

LAB #2

TC

2022/23

Nº 59752

Nº 59939

Nº 60101

Nº 60461

Control Theory

Labwork 2



Altitude Control of a Commercial Airliner

I. Modelo da Instalação

a) Função de Transferência do Ângulo de Inclinação como base do Modelo da Instalação

Regularmente, nos casos abordados na unidade curricular de Teoria de Controlo, existe a necessidade de criar um modelo matemático referente à instalação, com base nos fenómenos físicos que descrevem o problema em estudo, bem como, na aquisição de dados experimentais.

Neste caso, pretendemos inicialmente definir a função de transferência associada ao ângulo de inclinação θ , que tem como sinal de entrada o valor do elevador de posição δ_e e como saída o ângulo de inclinação. É a este bloco que adicionaremos, posteriormente, dois controladores em cascata: um que atuará no valor do elevador de posição, tendo como sinal de entrada o valor de referência para o ângulo de inclinação R_θ , e outro que atuará neste último, tendo como sinal de entrada o valor de referência da altitude R_H .

Assim, definimos a base do modelo de instalação de acordo com os parâmetros descritos no protocolo para a função de transferência associada ao ângulo de inclinação, com recurso à nossa ferramenta de cálculo numérico, MatLab:

```
b1 = 2.305148400378274;  
b2 = 0.861103913823527;  
b3 = 0.010626751575719;  
a1 = 0.740551982652778;  
a2 = 1.380681506199432;  
a3 = 0.019931577735793;  
a4 = 0.003595969889691;  
  
Fs = tf([ b1 b2 b3 ], [ 1 a1 a2 a3 a4]);
```

b) Validação da Função de Transferência

Com o objetivo de verificar se o modelo da instalação é coerente com a planta real, vamos comparar a resposta do simulador com a da função acima modelada, em ângulo de inclinação. Para tal, dirigimo-nos ao nosso simulador em modo de **controlo manual** e aplicamos uma série de *steps* diretamente no elevador – uma vez que é este o sinal de entrada da função de transferência que definimos.

Do ficheiro criado a partir deste processo, é-nos apresentado, entre outros, o gráfico da Figura 1 a), que descreve a resposta do simulador aos *steps* aplicados ao elevador, em ângulo de inclinação.

Procedemos, então, à comparação de respostas simulador/modelo: sobrepomos os dados de resposta no simulador com a resposta do modelo ao mesmo sinal de entrada anteriormente utilizado, fazendo uso da função `lsim()` no seguinte código:

```
figure(1)  
ysim = lsim(Fs, u, t);  
plot(t, [theta ysim] * rad2deg); ylabel('Pitch Angle [deg]'); xlabel('t [s]');
```

Obtemos, assim, o gráfico da Figura 1 b), em que a resposta do simulador é apresentada a azul, e a do nosso modelo, a laranja.

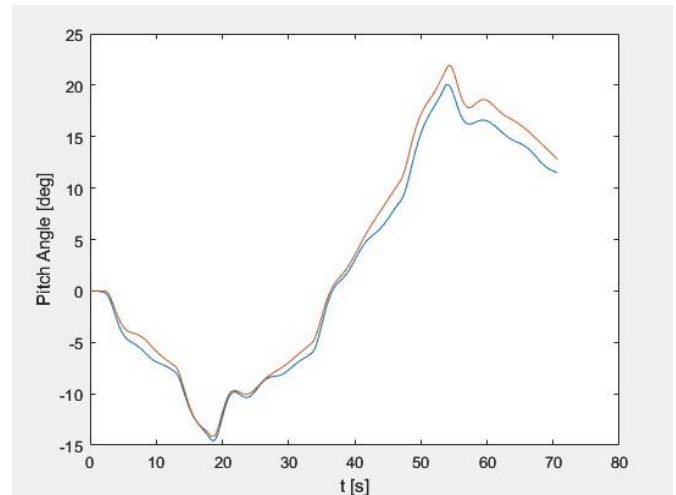
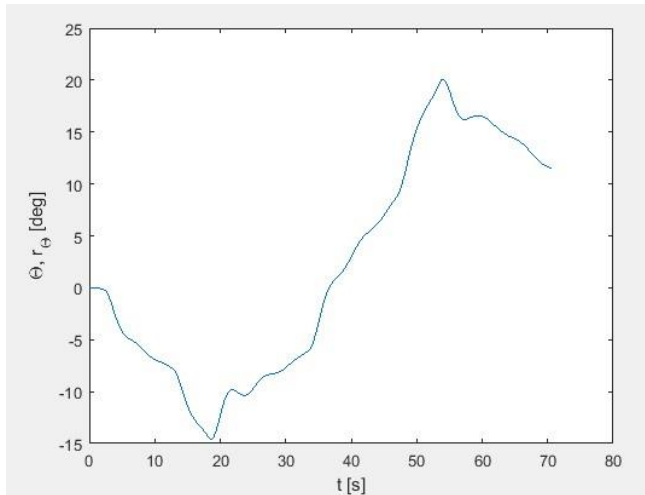


