

```

1  function [A,B,B2,ftarf,ftrg]=modelo_iss_cmg
2  % Dinamica e Controle de Veiculos Espaciais
3  % Secao 7.4 da referencia Wie, B. Space Vehicle Dynamics and Control. 2. ed., AIAA
4  % Education Series, Reston, VA: AIAA, 2008.
5  % Modelo da ISS com atuacao de atitude por CMG
6  % Saidas
7  % A,B,B2: matrizes da dinamica linearizada para a condicao de equilibrio
8  % nadir apontado em malha aberta (controle e perturbacoes nulos, matriz de inercia
9  % diagonal)
10 % ftarf, ftrg: matrizes de funcao de transferencia dos eixos de arfagem e
11 % rolamento/guinada, respectivamente.
12 clc;close all;clear all;
13 %% Entrada de parametros
14 global I I n
15 % Matriz de momentos de inercia
16 J=[50.28, -0.39, 0.16
17     -0.39, 10.80, 0.16
18     0.16, 0.16, 58.57]*10^6; % slug.ft^2
19 J=J*14.5939*0.3048^2;
20 % Movimento medio da orbita
21 n=0.0011; % rad/s

22 %% Equilibrio e linearizacao
23 % Matriz de momentos de inercia passada como parametro para as funcoes
24 I=J;
25 per=input('Deseja calcular o equilibrio para a parcela constante da perturbacao atuante? sim (1), nao (0): ');
26 if per
27     d=perturbacao_ISS(0);
28     [Xe,Ue,hp]=eqOrbCirc(d);
29     fprintf('Equilibrio para controle nulo e perturbacao d_x = %f N.m, d_y = %f N.m, d_z = %f N.m\n',...
30             d(1),d(2),d(3));
31     fprintf('Velocidades angulares: w_x=%f rad/s, w_y=%f rad/s, w_z=%f rad/s, \n',Xe(1),Xe(2),Xe(3));
32     fprintf('Angulos de atitude com respeito ao LVLH: phi=%f, theta=%f, psi=%f, \n',...
33             Xe(4)*180/pi,Xe(5)*180/pi,Xe(6)*180/pi);
34     fprintf('Derivada do momento angular do CMG: hp_x=%f N.m/s, hp_y=%f N.m/s, hp_z=%f N.m/s, \n',...
35             hp(1),hp(2),hp(3));
36 end
37 % Determinacao do equilibrio da atitude para perturbacao e controle nulos
38 d=[0;0;0];
39 [Xe,Ue,hp]=eqOrbCirc(d);
40 disp('***** Equilibrio para perturbacao e controle nulos *****');
41 fprintf('Velocidades angulares: w_x=%f rad/s, w_y=%f rad/s, w_z=%f rad/s, \n',Xe(1),Xe(2),Xe(3));
42 fprintf('Angulos de atitude com respeito ao LVLH: phi=%f, theta=%f, psi=%f, \n',...

```

```

43     Xe(4)*180/pi,Xe(5)*180/pi,Xe(6)*180/pi);
44     fprintf('Derivada do momento angular do CMG: hp_x=%f N.m/s, hp_y=%f N.m/s, hp_z=%f N.m/s, \n',...
45         hp(1),hp(2),hp(3));
46     % Linearizacao
47     [A,B,B2]=linearizar(Xe,Ue,d);
48     fprintf('\n Matriz A\n ');
49     fprintf('\t%0.2e\t%0.2e\t%0.2e\t%0.2e\t%0.2e\t%0.2e\t%0.2e\t%0.2e\n', A');
50     fprintf('\n Matriz B\n');
51     fprintf('\t%0.2e\t%0.2e\t%0.2e\n', B');
52     fprintf('\n Matriz B_2\n');
53     fprintf('\t%0.2e\t%0.2e\t%0.2e\n', B2');
54     %% Aproximacao de matriz de inercia diagonal
55     % Matriz de momentos de inercia passada como parametro para as funcoes
56     IJ=diag([J(1,1),J(2,2),J(3,3)]);
57     disp('***** Aproximacao de matriz de inercia diagonal *****');
58     per=input('Deseja calcular o equilibrio para a parcela constante da perturbacao atuante? sim (1), nao (0): ');
59     if per
60         d=perturbacao_ISS(0);
61         [Xe,Ue,hp]=eqOrbCirc(d);
62         fprintf('Equilibrio para controle nulo e perturbacao d_x = %f N.m, d_y = %f N.m, d_z = %f N.m\n',...
