

```

1  function problema7p14
2  % Dinamica e Controle de Veiculos Espaciais
3  % Problema 7.14 da referencia Wie, B. Space Vehicle Dynamics and Control. 2 . ed., AIAA
4  % Education Series, Reston, VA: AIAA, 2008.
5  clc;close all;clear all;
6  %% Entrada de parametros
7  global J tipo k c1 c2 c3 a b vu tu
8  % Matriz de momentos de inercia
9  J=[1200 100 -200
10     100 2200 300
11     -200 300 3100]; % kg.m^2
12  % Condicao inicial
13  Q0=[0.5;0.5;0.5;-0.5];
14  w0=[0;0;0];
15  % Intervalo de simulacao
16  T=500; % s
17  %% Projeto do controle
18  tipo=input('Que tipo de matriz de ganhos pretende usar? 1, 2, 3, ou 4? ');
19  switch tipo
20      case 1
21          disp('K=k*I33, C=diag(c1,c2,c3) - Digite os valores: ');
22          k=input('k: ');c1=input('c1: ');c2=input('c2: ');c3=input('c3: ');
23      case 2
24          disp('K=(k/q4^3)*I33, C=diag(c1,c2,c3) - Digite os valores: ');
25          k=input('k: ');c1=input('c1: ');c2=input('c2: ');c3=input('c3: ');
26      case 3
27          disp('K=k*sgn(q4)*I33, C=diag(c1,c2,c3) - Digite os valores: ');
28          k=input('k: ');c1=input('c1: ');c2=input('c2: ');c3=input('c3: ');
29      case 4
30          disp('K=(a*diag(J)+b*I33)^-1, C=diag(c1,c2,c3) - Digite os valores: ');
31          a=input('a>0: ');b=input('b>0: ');c1=input('c1>0: ');c2=input('c2>0: ');c3=input('c3>0: ');
32      otherwise
33          disp('Digite um valor de tipo valido. ');
34  end
35
36  %% Integracao da dinamica
37  % Condicao inicial
38  x0=[w0;Q0];
39  % Vetores para guardar os controles e o tempo associado
40  vu=[0 0 0];
41  tu=0;
42  % As tolerancias são importantes, pois pequenos desvios prejudicam os resultados
43  opt=odeset('RelTol',1e-12,'AbsTol',1e-12,'MaxStep',1);
44  [t,x]=ode45(@dinamica_VE_quat,[0 T],x0,opt);
45  % Angulo principal
46  PHI=2*acos(x(:,7));
47  %% Graficos
48  figure
49  subplot(421);plot(t,x(:,1));grid;xlabel('t (s)');ylabel('\omega_x (rad/s)');axis tight
50  subplot(423);plot(t,x(:,2));grid;xlabel('t (s)');ylabel('\omega_y (rad/s)');axis tight
51  subplot(425);plot(t,x(:,3));grid;xlabel('t (s)');ylabel('\omega_z (rad/s)');axis tight
52  subplot(422);plot(t,x(:,4));grid;xlabel('t (s)');ylabel('q_1');axis tight

```

```

53 - subplot(424);plot(t,x(:,5));grid;xlabel('t (s)');ylabel('q_2');axis tight
54 - subplot(426);plot(t,x(:,6));grid;xlabel('t (s)');ylabel('q_3');axis tight
55 - subplot(428);plot(t,x(:,7));grid;xlabel('t (s)');ylabel('q_4');axis tight
56 - figure
57 - subplot(311);plot(tu,vu(:,1));grid;xlabel('t (s)');ylabel('u_x - N.m');axis tight
58 - subplot(312);plot(tu,vu(:,2));grid;xlabel('t (s)');ylabel('u_y - N.m');axis tight
59 - subplot(313);plot(tu,vu(:,3));grid;xlabel('t (s)');ylabel('u_z - N.m');axis tight
60 - figure
61 - subplot(221);
62 - plot(t,PHI*180/pi);grid;xlabel('t (s)');ylabel('angulo principal \Phi (graus)');axis tight
63 - subplot(222);plot(x(:,4),x(:,5));grid;xlabel('q_1');ylabel('q_2');axis tight
64 - subplot(223);plot(x(:,4),x(:,6));grid;xlabel('q_1');ylabel('q_3');axis tight
65 - subplot(224);plot(x(:,5),x(:,6));grid;xlabel('q_2');ylabel('q_3');axis tight
66
67 %% Estudo de angulos de Euler
68 % Converte de quaternion para os angulos de Euler na sequencia 123
69 % Tomar cuidado, usamos o quaternion com a parte real no elemento 4,
70 % mas o MATLAB usa no elemento 1
71 [the1,the2,the3]=quat2angle([x(:,7),x(:,4),x(:,5),x(:,6)],'XYZ');
72 % Calcula os angulos de nutacao theta, spin relativo psi e precessao phi
73 phi=the1+pi/2; % phi=theta1+pi/2
74 theta=the2; % theta=theta2
75 psi=the3-pi/2; % psi=theta3-pi/2
76 % Graficos
77 figure
78 subplot(231);plot(t,the1*180/pi);grid;xlabel('t (s)');ylabel('\theta_1 (graus)');axis tight