Figura 1 a) – Resposta do simulador aos *steps* aplicados ao elevador (ângulo de inclinação)

Figura 1 b) – Comparação da resposta do simulador com a resposta da função de transferência modelada (ângulo de inclinação)

Os dois gráficos da Figura 1 b) são bastante semelhantes, pelo que se conclui que a função de transferência anteriormente obtida é adequada para funcionar como base do modelo da instalação.

c) Estabilidade da Dinâmica Longitudinal do Avião

Com o objetivo de concluir sobre a estabilidade da dinâmica longitudinal do avião, analisamos o mapa de polos-zeros da função de transferência anteriormente estudada. Para tal, utilizamos a função `pzmap()` no seguinte código, que resulta no diagrama da [Figura 3](#).

```
figure(2)
pzmap(Fs);
```

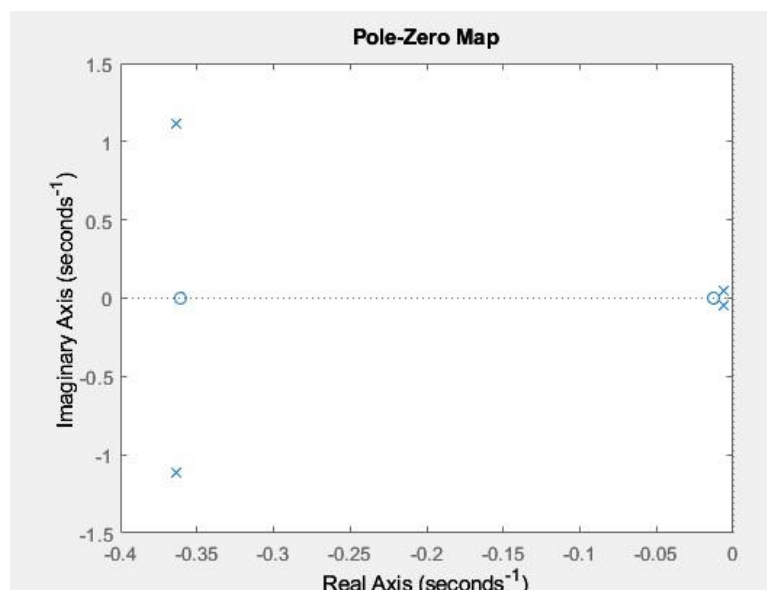


Figura 2 – Mapa de Polos-Zeros de $F_s(s)$

Uma vez que a parte real de todos os polos é negativa, isto é, encontram-se todos à esquerda do eixo imaginário, conclui-se que a função de transferência com que trabalhamos é estável: existe estabilidade da dinâmica longitudinal do avião.

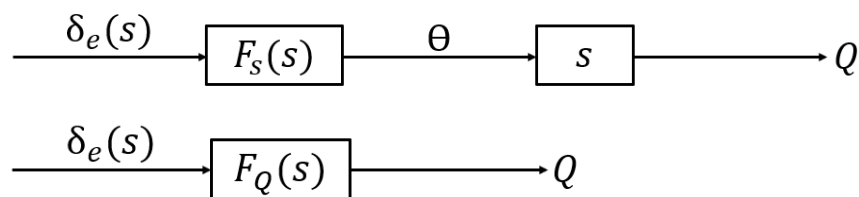
d) Funções de Transferência para a Taxa de Inclinação e a Altitude

A partir dos aspetos físicos do problema, podemos concluir que tanto a Taxa de Inclinação Q como a Altitude H podem ser derivadas a partir do Ângulo de Inclinação, cuja função de transferência já nos é conhecida.

1. Taxa de Inclinação: é a derivada do ângulo de inclinação.

$$F_Q(s) = \frac{Q(s)}{\delta_e(s)} \Rightarrow F_Q(s) = sF_s(s)$$

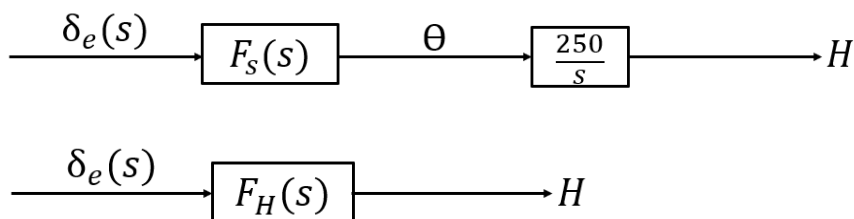
Diagramas de Blocos (equivalentes):



2. Altitude: a variação da altitude é igual à velocidade do avião - 250 m/s - a multiplicar pelo seno do ângulo de inclinação - que pode ser aproximado pelo próprio ângulo de inclinação, para inclinações reduzidas. Isto é, a altitude é igual ao integral da multiplicação de 250 pelo ângulo de inclinação.

$$F_H(s) = \frac{H(s)}{\delta_e(s)} \Rightarrow F_H(s) = \frac{250}{s} F_s(s)$$

Diagramas de Blocos (equivalentes):



II. Desenvolvimento e Testagem do Controlador de Inclinação

a) *Design* do Controlador utilizando o Diagrama do Lugar de Raízes

O diagrama do lugar de raízes representa a localização geométrica das raízes (polos e zeros) de um sistema em anel fechado no plano complexo s , permitindo examinar como é que estas raízes mudam com a variação de um determinado parâmetro, geralmente um ganho associado ao sistema. Para além de critério de estabilidade, esta técnica pode ser utilizada para determinar o fator de amortecimento (D) e a frequência natural (w_n) de um sistema de controlo por retroação.

