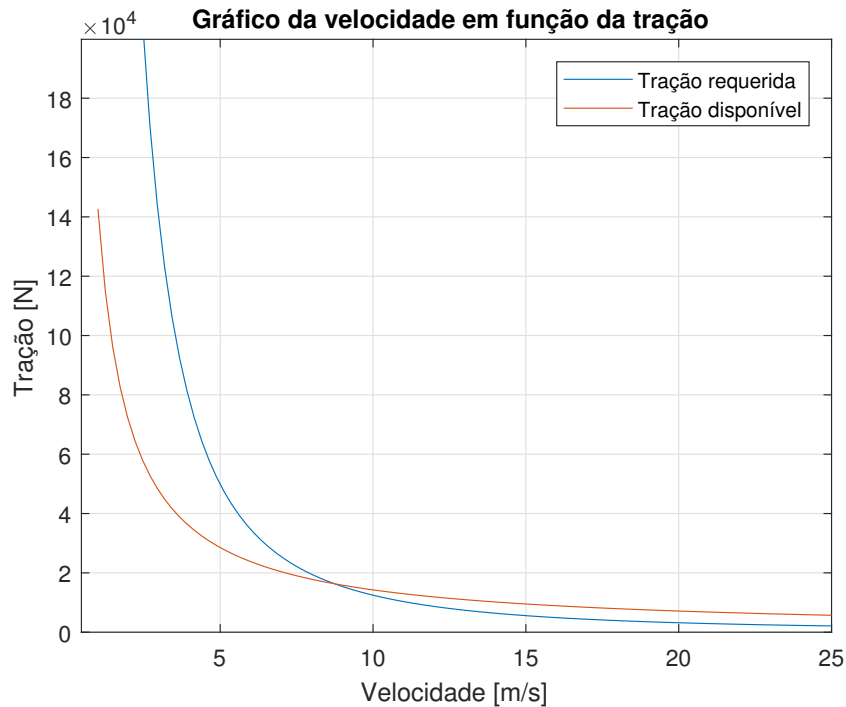


Figura 6: Envelope de voo para várias condições de razão de subida.

1.12 Item 12

Será incluído na versão de relatório final.

1.13 Item 13

Será incluído na versão de relatório final.