```

```

79 - subplot(232);plot(t,the2*180/pi);grid;xlabel('t (s)');ylabel('\theta_2 (graus)');axis tight
80 - subplot(233);plot(t,the3*180/pi);grid;xlabel('t (s)');ylabel('\theta_3 (graus)');axis tight
81 - subplot(234);plot(t,phi*180/pi);grid;xlabel('t (s)');ylabel('\phi (graus)');axis tight
82 - subplot(235);plot(t,theta*180/pi);grid;xlabel('t (s)');ylabel('\theta (graus)');axis tight
83 - subplot(236);plot(t,psi*180/pi);grid;xlabel('t (s)');ylabel('\psi (graus)');axis tight
84

```

```

85 %% Animacao 3D
86 % Insere os angulos na sequencia 313 para gerar a animacao
87 x(:,4:6)=[phi theta psi];
88 % Escala de tempo
89 et=10;
90 % Chama a animacao
91 animacao_3d(t,x,J,et);
92

```

```

93 - end
94

```

```

95 %% Dinamica de um veiculo espacial rigido com cinematica pelos quaternions

```

```

96 - function xp=dinamica_VE_quat(t,x)
97 - % Vetor de estado: x=[wx;wy;wz;q1;q2;q3;q4]
98 % wx, wy, wz: velocidades de rotaçao em torno dos eixos x, y e z do SRC
99 % q1, q2, q3, q4: Componentes do quaternion de atitude do referencial do
100 % corpo com respeito ao referencial inercial
101 %% Passagem de parametros por variaveis globais
102 - global J vu tu
103 %% Desmembra o vetor de estado
104 - w=[x(1);x(2);x(3)];

```

```

1105 - Q=[x(4);x(5);x(6);x(7)];
1106 %% Vetor de controle
1107 u=controle_real_quat(x);
1108 % Salva o historico do controle em uma variavel global
1109 vu=[vu;u'];
1110 tu=[tu;t];
1111 %% Equacoes de dinamica
1112 wp=J^(-1)*(u-skew(w)*J*w);
1113 %% Equacoes de cinematica de quaternion
1114 Qp=0.5*[-skew(w) w;-w' 0]*Q;
1115 %% Derivada do vetor de estado
1116 xp=[wp;Qp];
1117 end
1118
1119 %% Vetor de controle
1120 function u=controle_real_quat(x)
1121 % O controle objetiva regular o satellite para o equilibrio, ou seja, para
1122 % velocidade angular nula, com o referencial do corpo alinhado ao
1123 % referencial inercial. Assim, a referencia do quaternion eh o quaternion
1124 % identidade e o erro de rastreo eh o proprio quaternion medido.
1125 % A estrutura do controle eh generica, consistindo de um analogo de
1126 % controle proporcional derivativo:  $u = -K*q - C*w$  (" $q$ " eh a parte vetorial do
1127 % quaternion e " $w$ " a velocidade angular com respeito ao referencial
1128 % inercial).
1129 % De acordo com o conteudo da secao 7.3.1, sao 4 possibilidades sugeridas
1130 % para montar as matrizes de realimentacao K e C. Que sao identificadas

```

```

1311 % pela variavel "tipo"
1312 global tipo k c1 c2 c3 J a b
1313 % Desmembra a entrada
1314 w=[x(1);x(2);x(3)]; % Velocidade angular
1315 q=[x(4);x(5);x(6)]; % Parte vetorial do quaternion
1316 q4=x(7); % Parte escalar do quaternion
1317 % Escolhe a forma de calculo dos ganhos
1318 switch tipo
1319 case 1
1320 K=k*eye(3);
1321 C=diag([c1,c2,c3]);
1322 case 2
1323 K=(k/q4^3)*eye(3);
1324 C=diag([c1,c2,c3]);
1325 case 3
1326 K=k*sign(q4)*eye(3);
1327 C=diag([c1,c2,c3]);
1328 case 4
1329 % Usar somente os momentos de inercia no calculo dos controles
1330 K=(a*diag([J(1,1),J(2,2),J(3,3)])+b*eye(3))^(-1);
1331 C=diag([c1,c2,c3]);
1332 end
1333 % Lei de controle
1334 u=-K*q-C*w;
1335 end
1336 %% Matriz anti simetrica do produto vetorial

```

```
157 □ function S=skew(w)
158 -     S=[0 -w(3) w(2)
159 -         w(3) 0 -w(1)
160 -         -w(2) w(1) 0];
161 - end
```