63             d(1),d(2),d(3));

64         fprintf('Velocidades angulares: w_x=%f rad/s, w_y=%f rad/s, w_z=%f rad/s, \n',Xe(1),Xe(2),Xe(3));
65         fprintf('Angulos de atitude com respeito ao LVLH: phi=%f, theta=%f, psi=%f, \n',...
66             Xe(4)*180/pi,Xe(5)*180/pi,Xe(6)*180/pi);
67         fprintf('Derivada do momento angular do CMG: hp_x=%f N.m/s, hp_y=%f N.m/s, hp_z=%f N.m/s, \n',...
68             hp(1),hp(2),hp(3));
69     end
70     % Determinacao do equilibrio da atitude para perturbacao e controle nulos
71     d=[0;0;0];
72     [Xe,Ue,hp]=eqOrbCirc(d);
73     disp('***** Equilibrio para perturbacao e controle nulos *****');
74     fprintf('Velocidades angulares: w_x=%f rad/s, w_y=%f rad/s, w_z=%f rad/s, \n',Xe(1),Xe(2),Xe(3));
75     fprintf('Angulos de atitude com respeito ao LVLH: phi=%f, theta=%f, psi=%f, \n',...
76         Xe(4)*180/pi,Xe(5)*180/pi,Xe(6)*180/pi);
77     fprintf('Derivada do momento angular do CMG: hp_x=%f N.m/s, hp_y=%f N.m/s, hp_z=%f N.m/s, \n',...
78         hp(1),hp(2),hp(3));
79     [A,B,B2]=linearizar(Xe,Ue,d);
80     fprintf('\n Matriz A\n ');
81     fprintf('\t%0.2e\t%0.2e\t%0.2e\t%0.2e\t%0.2e\t%0.2e\t%0.2e\t%0.2e\n', A');
82     fprintf('\n Matriz B\n');
83     fprintf('\t%0.2e\t%0.2e\t%0.2e\n', B');
84     fprintf('\n Matriz B_2\n');

85     fprintf('\t%0.2e\t%0.2e\t%0.2e\n', B2');
86     %% Estabilidade
87     damp(A)
88     %% Modelos de espaco de estado e funcoes de transferencia
89     % Eixo de arfagem
90     b=B(:,2); % Entrada u_y
91     C=[0 0 0 0 1 0 0 0 0;0 0 0 0 0 0 0 1 0];D=[0;0]; % Saidas theta e hy
92     ssarf=ss(A,b,C,D);
93     disp('*** Funcoes de transferencia do eixo de arfagem - saidas theta e hy, entrada uy ***');
94     zpk(ssarf)
95     ftarf=tf(ssarf);
96     % Eixos de rolamento e guinada
97     b=[B(:,1) B(:,3)]; % Entradas u_x e u_z
98     % Saidas phi, psi, hx e hz
99     C=[0 0 0 1 0 0 0 0;0 0 0 0 0 1 0 0;0 0 0 0 0 0 1 0;0 0 0 0 0 0 0 1];
100    D=[0 0 0 0 0 0 0 0];
101    ssrg=ss(A,b,C,D);
102    disp('*** Funcoes de transferencia dos eixos de rolamento/guinada - saidas phi, psi, hx e hz, entradas ux e uz ***');
103    zpk(ssrg)
104    ftrg=zpk(ssrg);
105    end

```

```

106
107 %% Dinamica de atitude de um veiculo espacial em orbita circular baixa
108 % com torque de gradiente gravitacional e atuador CMG idealizado (sem
109 % dinamica dos guimbais)
110 function xp=dinamica_VE_tgg_cmg(t,x,u,d)
111 % Vetor de estado: x=[wx;wy;wz;phi;theta;psi;hx;hy;hz]
112 % wx, wy, wz: velocidades de rotaçao inerciais em torno dos eixos x, y e z do SRC
113 % phi, theta, psi [rad]: Angulos de Euler (rolamento, arfagem e guinada) do
114 % SRC com respeito ao LVLH de acordo com a sequencia 321
115 % hx, hy, hz [N.m.s]: quantidade de movimento angular do CMG nos eixos do SRC
116 % u [N.m]: vetor de controle, torques nos eixos x, y e z do SCR aplicados
117 % pelo atuador
118 % d [N.m]: vetor de disturbio, torques nos eixos x, y e z do SCR aplicados
119 % por alguma causa externa
120 %%
121 % Passagem de parametros por variaveis globais
122 global II n
