

## Exemplo de solução das equações de cinemática de atitude e conversão entre elas.

```

exemplosCinematicaRotacao.m  matpEixoAng.m  cinematicaEuler321.m  cinematicaEuler313.m  cinematicaMCD.m  cinematicaQuatern
1  %% Exemplos de cinematica de rotacao
2  %% Utiliza-se a funcao ode45 para integrar a cinematica de rotacao para os
3  %% casos: angulos de Euler 321, angulos de Euler 313, matriz de cossenos
4  %% diretores, quaternio, parametros de Rodrigues modificados.
5  %% Aplica-se a mesma velocidade angular (definida em uma funcao separada) e
6  %% mesmas condicoes iniciais em todos os exemplos.
7  clc
8  %% Condiicoes iniciais definidas para que todas as parametrizacoes
9  %% representem a mesma situacao
10 alpha10=0*pi/180;alpha20=45*pi/180;alpha30=0*pi/180;    % Angulos de Euler 313
11 Euler3130=[alpha10;alpha20;alpha30];
12 C0=Euler313pMatriz(Euler3130);    % MCD
13 E3210=matrizpEuler321(C0); % Angulos de Euler 321
14 [e,Phi]=matpEixoAng(C0);    % Eixo de Euler e angulo principal
15 Q0=[e*sin(Phi/2)
16      cos(Phi/2)];    % Quaternio
17 p0=Q0(1:3)/(1+Q0(4));    % Parametros de Rodrigues modificados
18 %% Tempo das simulacoes
19 % Definiu-se a velocidade angular como uma funcao vetorial periodica de
20 % periodo 4 segundos. Define-se o tempo para simular 2 periodos
21 T=8;
22
23 %% Resolucao da equacao de cinematica dos angulos de Euler 321
24 opt=odeset('RelTol',1e-4,'AbsTol',1e-8,'MaxStep',1e-2);
25 [t1, E321]=ode45('cinematicaEuler321',[0 T],E3210,opt);
26 % Velocidade angular usada na simulacao
27 N=length(t1);
28 omega=zeros(N,3);
29 for i=1:N
30     omega(i,:)=velocidadeAngular(t1(i));
31 end
32 figure(1);
33 subplot(211);plot(t1,omega(:,1),t1,omega(:,2),t1,omega(:,3));grid;
34 xlabel('t [s]');ylabel('\omega [rad/s]');legend('\omega_x','\omega_y','\omega_z');
35 subplot(212);plot(t1,E321(:,1)*180/pi,t1,E321(:,2)*180/pi,t1,E321(:,3)*180/pi);
36 grid;xlabel('t [s]');ylabel('Ang. Euler 321 [°]');legend('\phi','\theta','\psi');
37
38 %% Resolucao da equacao de cinematica dos angulos de Euler 313
39 [t2, E313]=ode45('cinematicaEuler313',[0 T],Euler3130,opt);
40 % Converte para angulos de Euler 321 para comparar as respostas
41 N=length(t2);
42 E321_E313=zeros(N,3);
43 for i=1:N
44     C=Euler313pMatriz(E313(i,:));    % Converte de Euler 313 para matriz
45     E321_E313(i,:)=matrizpEuler321(C)';    % Converte de matriz para Euler 321
46 end
47 figure(2);
48 subplot(211);plot(t1,omega(:,1),t1,omega(:,2),t1,omega(:,3));grid;
49 xlabel('t [s]');ylabel('\omega [rad/s]');legend('\omega_x','\omega_y','\omega_z');
50 subplot(212);plot(t2,E313(:,1)*180/pi,t2,E313(:,2)*180/pi,t2,E313(:,3)*180/pi);
51 grid;xlabel('t [s]');ylabel('Ang. Euler 313 [°]');legend('\alpha_1','\alpha_2','\alpha_3');
52
53 %% Resolucao da equacao de cinematica da matriz dos cossenos diretores
54 % Monta um vetor
55 xm0=[C0(1,1);C0(1,2);C0(1,3);C0(2,1);C0(2,2);C0(2,3);C0(3,1);C0(3,2);C0(3,3)];
56 [t3,xm]=ode45('cinematicaMCD',[0 T],xm0,opt);
57 % Converte para angulos de Euler 321 para comparar as respostas
58 N=length(t3);
59 E321_mcd=zeros(N,3);
60 for i=1:N
61     C=[xm(i,1) xm(i,2) xm(i,3)
62        xm(i,4) xm(i,5) xm(i,6)
63        xm(i,7) xm(i,8) xm(i,9)];    % Matriz de cossenos diretores
64     E321_mcd(i,:)=matrizpEuler321(C)';    % Converte de matriz para Euler 321
65 end
66 figure(3);
67 subplot(211);plot(t1,omega(:,1),t1,omega(:,2),t1,omega(:,3));grid;
68 xlabel('t [s]');ylabel('\omega [rad/s]');legend('\omega_x','\omega_y','\omega_z');
69 subplot(212);
70 plot(t3,xm(:,1),t3,xm(:,2),t3,xm(:,3),t3,xm(:,4),t3,xm(:,5),t3,xm(:,6),t3,xm(:,7),t3,xm(:,8),t3,xm(:,9));
71 grid;
72 xlabel('t [s]');ylabel('MCD [-]');legend('c_11','c_12','c_13','c_21','c_22','c_23','c_31','c_32','c_33');

