

## Animação 3D da dinâmica de rotação

```
animacao_3d.m exemplo_axis_simetrico_com_torque.m exemplo_axis_simetrico_livre_torque.m +
1 function animacao_3d(t,x,l,et)
2 % Dinamica e controle de veiculos espaciais
3 % Funcao de animacao dos corpos com spin (aulas 4 e 5)
4 % Entradas
5 % t [s]: instantes de tempo da simulacao. Cada elemento do vetor esta
6 % associado a uma linha da matriz 'x'
7 % x: matriz contendo o vetor de estado nos instantes de simulacao
8 % x(i,:)= [wx wy wz phi theta psi]: linha i da matriz
9 % wx, wy, wz [rad/s]: velocidades angulares escritas no sistema do corpo
10 % phi, theta, psi [rad]: angulos de precessao, nutacao e spin relativo,
11 % associados a sequencia 313 de angulos de Euler
12 % l: vetor com os momentos de inercia principais
13 % l=[lxx lyy lzz] [kg.m^2]
14 % et: escala de tempo. (<1: mais lento, >1: mais rapido)
15
16 %% Calcula a orientacao dos vetores quantidade de movimento angular e velocidade angular

17 N=length(t);
18 phih=zeros(N,1);thetah=phih;psih=phih;
19 phiw=zeros(N,1);thetaw=phiw;psiw=phiw;
20 for i=1:N
21 C=angle2dcm(x(i,4),x(i,5),x(i,6),'ZXZ'); % Matriz de atitude, transforma do fixo para o corpo
22 wb=[x(i,1);x(i,2);x(i,3)]; % Velocidade angular no referencial do corpo
23 Hb=[l(1)*x(i,1);l(2)*x(i,2);l(3)*x(i,3)]; % Quantidade de movimento angular no referencial do corpo
24 w0=transpose(C)*wb; % Transformacao do referencial do corpo para o inercial
25 H0=transpose(C)*Hb; % Transformacao do referencial do corpo para o inercial
26 psiw(i)=atan2(w0(2),w0(1)); % Angulo de guinada da velocidade angular (rotacao z+)
27 thetaw(i)=-atan2(w0(3),sqrt(w0(1)^2+w0(2)^2)); % Angulo de elevacao da velocidade angular (rotacao y+)
28 psih(i)=atan2(H0(2),H0(1)); % Angulo de guinada da quantidade de movimento angular (rotacao z+)
29 thetah(i)=-atan2(H0(3),sqrt(H0(1)^2+H0(2)^2)); % Angulo de elevacao da quantidade de movimento angular
30 end

31 %% Gera a animação
32 % Converter os angulos de Euler da sequencia 313 para 321, que eh o padrao
33 % da funcao de animacao do aerospace toolbox
34 ang313=x(:,4:6);
35 ang321=conv_313_321(ang313);
36 % A funcao que realiza a conversao ordena o resultado da seguinte maneira:
37 % coluna 1 rotacao psi_z, coluna 2 rotacao theta_y e coluna 3 rotacao phi_x
38 % A funcao de animacao precisa na ordem phi_x, theta_y, psi_z:
39 rot=[ang321(:,3) ang321(:,2) ang321(:,1)];
40 % Montagem da matriz de dados da funcao de animacao. A primeira coluna eh o
41 % tempo, as proximas tres sao as posicoes, as ultimas tres sao os angulos
42 % de Euler na sequencia 321, primeiro phi_x, depois theta_y, ultimo psi_z
43 pos=zeros(size(rot)); % Define posicao fixa e igual a zero
44 dados_corpo=[t pos rot];
45 % Dados para animacao da linha dos nodos (movimento de precessao)
46 rot=[zeros(N,2) x(:,4)]; % Angulo de precessao phi
```

```

47 - dados_set1=[t pos rot];
48 - % Dados para animacao do vetor quantidade de movimento angular
49 - rot=[phi h th eta h psi h];
50 - dados_set2=[t pos rot];
51 - % Dados para animacao do vetor velocidade angular
52 - rot=[phi w th eta w psi w];
53 - dados_set3=[t pos rot];
54 - %% No caso estudado, o eixo z aponta para cima.
55 - % No entanto, no toolbox de animacao do MATLAB, o eixo z aponta para
56 - % baixo. Uma sequencia de rotações que transforma dos eixos do MATLAB
57 - % para os eixos do estudo consiste em uma rotação de pi rad em torno de y.
58 - % O quaternion associado (parte escalar no primeiro elemento) eh
59 - Qrot=[0 0 1 0];
60 - % Chamada da funcao de animacao
61 - anima_1corpo_3setas_1ref(Qrot,dados_corpo,dados_set1,dados_set2,dados_set3,et);

62 - end
63 - %% Funcao para converter os angulos de Euler da sequencia 313 para 321
64 - function ang321=conv_313_321(ang313)
65 - % Entrada
66 - % ang313: matriz com N linhas e 3 colunas. Cada linha representa um
67 - % instante de tempo. Cada coluna tem os angulos na ordem phi_z, theta_x,
68 - % psi_z (da primeira para a ultima) no padrao de rotacao 313
69 - % Saida
70 - % ang321: matriz com N linhas e 3 colunas. Cada linha representa um
71 - % instante de tempo, cada coluna tem os angulos na ordem psi_z, theta_y,
72 - % phi_x (da primeira para a ultima) no padrao de rotacao 321.
73 - %% Desmembramento da entrada
74 - phi_z=ang313(:,1); % Primeira rotacao da sequencia 313
75 - theta_x=ang313(:,2); % Segunda rotacao da sequencia 313
76 - psi_z=ang313(:,3); % Terceira rotacao da sequencia 313
77 - %% Chamada da funcao que converte para quaternion

78 - Q=angle2quat(phi_z, theta_x, psi_z, 'ZXZ');
79 - %% Chamada da funcao que converte de quaternion para angulos de Euler
80 - ang321=zeros(size(ang313));
81 - [ang321(:,1),ang321(:,2),ang321(:,3)]= quat2angle(Q);
82 - end
83 - function anima_1corpo_3setas_1ref(Qrot,dados_corpo,dados_set1,dados_set2,dados_set3,esc_tempo)
84 - % Entradas
85 - % Qrot: quaternion de rotacao que gira a animacao do sistema
86 - % eixos do MATLAB para o de interesse
87 - % dados_corpo, dados_set1, dados_set2, dados_set3: 'Array6DoF' A double-precision array in
88 - % n rows and 7 columns for 6-DoF data: time x y z phi theta psi.
89 - % corpo: inteiro para identificar o corpo desejado.
90 - % Lista de corpos padrão em:
91 - % https://www.mathworks.com/help/aerotbx/ug/ac3d-files-and-thumbnails-overview.html
92 - % esc_tempo: variavel que dilata ou comprime o tempo de animacao
93 - % esc_tempo=1: tempo real e simulado iguais