2 Anexos

```
1 W = 1315*9.81;% kg
2 s = 16.25;% m2
3 Pmax_s = 216253; %W
4
5
6 P = @(delt, eta, rho) delt*eta*(rho./1.225).^0.6*Pmax_s; %W
7 Cd0 = 0.026;
8 k = 0.054;
9
10 CD = @(CL) Cd0 + k*CL^2;
11
12 CLmax = 2.4;
13 CL = @(alfa) 0.02 + 0.12*alfa;
14 CM = @(alfa, delta_profund) 0.12 - 0.08*alfa + 0.075*delta_profund;
15 nmax = 2;
16 Vclimb = 5;%m/s
17 eta = 0.8;
18
19
20
21 %% item 2
22 [~,~,~,rho] = atmosisa(3000);
23 Vel = linspace(1,100,1000);
24 Vstall = sqrt(2*W/(rho*s*CLmax));
25 Preq = W*Vclimb + 0.5*rho*Vel.^3*s*Cd0+2*k*W^2./(rho*Vel*s);
26
27 figure
28 plot(Vel, Preq)
29 hold on
30 grid on
31 plot(Vel, Pmax_s*ones(1,1000), '—r')
32 plot(Vstall*ones(1,1000), linspace(0,max(Preq),1000), '—k')
33 hold off
34 xlabel('Velocidade [m/s]')
35 ylabel('Potência [W]')
36 title('Gráfico da potência requerida numa dada velocidade');
37 legend('Potência requerida','Potência máxima','Velocidade de estol');
38
39 %% item 3
40 % Caso do grafico
41 PminGraf = min(Preq)
42 VpotMinGraf = interp1(Preq, Vel, PminGraf)
43
44 % Caso da formula de potencia minima
45 VpotMinTeo = sqrt((2*W/(rho*s))*sqrt(k/(3*Cd0)))
46 PminTeo = W*Vclimb + 0.5*rho*VpotMinTeo.^3*s*Cd0+2*k*W^2./(rho*VpotMinTeo*s)
47
48 % Erro relativo
49 ErroVpotMin = abs(VpotMinTeo - VpotMinGraf)/VpotMinGraf*100
50 ErroPmin = abs(PminGraf-PminTeo)/PminGraf*100
51
52 % angulo de ataque
53 CLPotMin = W/(0.5*rho*VpotMinTeo^2*s);
54 alfaPotMin = fsolve(@(alfa) CL(alfa) - CLPotMin, 0.5)
55
56 % Deflexão de profundor
57 deflexProfundPotMin = fsolve(@(delta) CM(alfaPotMin, delta), 0.5)
58
```

```

59 % Posi o de manete
60 PosicManetePotMin = fsolve(@(delta) P(delta,eta,rho)-PminTeo,10^5)
61
62 %% Item 4
63 poli = [0.5*rho*s*Cd0 0 0 (W*Vclimb-eta*(rho/1.225)^0.6*Pmax_s) 2*k*W^2/(
        rho*s)];
64 VMaxMin = sort(roots(poli));
65 Vmin = VMaxMin(1,1)
66 Vmax = VMaxMin(2,1)
67
68 % angulo de ataque
69 CLVelMax = W/(0.5*rho*Vmax^2*s);
70 alfaVelMax = fsolve(@(alfa) CL(alfa) - CLVelMax,0.5)
71
72 % Deflex o de profundor
73 deflexProfundVelMax = fsolve(@(delta)CM(alfaVelMax,delta),0.5)
74
75 %% Item 5
76
77 % angulo de ataque
78 CLVelMin = W/(0.5*rho*Vmin^2*s);
79 alfaVelMin = fsolve(@(alfa) CL(alfa) - CLVelMin,0.5)
80
81 % Deflex o de profundor
82 deflexProfundVelMin = fsolve(@(delta)CM(alfaVelMin,delta),0.5)
83
84 %% Item 6
85 % Variando a altitude
86 HVarAlt = 0:5000:20000;
87 [~,~,~,rhoVarAlt] = atmosisa(HVarAlt);
88 rhoVarAlt = rhoVarAlt';
89 Vel = linspace(1,50,1000);
90 PreqVarAlt = W*Vclimb +0.5*rhoVarAlt*Vel.^3*s*Cd0+2*k*W^2./(rhoVarAlt*Vel*
        s);
91
92 figure
93 plot(Vel,PreqVarAlt)
94 grid on
95 xlabel('Velocidade [m/s]')
96 ylabel('Potencia requerida [W]')
97 legend('Altitude 0 m', 'Altitude 5000 m', 'Altitude 10000 m', 'Altitude
        15000 m', 'Altitude 20000 m');
98
99 % Variando a Vclimb
100 HVarRazSub = 3000;
101 [~,~,~,rhoVarRazSub] = atmosisa(HVarRazSub);
102 Vel = linspace(1,50,1000);
103 VclimbVarRazSub = (0:50:200)';
104 PreqVarRazSub = W*VclimbVarRazSub +0.5*rhoVarRazSub*Vel.^3*s*Cd0+2*k*W
        ^2./(rhoVarRazSub*Vel*s);
105
106 figure
107 plot(Vel,PreqVarRazSub)
108 grid on
109 xlabel('Velocidade [m/s]');
110 ylabel('Pot ncia requerida [W]');
111 legend('Raz o de subida 0 m/s', 'Raz o de subida 50 m/s', 'Raz o de
        subida 100 m/s', 'Raz o de subida 150 m/s', 'Raz o de subida 200 m/s')
        ;
112
113 %% Item 7