```

```

73
74 %% Resolucao da equacao de cinematica do quaternio
75 [t4, quat]=ode45('cinematicaQuaternio',[0 T],Q0,opt);
76 % Converte para angulos de Euler 321 para comparar as respostas
77 N=length(t4);
78 E321_quat=zeros(N,3);
79 for i=1:N
80     C=quatpMatriz(quat(i,:)); % Converte de quaternio para matriz
81     E321_quat(i,:)=matrizpEuler321(C); % Converte de matriz para Euler 321
82 end
83 figure(4);
84 subplot(211);plot(t1,omega(:,1),t1,omega(:,2),t1,omega(:,3));grid;
85 xlabel('t [s]');ylabel('\omega [rad/s]');legend('\omega_x','\omega_y','\omega_z');
86 subplot(212);plot(t4,quat(:,1),t4,quat(:,2),t4,quat(:,3),t4,quat(:,4));
87 grid;xlabel('t [s]');ylabel('quaternio [-]');legend('Q_1','Q_2','Q_3','Q_4');
88
89 %% Resolucao da equacao de cinematica dos parametros de Rodrigues modificados
90 [t5, prm]=ode45('cinematicaPRM',[0 T],p0,opt);
91 % Converte para angulos de Euler 321 para comparar as respostas
92 N=length(t5);
93 E321_prm=zeros(N,3);
94 for i=1:N
95     C=prmpMatriz(prm(i,:)); % Converte de PRM para matriz
96     E321_prm(i,:)=matrizpEuler321(C); % Converte de matriz para Euler 321
97 end
98 figure(5);
99 subplot(211);plot(t1,omega(:,1),t1,omega(:,2),t1,omega(:,3));grid;
100 xlabel('t [s]');ylabel('\omega [rad/s]');legend('\omega_x','\omega_y','\omega_z');
101 subplot(212);plot(t5,prm(:,1),t5,prm(:,2),t5,prm(:,3));
102 grid;xlabel('t [s]');ylabel('prm [-]');legend('p_1','p_2','p_3');
103
104 %% Compara os resultados
105 figure(6);
106 subplot(311);plot(t1,E321(:,1),t2,E321_E313(:,1),t3,E321_mcd(:,1),t4,E321_quat(:,1),t5,E321_prm(:,1));
107 grid;xlabel('t [s]');ylabel('\phi [rad]');legend('Euler 321','Euler 313','MCD','quaternio','PRM');
108 subplot(312);plot(t1,E321(:,2),t2,E321_E313(:,2),t3,E321_mcd(:,2),t4,E321_quat(:,2),t5,E321_prm(:,2));
109 grid;xlabel('t [s]');ylabel('\theta [rad]');legend('Euler 321','Euler 313','MCD','quaternio','PRM');
110 subplot(313);plot(t1,E321(:,3),t2,E321_E313(:,3),t3,E321_mcd(:,3),t4,E321_quat(:,3),t5,E321_prm(:,3));
111 grid;xlabel('t [s]');ylabel('\psi [rad]');legend('Euler 321','Euler 313','MCD','quaternio','PRM');

