Kathia Coronado

Robot Dynamics Hw3

Contents

problem 1a- derive dynamic model of robot

```
clear; clc;
syms m1 m2 m3 l1 l2 l3 q1(t) q2(t) q3(t) g

o1= [0;0;l1];
o2= [(l2*cos(q2))*cos(q1);(l2*cos(q2))*sin(q1);l1+l2*sin(q2)];
o3= [(l2*cos(q2)+l3*cos(q2+q3))*cos(q1);(l2*cos(q2)+l3*cos(q2+q3))*sin(q1);l1+l2*sin(q2)+l3*sin(q2+q3)];

v1= diff(o1,t);
v2= diff(o2,t);
v3=diff(o3,t);

% i had to separte the lement of v because its a symbolic function and
% and i dont know how to index
v2x= - l2*cos(q2(t))*sin(q1(t))*diff(q1(t), t) - l2*cos(q1(t))*sin(q2(t))*diff(q2(t), t);
v2y=   l2*cos(q1(t))*cos(q2(t))*diff(q1(t), t) - l2*sin(q1(t))*sin(q2(t))*diff(q2(t), t);
v2z=   l2*cos(q2(t))*diff(q2(t), t);
v3x= - cos(q1(t))*(l3*sin(q2(t) + q3(t))*(diff(q2(t), t) + diff(q3(t), t)) + l2*sin(q2(t))*diff(q2(t), t)) - sin(q1(t))*(l2*cos(q2(t)) + l3*cos(q2(t) + q3(t)))*diff
v3y=  cos(q1(t))*diff(q1(t), t)*(l2*cos(q2(t)) + l3*cos(q2(t) + q3(t))) - sin(q1(t))*(l3*sin(q2(t) + q3(t))*(diff(q2(t), t) + diff(q3(t), t)) + l2*sin(q2(t))*diff(q
v3z=   l3*cos(q2(t) + q3(t))*(diff(q2(t), t) + diff(q3(t), t)) + l2*cos(q2(t))*diff(q2(t), t);

%kinetic engery
k1= 0.5*m1*(v1(1)^2 + v1(2)^2 + v1(3)^2);
k2= 0.5*m2*(v2x^2 + v2y^2 + v2z^2);
k3= 0.5*m3*(v3x^2 + v3y^2 + v3z^2);

k= k1+k2+k3
% potential energy
p1= m1*g*l1;
p2= m2*g*(l1+l2*sin(q2));
p3= m3*g*(l1+l2*sin(q2)+l3*sin(q2+q3));
p= p1+p2+p3
% calc legrangian
L= k-p

% dynamical model
dL_qd1=diff(L,(diff(q1(t), t))); % derivate L wrt to qd1
dL_qd1t= diff(dL_qd1,t); % time derivative of L wrt to qd1
dL_q1= diff(L, q1); % derivative of L wrt to q1

dL_qd2=diff(L,(diff(q2(t), t))); % L wrt to qd2
dL_qd2t= diff(dL_qd2,t); % time derivative of L wrt to qd2
dL_q2= diff(L, q2); % derivative of L wrt to q1

dL_qd3=diff(L,(diff(q3(t), t))); % L wrt to qd3
dL_qd3t= diff(dL_qd3,t); % time derivative of L wrt to qd2
dL_q3= diff(L, q3); % derivative of L wrt to q1

tau1= dL_qd1t - dL_q1;
tau2= dL_qd2t - dL_q2;
tau3= dL_qd3t - dL_q3;
tau= tau1+tau2+tau3;
simplify(tau)
```

```
k =

(m2*((l2*cos(q2(t))*sin(q1(t))*diff(q1(t), t) + l2*cos(q1(t))*sin(q2(t))*diff(q2(t), t))^2 + (l2*cos(q1(t))*cos(q2(t))*diff(q1(t), t) - l2*sin(q1(t))*sin(q2(t))*dif


p(t) =

g*m2*(l1 + l2*sin(q2(t))) + g*m3*(l1 + l2*sin(q2(t)) + l3*sin(q2(t) + q3(t))) + g*l1*m1


L(t) =

(m2*((l2*cos(q2(t))*sin(q1(t))*diff(q1(t), t) + l2*cos(q1(t))*sin(q2(t))*diff(q2(t), t))^2 + (l2*cos(q1(t))*cos(q2(t))*diff(q1(t), t) - l2*sin(q1(t))*sin(q2(t))*dif


ans(t) =

(l2^2*m2*diff(q1(t), t, t))/2 + l2^2*m2*diff(q2(t), t, t) + (l2^2*m3*diff(q1(t), t, t))/2 + l2^2*m3*diff(q2(t), t, t) + (l3^2*m3*diff(q1(t), t, t))/2 + 2*l3^2*m3*di
```

problem 1b

```
l1= 0.3;l2= 0.3;l3= 0.3;
m1=0.5;m2=0.5;m3=0.5;
q1=0;q2=0;q3=0;
g= 9.8;
answer= subs(tau)
```

```
answer(t) =

147/25
```