Nesta fase, pretendemos recorrer ao diagrama do lugar de raízes para desenvolver um controlador, responsável por regular o ângulo de inclinação.

$$\delta_e(t) = K_\theta[R_\theta(t) - \theta(t)] - K_Q Q(t)$$

Nesta expressão, R_θ é o valor de referência do ângulo de inclinação, que é o sinal de entrada do sistema de controlo, e os valores K_θ e K_Q são os ganhos do controlador de ângulo de inclinação e taxa de inclinação, respetivamente, que serão adiante definidos durante o desenvolvimento do controlador desejado.

No desenvolvimento do controlador de inclinação, devemos cumprir alguns requisitos, ao que chamamos a especificação: o sistema de controlo deve apresentar, como resposta a uma entrada *step*, uma sobre-elevação inferior a 5% e um tempo de subida máximo de 0,9 s.

Determinemos, então, os ganhos K_θ e K_Q :

1. Construção da função de especificação

Escolhemos uma sobre-elevação de 5% e um tempo de subida máximo de 0,9 s.

Para projetar a função de especificação, devemos encontrar os valores de D e w_n que satisfaçam as condições de sobre-elevação e de tempo de subida, através das seguintes expressões:

$$\omega_n = \frac{3,7D}{t_{r[90\%]}} \quad \text{e} \quad s[\%] = 100e^{-D\pi/\sqrt{1-D^2}} \Rightarrow D = \sqrt{\frac{2\ln(s[\%]/100)}{\pi^2 + 2\ln(s[\%]/100)}}$$

O que é incorporado no seguinte código:

```
s_spec = 5.0;
tr_spec = 0.9;
D = sqrt(log(s_spec/100)^2/(pi^2+log(s_spec/100)^2))
wn = 3.7 * D / tr_spec
Hs_spec = tf( wn^2, [1 2*D*wn wn^2] );
```

2. Confirmação dos requisitos para a resposta ao *step*

Ao aplicar a função *step* à função de especificação, obtemos o gráfico da Figura 3, onde verificamos que a sobre-elevação é de 5% e o tempo de subida é de 0.9 s, como desejado.

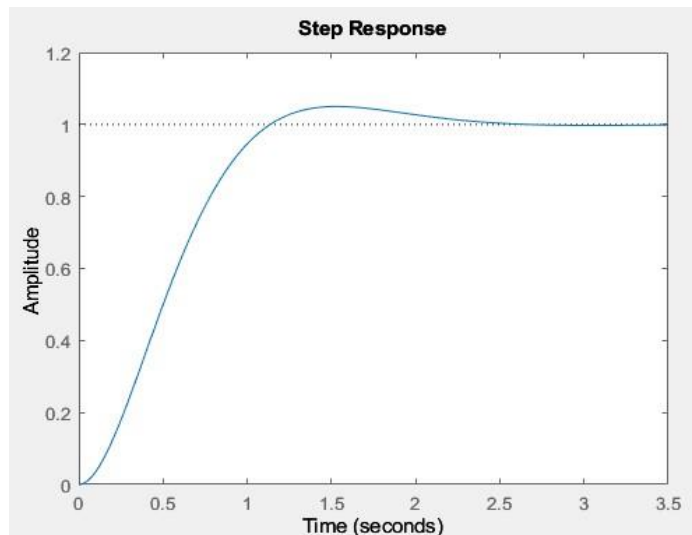


Figura 3 – Resposta da função de especificação ao degrau unitário

3. Polos da função de especificação

De modo a encontrar os polos da função de especificação construída, utilizamos a função `pole()` que resulta em: $-1.9579 + 2.0532i$ e $-1.9579 - 2.0532i$.

4. Fechar o sistema de controlo por retroação

Para encontrar um sistema fechado para o qual um dado ganho K_θ nos dá os polos da função de especificação, temos de inserir um zero de retroação (relacionado com ambos os ganhos K_θ e K_Q) que permite que os traços do diagrama cruzem estes polos. Definimos este zero como $-\frac{1}{\alpha}$ e “fechamos” o sistema de controlo do seguinte modo:

```
 $\alpha = ?$ 
Us = tf([ $\alpha$  1], 1)
Gs = series(Fs,Us)
```

Por tentativa e erro, encontraremos no seguinte passo o α de que precisamos, e sabemos que se relaciona com os ganhos através da expressão $\alpha = \frac{K_Q}{K_\theta}$.