123 % Desmembra o vetor de estado
124 w=[x(1);x(2);x(3)];
125 phi=x(4);theta=x(5);psi=x(6);
126 h=[x(7);x(8);x(9)]; % Vetor quantidade de movimento angular do atuador

127 % Matriz de rotacao do SRC com respeito ao LVLH
128 C1=[1, 0, 0;0, cos(phi), sin(phi);0, -sin(phi), cos(phi)];
129 C2=[cos(theta), 0, -sin(theta);0, 1, 0;sin(theta), 0, cos(theta)];
130 C3=[cos(psi), sin(psi), 0;-sin(psi), cos(psi), 0;0, 0, 1];
131 C=C1*C2*C3;
132 % Velocidade relativa do SRC com respeito ao LVLH escrita no SRC
133 wlvh=C*[0;-n;0]; % Velocidade do LVLH escrita no SRC
134 wblvh=w-wlvh;
135 % Equacoes de cinematica de angulos de Euler 321
136 Angp=(1/cos(theta))*[cos(theta) sin(phi)*sin(theta) cos(phi)*sin(theta)
137 0 cos(phi)*cos(theta) -sin(phi)*cos(theta)
138 0 sin(phi) cos(phi)]*wblvh;
139 % Torque de gradiente gravitacional em orbita circular
140 Mg=TGG(C,n);
141 % Equacao de dinamica de rotacao do satelite. Equacao classica do corpo rigido,
142 % mais torque de gradiente gravitacional, acao de controle "u" aplicada pelo
143 % atuador e disturbio "d"
144 wp=II^(-1)*(-skew(w)*II*w+Mg-u+d);
145 % Dinamica do CMG ideal (ignorando o tempo de resposta do driver de controle
146 % e a resposta dos guimbais). Neste caso, se assemelha as rodas de reacao
147 hp=-skew(w)*h+u;

```

```

148 % Derivada do vetor de estado
149 xp=[wp;Angp;hp];
150 end
151 %% Perturbacao
152 function d=perturbacao_ISS(t)
153 global n
154 % Torque perturbativo atuante sobre a ISS, resultante da acao aerodinamica
155 d=[1+sin(n*t)+0.5*sin(2*n*t)
156     4+2*sin(n*t)+0.5*sin(2*n*t)
157     1+sin(n*t)+0.5*sin(2*n*t)]; % foot.pound
158 d=d*0.3048*4.4482216; % N.m
159 end
160 %% Torque de gradiente gravitacional
161 function Mg=TGG(C,n)
162 % Funcao para calculo de torque de gradiente gravitacional em orbita
163 % circular
164 % Entradas
165 % C: Matriz de rotacao do SRC com respeito ao LVLH
166 % n [rad/s]: movimento medio da orbita circular
167 %%
168 % Passagem de parametros por variaveis globais

169 global I
170 % Torque de gradiente gravitacional no SRC
171 Mg=3*n^2*cross(C(:,3),I*C(:,3));
172 end
173 %% Matriz anti simetrica do produto vetorial
174 function S=skew(w)
175 S=[0 -w(3) w(2)
176     w(3) 0 -w(1)
177     -w(2) w(1) 0];
178 end
179 %% Funcao de linearizacao do modelo
180 function [A,B,B2]=linearizar(Xe,Ue,De)
181 % Calcula as matrizes do modelo linear em torno de um ponto de equilibrio
182 % Entradas:
183 % Xe: vetor de estado no equilibrio
184 % Ue: vetor de controle no equilibrio
185 % De: vetor de disturbio no equilibrio
186 % Saidas:
187 % A: matriz de estado do modelo linear
188 % B: matriz de controle do modelo linear
189 % B2: matriz do disturbio do modelo linear

```

```

190 %%
191 % Incrementos para calcular diferenças finitas das variáveis de estado
192 dx=1e-3;inc=dx*eye(9,9);
193 % Inicializa a matriz A
194 A=zeros(9,9);
195 % Função de estado no equilíbrio
196 fe=dinamica_VE_tgg_cmg(0,Xe,Ue,De);
197 % Calcula a matriz "A" a partir de diferenças finitas das variáveis de estado
198 for j=1:9
199     f=dinamica_VE_tgg_cmg(0,Xe+inc(:,j),Ue,De);