problem 2: lagrange method

```
syms m1 m2 m3 l1 l2 l3 lc1 lc2 lc3 q1 q2 q3

vector= [l2*cos(q2)*cos(q1); l2*cos(q2)*sin(q1);0];

Slist=[[0;0;1;0;0;0],[0;-1;0;l1;0;0],[0 ;-1;0;l1;0;-l2]]
thetalist= [q1;q2;q3];
I = eye(3);
T={}

for i = 1 : length(thetalist)
    w = vector_2_skew(Slist(1:3,i));
    v = Slist(4:6,i);
    theta = thetalist(i);

    R = I + sin(theta)* w + (1-cos(theta))*w^2;

    star = (I * theta+(1-cos(theta))*w+(theta-sin(theta))*w^2)*v;

    T{i} = [R star; 0 0 0 1];

end

M01= [1 0 0 0; 0 1 0 0;0 0 1 0; 0 0 0 1];
T01=simplify(T{1}*M01)

M02=[1 0 0 0; 0 0 -1 0;0 1 0 l1; 0 0 0 1];
T02= simplify(T{1}*T{2}*M02)

M03=[1 0 0 l2;0 0 -1 0;0 1 0 l1;0 0 0 1];
T03= simplify(T{1}*T{2}*T{3}*M03)

T1_c1 = [1 0 0 0; 0 0 -1 0; 0 1 0 lc1; 0 0 0 1];
T0_c1 = simplify(T01*T1_c1);
T2_c2 = [1 0 0 lc2-l2; 0 1 0 0; 0 0 1 0; 0 0 0 1];
T0_c2 = simplify(T02*T2_c2);
T3_c3 = [1 0 0 lc3-l3; 0 1 0 0; 0 0 1 0; 0 0 0 1];
T0_c3 = simplify(T01*T3_c3);

oc1 = T0_c1(1:3,4);
Jv_c1 = jacobian(oc1, [q1 q2 q3]);

z0 = T01(1:3, 3);
zero = zeros([3,1]);
Jw_c1 = [z0 zero zero zero];

oc2 = T0_c2(1:3, 4);
z1 = T02(1:3,3);
Jv_c2 = jacobian(oc2, [q1 q2 q3]);

Jw_c2 = [z0 z1 zero zero];

oc3 = T0_c3(1:3, 4);
z2 = T03(1:3,3);
Jv_c3 = jacobian(oc3, [q1 q2 q3]);

Jw_c3 = [z0 z1 z2 zero];

syms m1 m2 m3
Dv = m1*Jv_c1.'*Jv_c1 + m2*Jv_c2.'*Jv_c2 + m3*Jv_c3.'*Jv_c3;
Dv = [Dv zeros(3,1); 0 0 0 0];

Rc1 = T0_c1(1:3, 1:3);
Rc2 = T0_c2(1:3, 1:3);
Rc3 = T0_c3(1:3, 1:3);

syms Iz1 Iz2 Iz3
I1 = sym('I1', [3,3]);
I2 = sym('I2', [3,3]);
I3 = sym('I3', [3,3]);

Dw = Jw_c1.'*Rc1*I1*Rc1.'*Jw_c1 + Jw_c2.'*Rc2*I2*Rc2.'*Jw_c2 + Jw_c3.'*Rc3*I3*Rc3.'*Jw_c3;
```

```
I1(3, 3) = Iz1;
I2(3, 3) = Iz2;
I3(3, 3) = Iz3;

Dw = Jw_c1.'*Rc1*I1*Rc1.'*Jw_c1 + Jw_c2.'*Rc2*I2*Rc2.'*Jw_c2 + Jw_c3.'*Rc3*I3*Rc3.'*Jw_c3;

D = simplify(Dv+Dw)
```

```
Slist =

[0,  0,   0]
[0, -1,  -1]
[1,  0,   0]
[0, l1,  l1]
[0,  0,   0]
[0,  0, -l2]


T =

  0×0 empty cell array
```

```
function X = vector_2_skew(x)

X=[0 -x(3) x(2) ; x(3) 0 -x(1) ; -x(2) x(1) 0 ];

end
```

```
T01 =

[cos(q1), -sin(q1), 0, 0]
[sin(q1),  cos(q1), 0, 0]
[      0,        0, 1, 0]
[      0,        0, 0, 1]


T02 =

[cos(q1)*cos(q2), -cos(q1)*sin(q2),  sin(q1),  0]
[cos(q2)*sin(q1), -sin(q1)*sin(q2), -cos(q1),  0]
[        sin(q2),          cos(q2),        0, l1]
[              0,                0,        0,  1]


T03 =

[cos(q2 + q3)*cos(q1), -sin(q2 + q3)*cos(q1),  sin(q1), l2*cos(q1)*cos(q2)]
[cos(q2 + q3)*sin(q1), -sin(q2 + q3)*sin(q1), -cos(q1), l2*cos(q2)*sin(q1)]
[        sin(q2 + q3),          cos(q2 + q3),        0,    l1 + l2*sin(q2)]
[                   0,                     0,        0,                  1]


D =

[I12_2 + Iz3 + l3^2*m3 + lc3^2*m3 + I22_2*cos(q2)^2 + (I21_2*sin(2*q2))/2 + (I22_1*sin(2*q2))/2 + I21_1*sin(q2)^2 - 2*l3*lc3*m3 + l2^2*m2*cos(q2)^2 + lc2^2*m2*cos(q
[                                                                                                                                                                  I23_2*cos
[
[
```

Published with MATLAB® R2020b