5. Computação do diagrama do lugar de raízes

Aplicamos, agora, o diagrama de lugar de raízes ao sistema de controlo “fechado”¹ para diferentes valores de α até encontrar aquele que faz com que os traços do diagrama cruzem os polos desejados para determinado valor de K_θ .

```
figure(3)
rlocus(Gs)
hold on
plot( real( pole(Hs_spec) ), imag( pole(Hs_spec)), 'rs' )
hold off
```

Nota¹: o diagrama de lugar de raízes é sempre desenhado para um sistema em anel aberto. Neste caso, uma vez que a retroação é equivalente a colocar os blocos em série (comportam-se da mesma maneira a nível de diagrama do lugar de raízes), é isto que fazemos no passo 4., de modo que a ferramenta MatLab não “reconhece” o sistema como estando em anel fechado.

Quando tal acontece, registamos os valores de α e de K_θ . O diagrama que obtemos é o da Figura 4.

$$K_\theta = 2.85 \quad , \quad \alpha = 0.53 \quad \leftrightarrow \quad K_Q = 1.51$$

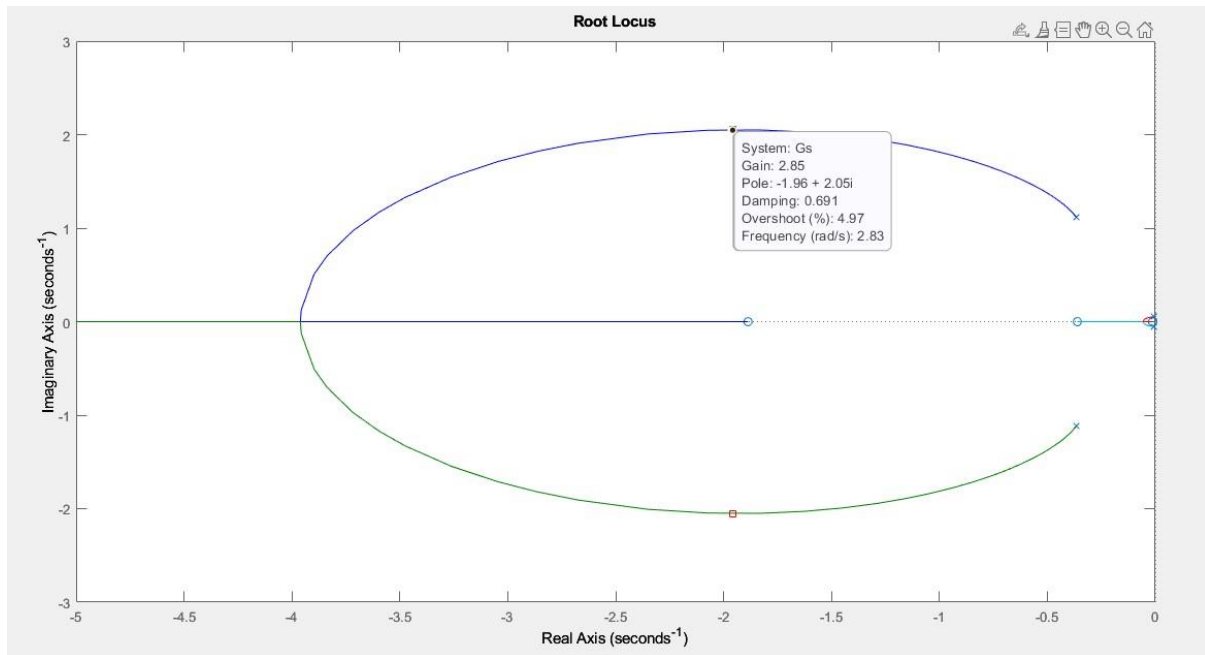


Figura 4 – Diagrama de lugar de raízes (anel aberto do sistema de controlo de inclinação)

b) Verificação do Controlador

Procedemos à verificação do controlador de inclinação analisando a resposta do sistema de controlo por retroação negativa a um *step*.

```
Us = tf([0.53 1], 1)
Ktheta = 2.85
Ts = feedback(Ktheta * Fs, Us, -1)

figure(4)
step(Ts)
```

Como vemos na Figura 5 b), a resposta ao *step* tem um tempo de subida de 0.9 segundos, como desejado. Na Figura 5 a), no entanto, podemos observar que a sobre-elevação ultrapassa ligeiramente os 5%, chegando aos 7%. Este não é, portanto, o caso que idealizávamos.

Decidimos, então, experimentar baixar os valores da especificação de tempo de subida e de sobre-elevação, tanto alternadamente como simultaneamente, e após várias tentativas concluímos que não é possível cumprir ambas as especificações com rigor.

Esta situação era expectável, uma vez que sabíamos previamente que aumentar o coeficiente de amortecimento – e consequentemente diminuir a sobre-elevação - afasta os polos do eixo real e, consequentemente, aumenta o tempo de resposta, o que é refletido no tempo de subida.