```

```
114
115 [~,~,~,rho] = atmosisa(3000);
116 VVelClimbMax = sqrt((2*W/(rho*s))*sqrt(k/(3*Cd0)))
117 VclimbMax = P(1,eta,rho)/W - (4/3)*sqrt(2*W/(rho*s))*sqrt(3*k^3*Cd0))
118
119 % Angulo de ataque
120 CLVclimbMax = W/(0.5*rho*VVelClimbMax^2*s)
121 alfaVclimbMax = fsolve(@(alfa) CL(alfa) - CLVclimbMax,0.5)
122
123 % Deflexão de profundor
124 deflexProfundVclimbMax = fsolve(@(delta)CM(alfaVclimbMax,delta),0.5)
125
126 %% Item 8
127 VclimbTeto = 5;
128
129 TetoVoo = @(rhoTeto) W*VclimbTeto + (4/3)*sqrt(2*W^3/(rhoTeto*s))*sqrt(3*k
    ^3*Cd0)) - P(1,eta,rhoTeto); % corrigir
130
131 rhoTeto = fsolve(TetoVoo, 0.8);
132
133 HtetoAux = 0:5:20000;
134 [~,~,~,rhoTetoAux] = atmosisa(HtetoAux);
135 rhoTetoAuxPosic = find(abs(rhoTetoAux - rhoTeto) < 10^-4);
136 Hteto = HtetoAux(rhoTetoAuxPosic(1,1))
137 clear HtetoAux rhoTetoAux rhoTetoAuxPosic
138
139 %% Item 9
140 HEnvel = linspace(0,Hteto,100);
141 Vclimb = 5;
142 [~,~,~,rhoEnvel] = atmosisa(HEnvel);
143 rhoEnvel = rhoEnvel';
144
145 PoliVreq = [0.5*rhoEnvel*s*Cd0 zeros(length(rhoEnvel),1) zeros(length(
    rhoEnvel),1) (W*Vclimb - P(1,eta,rhoEnvel)) 2*k*W^2./(rhoEnvel*s)];
146
147 Vreq = zeros(length(rhoEnvel),1);
148 Vreq2 = zeros(length(rhoEnvel),1);
149
150 for i = 1:length(rhoEnvel)
151 VreqAux = sort(roots(PoliVreq(i,:)));
152 Vreq(i) = VreqAux(1,1);
153 Vreq2(i) = VreqAux(2,1);
154 end
155
156 HEnvel = [HEnvel HEnvel];
157 Vreq = [Vreq' sort(Vreq2','descend')];
158 Vstall = sqrt(2*W./(rhoEnvel*s*CLmax));
159
160 figure
161 area(Vreq,HEnvel)
162 hold on
163 plot(Vstall, HEnvel(1,1:100),'LineWidth',2)
164 xlabel('Velocidade [m/s]')
165 ylabel('Altitude da aeronave [m]')
166 title('Envelope de voo da aeronave')
167 grid on
168 hold off
169
170 %% Item 10
171
172 HEnvel1 = linspace(0,Hteto,100);
```

```

173 Vclimb = linspace(1,5,5);
174 [~,~,~,rhoEnvel] = atmosisa(HEnvel1);
175 rhoEnvel = rhoEnvel';
176
177 PoliVreq1 = [0.5*rhoEnvel*s*Cd0 zeros(length(rhoEnvel),1) zeros(length(
    rhoEnvel),1) (W*Vclimb(1) - P(1,eta,rhoEnvel)) 2*k*W^2./(rhoEnvel*s)];
178
179 PoliVreq2 = [0.5*rhoEnvel*s*Cd0 zeros(length(rhoEnvel),1) zeros(length(
    rhoEnvel),1) (W*Vclimb(2) - P(1,eta,rhoEnvel)) 2*k*W^2./(rhoEnvel*s)];
180
181 PoliVreq3 = [0.5*rhoEnvel*s*Cd0 zeros(length(rhoEnvel),1) zeros(length(
    rhoEnvel),1) (W*Vclimb(3) - P(1,eta,rhoEnvel)) 2*k*W^2./(rhoEnvel*s)];
182
183 PoliVreq4 = [0.5*rhoEnvel*s*Cd0 zeros(length(rhoEnvel),1) zeros(length(
    rhoEnvel),1) (W*Vclimb(4) - P(1,eta,rhoEnvel)) 2*k*W^2./(rhoEnvel*s)];
184
185 PoliVreq5 = [0.5*rhoEnvel*s*Cd0 zeros(length(rhoEnvel),1) zeros(length(
    rhoEnvel),1) (W*Vclimb(5) - P(1,eta,rhoEnvel)) 2*k*W^2./(rhoEnvel*s)];
186
187
188
189 Vreq1 = zeros(length(rhoEnvel),1);
190 Vreq12 = zeros(length(rhoEnvel),1);
191 Vreq2 = zeros(length(rhoEnvel),1);
192 Vreq22 = zeros(length(rhoEnvel),1);
193 Vreq3 = zeros(length(rhoEnvel),1);
194 Vreq32 = zeros(length(rhoEnvel),1);
195 Vreq4 = zeros(length(rhoEnvel),1);
196 Vreq42 = zeros(length(rhoEnvel),1);
197 Vreq5 = zeros(length(rhoEnvel),1);
198 Vreq52 = zeros(length(rhoEnvel),1);
199
200 for i = 1:length(rhoEnvel)
201     VreqAux1 = sort(roots(PoliVreq1(i,:)));
202     Vreq1(i) = VreqAux1(1,1);
203     Vreq12(i) = VreqAux1(2,1);
204
205     VreqAux2 = sort(roots(PoliVreq2(i,:)));
206     Vreq2(i) = VreqAux2(1,1);
207     Vreq22(i) = VreqAux2(2,1);
208
209     VreqAux3 = sort(roots(PoliVreq3(i,:)));
210     Vreq3(i) = VreqAux3(1,1);
211     Vreq32(i) = VreqAux3(2,1);
212
213     VreqAux4 = sort(roots(PoliVreq4(i,:)));
214     Vreq4(i) = VreqAux4(1,1);
215     Vreq42(i) = VreqAux4(2,1);
216
217     VreqAux5 = sort(roots(PoliVreq5(i,:)));
218     Vreq5(i) = VreqAux5(1,1);
219     Vreq52(i) = VreqAux5(2,1);
220 end
221
222 HEnvel1 = [HEnvel1 HEnvel1];
223 Vreq1 = [Vreq1' sort(Vreq12','descend')];
224
225 Vreq2 = [Vreq2' sort(Vreq22','descend')];
226
227 Vreq3 = [Vreq3' sort(Vreq32','descend')];
228