```

```

exemplosCinematicaRotacao.m  matpEixoAng.m  cinematicaEuler321.m  cinematicaEuler313.m  cinematicaMCD.m
1  function [e,Phi]=matpEixoAng(C)
2  % Funcao que converte de matriz para eixo de euler e angulo principal
3  % Entrada:
4  % C: matriz de cossenos diretores
5  % Saidas
6  % e: eixo de Euler
7  % Phi: [rad] angulo principal
8  %%
9  % Autovalores e autovetores
10 [V,D] = eig(C);
11 % Autovalores
12 lamb=[D(1,1) D(2,2) D(3,3)];
13 % Separa o autovalor real e os imaginarios
14 for i=1:3
15     if imag(lamb(i))==0
16         iR=i;
17     end
18     if imag(lamb(i))>0
19         lambC=lamb(i);
20     end
21 end
22 % Eixo de Euler
23 e=V(:,iR)/norm(V(:,iR));
24 % Angulo principal
25 Phi=phase(lambC);
26 end

```

```

exemplosCinematicaRotacao.m  matpEixoAng.m  cinematicaEuler321.m  cinematicaEuler313.m  cinematicaMCD.m  cine
1  function xp=cinematicaEuler321(t,x)
2  % Funcao de cinematica dos angulos de Euler 321 no formato dos solvers do
3  % MATLAB
4  % Entradas
5  % t: tempo - obrigatorio na sintaxe dos solvers do MATLAB. Necessario para
6  % a chamada interna da funcao que retorna a velocidade angular
7  % x: vetor de estado.
8  % x=[phi theta psi]: angulos da sequencia de rotacoes
9  % C3(psi)-> C2(theta)-> C1(phi)
10 % Saida
11 % xp: derivada do vetor de estado
12 %%
13 % Angulos de Euler
14 phi=x(1);theta=x(2);%psi=x(3);
15 % Velocidade angular
16 w=velocidadeAngular(t);
17 % Derivadas
18 xp=(1/cos(theta))*[cos(theta) sin(phi)*sin(theta) cos(phi)*sin(theta)
19                    0 cos(phi)*cos(theta) -sin(phi)*cos(theta)
20                    0 sin(phi) cos(phi)]*w;
21 end

```

```

exemplosCinematicaRotacao.m x matpEixoAng.m x cinematicaEuler321.m x cinematicaEuler313.m x cinematicaMCD.m x cinematicaQuaternio.r
1 function xp=cinematicaEuler313(t,x)
2 % Funcao de cinematica dos angulos de Euler 313 no formato dos solvers do
3 % MATLAB
4 % Entradas
5 % t: tempo - obrigatorio na sintaxe dos solvers do MATLAB. Necessario para
6 % a chamada interna da funcao que retorna a velocidade angular
7 % x: vetor de estado.
8 % x=[alpha1 alpha2 alpha3]: angulos da sequencia de rotacoes
9 % C3(alpha1)-> C1(alpha2)-> C3(alpha3)
10 % Saida
11 % xp: derivada do vetor de estado
12 % Angulos de Euler
13 alpha1=x(1);alpha2=x(2);alpha3=x(3);
14 % Velocidade angular
15 w=velocidadeAngular(t);
16 % Derivadas
17 xp=(1/sin(alpha2))*[sin(alpha3) cos(alpha3) 0
18 sin(alpha2)*cos(alpha3) -sin(alpha2)*sin(alpha3) 0
19 -cos(alpha2)*sin(alpha3) -cos(alpha2)*cos(alpha3) sin(alpha2)]*w;
20 end