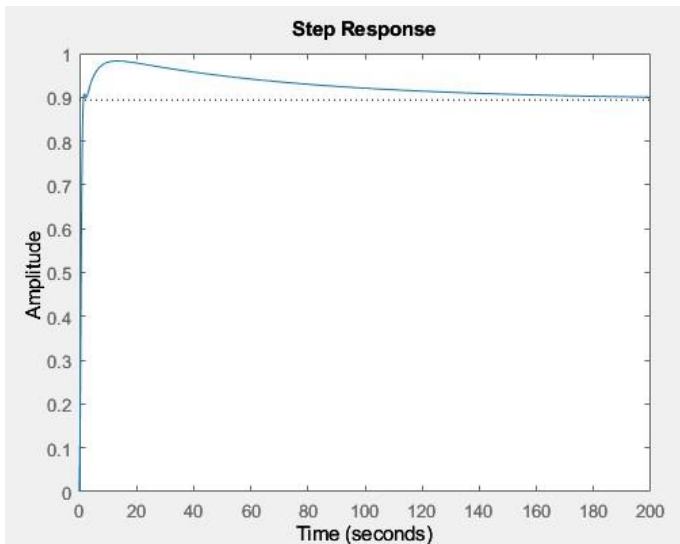


Figura 5 a) – Resposta do sistema de controlo de inclinação a um degrau unitário (análise da sobre-elevação)

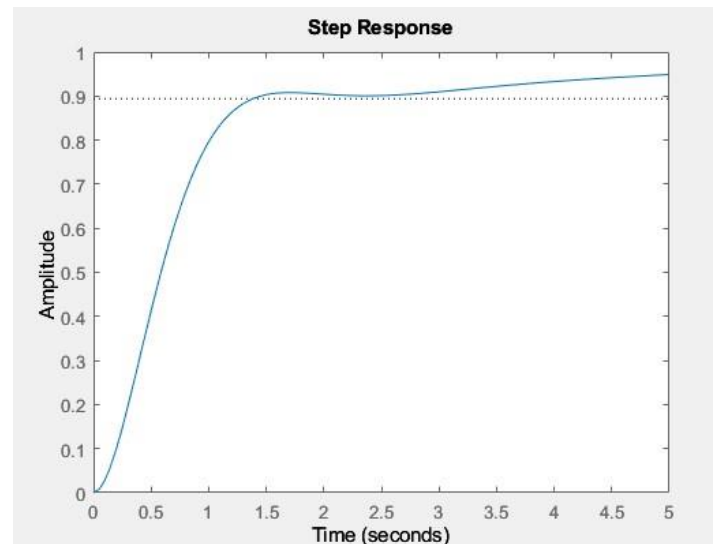


Figura 5 b) – Resposta do sistema de controlo de inclinação a um degrau unitário (análise do tempo de subida)

c) Testagem do Controlador no Simulador

Decidimos manter os valores para as especificações com os quais trabalhámos inicialmente, o que resultou nos seguintes valores de ganho:

$$K_{\theta} = 2.85 \text{ e } K_Q = 1.51$$

Inserindo estes valores no modo de **controlo de inclinação**, podemos testar o sistema de controlo no simulador, atuando manualmente no valor de referência do ângulo de inclinação.

Neste contexto, ainda não nos podemos basear no KPI como parâmetro único de testagem do sistema, uma vez que este controlador não atua sob a altitude. Podemos, no entanto, verificar que, dado um determinado valor de referência do ângulo de inclinação, o sistema fornece o valor de ângulo de inclinação de maneira rápida (reduzido tempo de subida), e praticamente sem qualquer sobre-elevação.

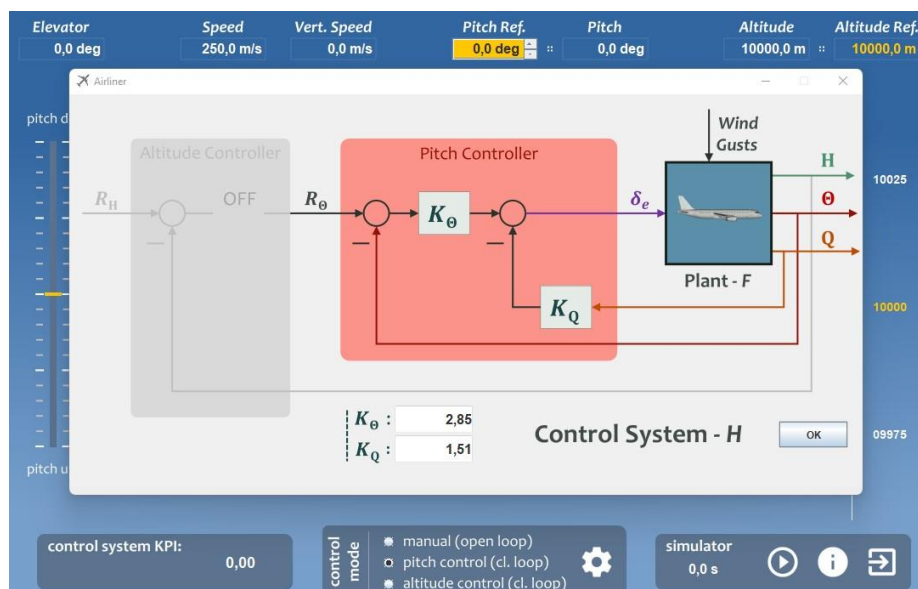


Figura 6 – Modo de controlo de inclinação

III. Desenvolvimento e Testagem do Controlador de Altitude

Nesta fase, e tendo em consideração o processo anteriormente efetuado, pretendemos recorrer ao diagrama do lugar de raízes para desenvolver um controlador, responsável por regular a altitude:

$$R_{\theta}(t) = K_H[R_H(t) - H(t)]$$

Nesta expressão, K_H é o ganho de altitude e R_H é o valor de referência de altitude, que é o sinal de entrada do sistema de controlo.

Neste caso, foi-nos dada a liberdade de escolher as nossas próprias especificações, de modo que nos limitamos a desenhar o diagrama de lugar de raízes e escolher um ganho, tendo em conta que um ganho alto resultará num comportamento mais oscilatório, e baixo numa resposta mais lenta.

a) *Design* do Controlador utilizando o Diagrama do Lugar de Raízes

1. Altitude derivada do sistema de controlo do ângulo de inclinação

Como explicado na secção I.d), obtém-se a altitude através do ângulo de inclinação aplicando à sua função de transferência o bloco integrador $\frac{250}{s}$, em série. Assim, juntamos o sistema controlador do ângulo de inclinação com este bloco com o seguinte código:

```
G2s = tf(250, [1 0])  
F2s = series(Ts, G2s)
```

2. Especificações e diagrama do lugar de raízes

Neste caso, foi-nos dada a liberdade de escolher as nossas próprias especificações, de modo que desenhamos o diagrama do lugar de raízes do sistema anterior em anel aberto e escolhemos um ganho para o qual o passo seguinte (3.) nos satisfaça, e tendo em consideração que um ganho alto resultará num comportamento mais oscilatório, e baixo numa resposta mais lenta. Ganho escolhido: $K_H = 0.00371$.

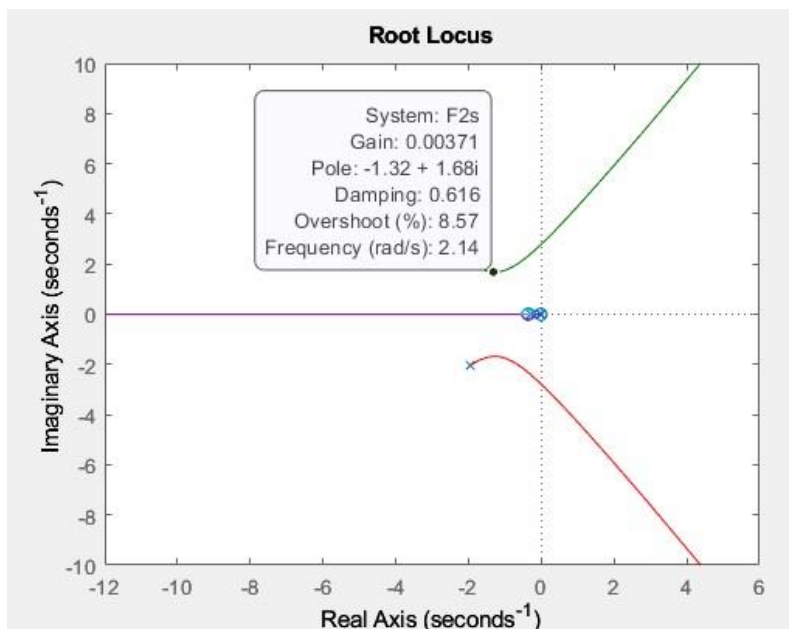


Figura 7 – Diagrama de lugar de raízes (anel aberto do sistema de controlo de altitude)

b) Verificação do Controlador

O valor de ganho acima mencionado foi escolhido mediante a sua contribuição para a resposta do sistema de controlo com retroação negativa ao *step*. Utilizando o seguinte código, analisámos as respostas para vários valores de ganho e concluímos que $K_H = 0.00371$ é aquele para o qual existe simultaneamente uma sobre-elevação pequena ($\sim 5\%$) e um tempo de subida relativamente curto ($< 2s$), o que se pode verificar na Figura 8.

```
Kh = 0.00371
Hs = feedback(Kh * F2s, 1, -1)

figure(5)
step(Hs)
```