200     A(:,j)=(f-fe)/dx;
201 end
202 % Incrementos para calcular as diferenças finitas das variáveis de controle
203 du=1e-2;inc=du*eye(3,3);
204 % Inicializa a matriz B
205 B=zeros(9,3);
206 % Calcula a matriz "B" a partir de diferenças finitas das variáveis de controle
207 for j=1:3
208     f=dinamica_VE_tgg_cmg(0,Xe,Ue+inc(:,j),De);
209     B(:,j)=(f-fe)/du;
210 end

211 % Incrementos para calcular as diferenças finitas das variáveis de distúrbio
212 dd=1e-2;inc=dd*eye(3,3);
213 % Inicializa matriz B2
214 B2=zeros(9,3);
215 % Calcula a matriz "B2" a partir de diferenças finitas das variáveis de distúrbio
216 for j=1:3
217     f=dinamica_VE_tgg_cmg(0,Xe,Ue,De+inc(:,j));
218     B2(:,j)=(f-fe)/dd;
219 end
220 end

221 %% Função para calcular o equilíbrio da atitude do satélite
222 function [Xe,Ue,hp]=eqOrbCirc(d)
223 % Calcula o equilíbrio da atitude do satélite em órbita baixa submetido a torque de
224 % gradiente gravitacional e distúrbio constante nos eixos do SRC.
225 % Entradas:
226 % d [N.m]: Distúrbio constante nos eixos do SRC
227 % Sidas:
228 % Xe: vetor de estado no equilíbrio
229 % Ue: vetor de controle no equilíbrio
230 % hp: derivada da quantidade de movimento angular do CMG
231 % Condições de cálculo de equilíbrio

```

```

232 % O controle no equilibrio eh zero - sem uso do atuador
233 % Sao consideradas as equacoes diferenciais do corpo somente, as do atuador
234 % sao ignoradas (variaveis ignoraveis)
235 %%
236 % Passagem e recebimento de parâmetros por variáveis globais
237 global D n
238 % Passagem de parâmetros para a função objetivo
239 D=d;
240 % Chute inicial para as incógnitas
241 % z=[wx wy wz phi theta psi]'
242 z0=[0 -n 0 0 0 0]'; % Solucao do caso sem disturbio
243 % Chamada da funcao objetivo
244 options=optimset('TolFun',1e-20,'MaxFunEvals',1e6,'MaxIter',1e5,'ToIX',1e-20);
245 [z, fval, flag]=fsolve(@obj_eq,z0,options);
246 disp('Valor final da função objetivo:');disp(fval);
247 disp('flag');disp(flag);
248 % Resultado
249 % X=[wx wy wz phi theta psi hx hy hz]'
250 Xe = [z(1) z(2) z(3) z(4) z(5) z(6) 0 0 0]'; % Valores nulos das variaveis ignoraveis
251 % U=[ux uy uz]'
252 Ue = [0 0 0]'; % O calculo do equilibrio eh feito sob a condicao de controle nulo

253 % Derivada da quantidade de movimento angular do CMG no equilibrio
254 Xp=dinamica_VE_tgg_cmg(0,Xe,Ue,D);
255 hp=Xp(7:9);
256 end
257 %% Função objetivo para cálculo do equilibrio
258 function f=obj_eq(z)
259 % Recebimento de parâmetros por variáveis globais
260 global D
261 % Montagem dos vetores de estado e controle a partir das incognitas,
262 % parametros fornecidos e condicoes de contorno
263 % z=[wx wy wz phi theta psi]'
264 % X=[wx wy wz phi theta psi hx hy hz]'
265 Xe = [z(1) z(2) z(3) z(4) z(5) z(6) 0 0 0]';
266 % U=[ux uy uz]'
267 Ue = [0 0 0]'; % O calculo do equilibrio eh feito sob a condicao de controle nulo
268 % Calcula a função de estado (derivadas das variaveis de estado)
269 Xp=dinamica_VE_tgg_cmg(0,Xe,Ue,D);
270 % Elimina as derivadas das variaveis ignoraveis
271 f=Xp(1:6);
272 end

```