```

```

exemplosCinematicaRotacao.m x matpEixoAng.m x cinematicaEuler321.m x cinematicaEuler313.m x cinematicaMCD.m x cinematicaQuaternio.r
1 function xp=cinematicaMCD(t,x)
2 % Funcao de cinematica da matriz de cossenos diretores no formato dos solvers do
3 % MATLAB
4 % Entradas
5 % t: tempo - obrigatorio na sintaxe dos solvers do MATLAB. Necessario para
6 % a chamada interna da funcao que retorna a velocidade angular
7 % x: vetor de estado contendo os elementos da matriz
8 % x=[c11 c12 c13 c21 c22 c23 c31 c32 c33]
9 % Saida
10 % xp: derivada do vetor de estado
11 %%
12 % Matriz de cossenos diretores
13 C=[x(1) x(2) x(3)
14 x(4) x(5) x(6)
15 x(7) x(8) x(9)];
16 % Velocidade angular
17 w=velocidadeAngular(t);
18 % Matriz antisimettrica da velocidade angular
19 Sw=[0 -w(3) w(2)
20 w(3) 0 -w(1)
21 -w(2) w(1) 0];
22 % Derivada da matriz de cossenos diretores
23 Cp=-Sw*C;
24 % Monta o vetor de saida
25 xp=[Cp(1,1)
26 Cp(1,2)
27 Cp(1,3)
28 Cp(2,1)
29 Cp(2,2)
30 Cp(2,3)
31 Cp(3,1)
32 Cp(3,2)
33 Cp(3,3)];
34 end

```

```

1  function xp=cinematicaQuaternio(t,x)
2  % Funcao de cinematica dos quaternios no formato dos solvers do MATLAB
3  % Entradas
4  % t: tempo - obrigatorio na sintaxe dos solvers do MATLAB. Necessario para
5  % a chamada interna da funcao que retorna a velocidade angular
6  % x: vetor de estado.
7  % x=[q1; q2; q3; q4]: q4 eh a parte real. Assume vetor coluna
8  % Saida
9  % xp: derivada do vetor de estado
10 % Quaternio
11 Q=x;
12 % Velocidade angular
13 w=velocidadeAngular(t);
14 % Matriz antisimettrica da velocidade angular
15 Sw=[0      -w(3) w(2)
16      w(3)  0      -w(1)
17      -w(2) w(1)  0];
18 % Matriz da velocidade angular na derivada do quaternion
19 OM=[-Sw w
20      -w' 0];
21 % Derivada do quaternion
22 xp=0.5*OM*Q;
23 end

```

```

1  function xp=cinematicaPRM(t,x)
2  % Funcao de cinematica dos parametros de Rodrigues modificados no formato
3  % dos solvers do MATLAB
4  % Entradas
5  % t: tempo - obrigatorio na sintaxe dos solvers do MATLAB. Necessario para
6  % a chamada interna da funcao que retorna a velocidade angular
7  % x: vetor de estado.
8  % x=[p1; p2; p3]. Assume vetor coluna.
9  % Saida
10 % xp: derivada do vetor de estado
11 %%
12 % Parametros de Rodrigues modificados
13 p=x;
14 % Velocidade angular
15 w=velocidadeAngular(t);
16 % Matriz antisimettrica dos PRMs
17 Sp=[0      -p(3) p(2)
18      p(3)  0      -p(1)
19      -p(2) p(1)  0];
20 % Derivadas dos PRMs
21 xp=0.5*(Sp +p*p'+0.5*(1-p'*p)*eye(3,3))*w;
22 end