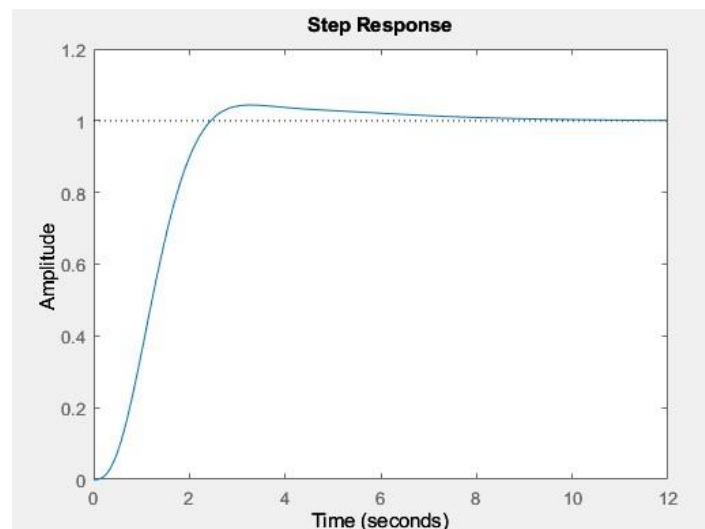


Figura 8 – Resposta do sistema de controlo de altitude a um degrau unitário (análise da sobre-elevação e do tempo de subida)

c) Testagem do Controlador no Simulador

Por fim, inserindo os valores de ganho no simulador:

$$K_{\theta} = 2.85, K_Q = 1.51 \text{ e } K_H = 0.00371$$

Podemos testar o sistema final de controlo de altitude, avaliando-o diretamente com o valor de *KPI* fornecido pela aplicação:

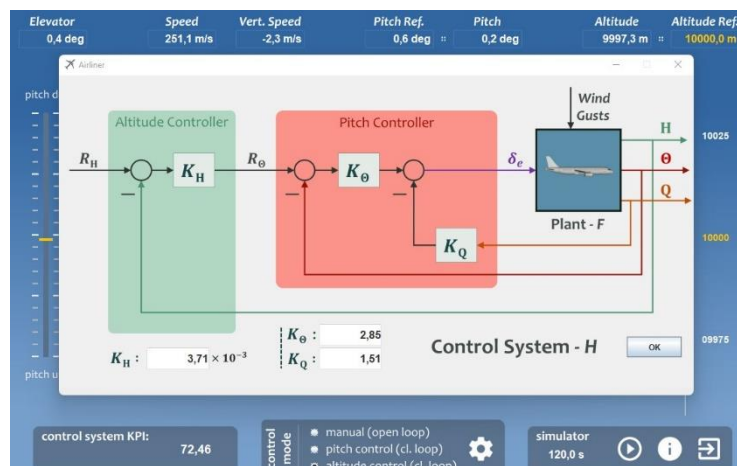


Figura 9 – Modo de controlo de altitude

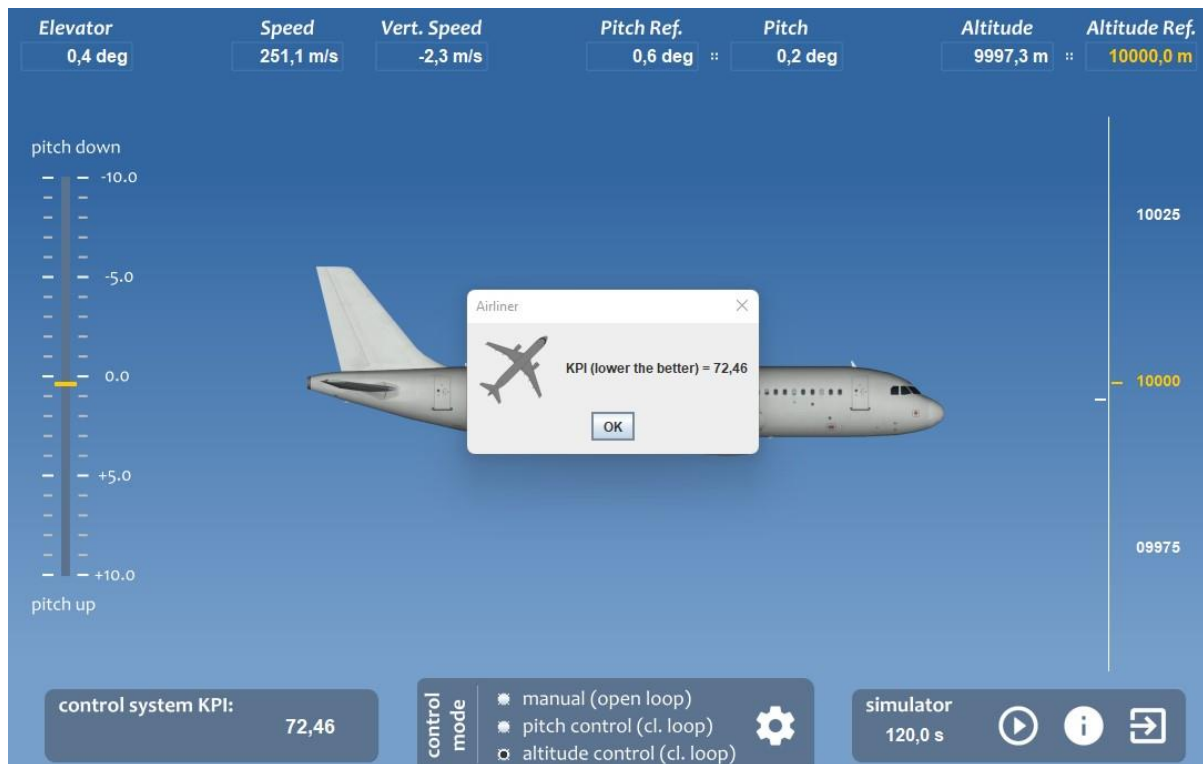


Figura 10 – *KPI* (Key Performance Indicator) do sistema de controlo

Através da Figura 10, que nos indica um *KPI* de 72.46, concluímos que o nosso controlador tem uma qualidade relativamente boa, sendo este valor não muito superior ao máximo que conseguimos obter em modo de controlo manual.

IV. Anexo – Código MatLab

No presente anexo expomos o código da ferramenta de cálculo MatLab, utilizado em todo o trabalho, devidamente organizado. O ficheiro onde foi desenvolvido foi aquele criado no sentido de comparar o funcionamento do sistema Airliner em modo manual no simulador com o funcionamento do nosso modelo.

```
close all
rad2deg = 180.0 / pi;
load y221108_122740.dat
x = y221108_122740;
clear y221108_122740;

% t - time
t = x(:,1);

% h - altitude [m]
h = x(:,2);

% theta - pitch angle [rad]
theta = x(:,3);

% q - pitch rate [rad/s]
q = x(:,4);

% u - elevator control input [rad]
u = x(:,5);

% r_theta - pitch angle reference [rad]
r_theta = x(:,6);

% r - altitude reference [m]
r_h = x(:,7);

% wg - vertical wind gust [m/s]
wg = x(:,8);

% kpiTotal - performance index Total
kpi = x(:,9);

clear x;

figure('Name','Altitude');
subplot(311);
plot(t, [h r_h]); title('Altitude [m]'); ylabel('h, r_h [m]');

subplot(312);
plot(t, [theta] * rad2deg); ylabel('\Theta, r_{\Theta} [deg]'); xlabel('t [s]');

subplot(313);
plot(t, [q] * rad2deg); ylabel('Q [deg/s]'); xlabel('t [s]');

figure('Name','Wind disturbance');
plot(t, wg); title('Wind gust speed [m/s]'); ylabel('wg [m/s]'); xlabel('t [s]');

figure('Name','Control action');
plot(t, u * rad2deg); title('Elevator position [deg]'); ylabel('\delta_e [deg]'); xlabel('t [s]');

figure('Name','Performance Index');
```

```

plot(t, kpi); title('KPI'); ylabel('kpi'); xlabel('t [s]');

% --- 1. MODELAÇÃO ---

b1 = 2.305148400378274 ;
b2 = 0.861103913823527 ;
b3 = 0.010626751575719 ;
a1 = 0.740551982652778 ;
a2 = 1.380681506199432 ;
a3 = 0.019931577735793 ;
a4 = 0.003595969889691 ;

Fs = tf( [ b1 b2 b3 ], [ 1 a1 a2 a3 a4]);

% --- 2. TESTAR A MODELAÇÃO ---

figure(5)
ysim = lsim(Fs, u, t);
plot(t, [theta ysim] * rad2deg); ylabel('Pitch Angle [deg]'); xlabel('t [s]');

% --- 3. PZ MAP para Estabilidade ---

figure(6)
pzmap(Fs);

% --- 4. PITCH CONTROLLER DESIGN AND TEST ---

% Construir Hs - Especificação:
s_spec = 5.0;
tr_spec = 0.9; %rise time
D = sqrt(log(s_spec/100)^2/(pi^2+log(s_spec/100)^2)) %sobre elevação
wn = 3.7 * D / tr_spec
Hs_spec = tf( wn^2, [1 2*D*wn wn^2] );

% Testar Especificação
figure(7)
step ( Hs_spec )

% Polos da Especificação
pole(Hs_spec)
figure(8)
pzmap ( Hs_spec ) % pole zero map

% Root Locus do anel aberto
Us = tf([0.53 1], 1) % (alpha.s + 1)
Gs = series(Fs,Us) % retroação (neste caso é equivalente a série)

figure(9)
rlocus(Gs)
hold on
plot( real( pole(Hs_spec) ), imag( pole(Hs_spec)), 'rs') %red squares
axis([-5 0 -3 3])
hold off

% Testar com Step

Ktheta = 2.85
Ts = feedback(Ktheta * Fs, Us, -1)

figure(10)
step(Ts)

```

```

axis([0 5 0 1])

% --- 5. ALTITUDE CONTROLLER DESIGN AND TEST ---

% Root Locus do anel aberto

G2s = tf(250, [1 0])
F2s = series(Ts, G2s)

figure(11)
rlocus(F2s) %escolher um ganho (alto - oscilatório; baixo - lento)

% Testar com Step

figure(12)
Kh = 0.00371
Hs = feedback(Kh * F2s, 1, -1)
step(Hs)

```