```

```

94 % esc_tempo>1: tempo real mais rapido que o simulado
95 % esc_tempo<1: tempo real mais lento que o simulado
96
97 % Cria o objeto
98 h = Aero.Animation;
99 % The combination of FramesPerSecond and TimeScaling property
.00 % determine the time step of the simulation.
.01 h.FramesPerSecond = 10;
.02 h.TimeScaling = esc_tempo;
.03 % Cria os objetos de desenho 3D
.04 h.createBody('body_xyzisrgb_up.ac','Ac3d');
.05 h.createBody('body_xyzisgray_up.ac','Ac3d');
.06 h.createBody('seta_cinza.ac','Ac3d');
.07 h.createBody('seta_laranja.ac','Ac3d');
.08 h.createBody('seta_verde.ac','Ac3d');
.09
110 h.Bodies{1}.TimeseriesSourceType='Array6DoF';
111 dados_corpo=gira_dados(dados_corpo,Qrot);
112 h.Bodies{1}.TimeSeriesSource = dados_corpo;
113 h.Bodies{2}.TimeseriesSourceType='Array6DoF';
114 dados=dados_corpo;
115 dados(:,2:7)=zeros(size(dados_corpo(:,2:7)));
116 dados=gira_dados(dados,Qrot);
117 h.Bodies{2}.TimeSeriesSource = dados;
118 h.Bodies{3}.TimeseriesSourceType='Array6DoF';
119 dados_set1=gira_dados(dados_set1,Qrot);
120 h.Bodies{3}.TimeSeriesSource = dados_set1;
121 h.Bodies{4}.TimeseriesSourceType='Array6DoF';
122 dados_set2=gira_dados(dados_set2,Qrot);
123 h.Bodies{4}.TimeSeriesSource = dados_set2;
124 h.Bodies{5}.TimeseriesSourceType='Array6DoF';
125 dados_set3=gira_dados(dados_set3,Qrot);
126 h.Bodies{5}.TimeSeriesSource = dados_set3;
127 % Realiza a animacao
128 h.play();
129 end
130 %% Funcao para girar os dados
131 function d_out=gira_dados(d_in,Qrot)
132 % Entradas
133 % dados: matriz de dados no mesmo formato de entrada da funcao principal
134 % Quaternion de rotacao do sistema de referencia do MATLAB para o desejado
135 % Saida: dados rotacionados
136 %% Calculos
137 n=length(d_in(:,1)); % Numero de instantes de tempo
138 C=quat2dcm(Qrot); % Matriz de rotacao
139 d_out=zeros(size(d_in));
140 d_out(:,1)=d_in(:,1);
141 for i=1:n
142 d_out(i,2:4)=(C*d_in(i,2:4))';
143 [d_out(i,7),d_out(i,6),d_out(i,5)]=dcm2angle(angle2dcm(d_in(i,7),d_in(i,6),d_in(i,5))*C);
144 end
145 end

```

Chamada no programa de simulação do movimento livre de torque da aula 5

```
66 - subplot(235);
67 - plot(t,x(:,5)*180/pi);grid;xlabel('t (s)');ylabel('\theta [^\circ]');axis tight
68 - subplot(236);
69 - plot(t,x(:,6)*180/pi);grid;xlabel('t (s)');ylabel('\psi [^\circ]');axis tight
70 - %% Animacao 3D
71 - % Momentos de inercia para calcular a animacao da quantidade de movimento
72 - % angular
73 - I=[J J lzz];
74 - % Escala de tempo
75 - et=0.1;
76 - % Chama a animacao
77 - animacao_3d(t,x,I,et);
78 - end
```

Chamada no programa de simulação do movimento com torque aplicado da aula 5

```
74 - disp('Frequencia da nutacao (rad/s):');
75 - disp(wn);
76 -
77 - %% Animacao 3D
78 - % Momentos de inercia para calcular a animacao da quantidade de movimento
79 - % angular
80 - I=[J J lzz];
81 - % Insere os angulos na sequencia 313 para gerar a animacao
82 - x(:,4:6)=[phi theta psi];
83 - % Escala de tempo
84 - et=0.1;
85 - % Chama a animacao
86 - animacao_3d(t,x,I,et);
87 -
88 - end
```