```

```

cinematicaPRM.m x matrizpEuler321.m x Euler313pMatriz.m x quatpMatriz.m x velocidadeAngular.m x prmpMatriz.m x
1 function E321=matrizpEuler321(C)
2 % Funcao que converte de matriz para angulos de Euler da sequencia 321
3 % Entrada:
4 % C: matriz de cossenos diretores
5 % Saida:
6 % E321=[phi theta psi]: angulos da sequencia de rotacoes
7 % C3(psi)-> C2(theta)-> C1(phi)
8 %%
9 % Conversao para os angulos de Euler 321
10 phi=atan2(C(2,3),C(3,3));
11 theta=-asin(C(1,3));
12 psi=atan2(C(1,2),C(1,1));
13 % Saida
14 E321=[phi; theta; psi];
15 end

```

```

cinematicaPRM.m x matrizpEuler321.m x Euler313pMatriz.m x quatpMatriz.m x velocidadeAngular.m x prmpMatriz.m x
1 function C=Euler313pMatriz(E313)
2 % Funcao que converte da sequencia de angulos de Euler 313 para matriz de
3 % cossenos diretores
4 % Entrada:
5 % E313=[alpha1 alpha2 alpha3]: angulos da sequencia de rotacoes
6 % C3(alpha1)-> C1(alpha2)-> C3(alpha3)
7 % Saida:
8 % C: Matriz de cossenos diretores
9 %%
10 % Angulos de Euler
11 alpha1=E313(1);alpha2=E313(2);alpha3=E313(3);
12 % Matrices de rotacao elementares
13 C3_a1=[cos(alpha1) sin(alpha1) 0
14        -sin(alpha1) cos(alpha1) 0
15         0 0 1];
16
17 C1_a2=[1 0 0
18        0 cos(alpha2) sin(alpha2)
19        0 -sin(alpha2) cos(alpha2)];
20
21 C3_a3=[cos(alpha3) sin(alpha3) 0
22        -sin(alpha3) cos(alpha3) 0
23         0 0 1];
24 % Matriz de cossenos diretores
25 C=C3_a3*C1_a2*C3_a1;
26 end

```

```

1 function C=quatpMatriz(Q)
2 % Funcao que converte de quaternion para matriz de cossenos diretores
3 % Entrada
4 % Q: quaternion. Q=[q1; q2; q3; q4], q4 eh a parte real. Assume vetor
5 % coluna.
6 % Saida
7 % C: matriz de cossenos diretores
8 %%
9 % Parte vetorial do quaternion
10 q=Q(1:3);
11 % Parte escalar
12 q4=Q(4);
13 % Matriz antisimetrica da parte vetorial do quaternion
14 Sq=[0      -q(3) q(2)
15      q(3)  0      -q(1)
16      -q(2) q(1)  0];
17 % Matriz de cossenos diretores
18 C=(q4^2-q'*q)*eye(3,3)+2*(q*q')-2*q4*Sq;
19 end

```

```

1 function C=prmpMatriz(p)
2 % Funcao que converte de parametros de Rodrigues modificados para matriz de
3 % cossenos diretores
4 % Entrada
5 % p: vetor de parametros de Rodrigues modificados.
6 % p=[p1; p2; p3]. Assume vetor coluna.
7 % Saida
8 % C: matriz de cossenos diretores
9 %%
10 % Matriz antisimetrica dos parametros de Rodrigues modificados
11 Sp=[0      -p(3) p(2)
12      p(3)  0      -p(1)
13      -p(2) p(1)  0];
14 % Matriz de cossenos diretores
15 C=eye(3,3)+(4*(p'*p-1)*Sp+8*Sp^2)/(1+p'*p)^2;
16 end

```

```

1 function w=velocidadeAngular(t)
2 % Funcao para fornecer a velocidade angular nos exemplos
3 w=[cos(2*pi*t)
4     cos(pi*t)
5     cos(0.5*pi*t)];
6 end

```