```

```
229 Vreq4 = [Vreq4' sort(Vreq42','descend')];
230
231 Vreq5 = [Vreq5' sort(Vreq52','descend')];
232
233 figure
234 h1=area(Vreq1,HEnvel1);
235 h1.FaceColor = [0.8 0.8 0];
236 hold on
237 h2=area(Vreq2,HEnvel1);
238 h2.FaceColor = [0.8 0 0.8];
239 h3=area(Vreq3,HEnvel1);
240 h3.FaceColor = [0.8 0 0];
241 h4=area(Vreq4,HEnvel1);
242 h4.FaceColor = [0 0 0.8];
243 h5=area(Vreq5,HEnvel1);
244 h5.FaceColor = [0 0 0];
245 plot(Vstall, HEnvel(1,1:100),'LineWidth',2)
246 grid on
247 hold off
248
249 xlabel('Velocidade [m/s]')
250 ylabel('Altitude [m]')
251 legend('Raz o de Subida 1 m/s', 'Raz o de Subida 2 m/s', 'Raz o de
252         Subida 3 m/s', 'Raz o de Subida 4 m/s', 'Raz o de Subida 5 m/s')
253 title('Envelope de voo da aeronave para v rias raz es de subida')
254 %% Item 11
255
256 [~,~,~,rhoDisp] = atmosisa(Hteto/2);
257 Vel = linspace(1,25);
258 VclimbDisp = 0;
259
260 Treq = W*VclimbDisp./Vel + 0.5*rhoDisp*Vel.^2*s*Cd0 + 2*k*W^2./(rhoDisp*s*
261         Vel.^2);
262 TDisp = P(1,eta,rhoDisp)./Vel;
263
264 figure
265 plot(Vel,Treq)
266 hold on
267 plot(Vel,TDisp)
268 hold off
269 grid on
270 xlabel('Velocidade [m/s]');
271 ylabel('Tra o [N]')
272 title('Gr fico da velocidade em fun o da tra o')
273 legend('Tra o requerida', 'Tra o dispon vel')
```