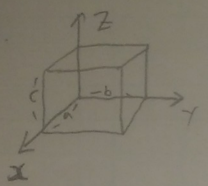
HW#5

ME 537 Jae Lee | 7-3, 7-5, 7-7, 7-10 | 2.

7-3.



$$I_{xx} = \int_0^c \int_0^b \int_0^a (y^2 + z^2)\rho(x,y,z)\,dx\,dy\,dz$$

$$= \int_0^c \int_0^b (y^2+z^2)a\,dy\,dz = \int_0^c \left[\frac{y^3}{3}a + z^2 a y\right]_0^b dz = \int_0^c \left(\frac{b^3}{3}a + z^2 ab\right)dz$$

$$= \left[\frac{b^3}{3}az + \frac{z^3}{3}ab\right]_0^c = \rho\frac{abc}{3}(b^2+c^2)$$

$$I_{xy} = -\int_0^c \int_0^b \int_0^a xy\,\rho(x,y,z)\,dx\,dy\,dz = -\int_0^c \int_0^b \frac{a^2}{2}y\,dy\,dz$$

$$= -\int_0^c \frac{a^2b^2}{4}\,dz = -\rho\frac{a^2b^2c}{4}$$

$$I = \rho\begin{bmatrix} \frac{abc}{3}(b^2+c^2) & -\frac{a^2b^2c}{4} & -\frac{a^2bc^2}{4} \\ -\frac{a^2b^2c}{4} & \frac{abc}{3}(a^2+c^2) & -\frac{ab^2c^2}{4} \\ -\frac{a^2bc^2}{4} & -\frac{ab^2c^2}{4} & \frac{abc}{3}(a^2+b^2) \end{bmatrix}$$

7-5. inertial matrix $D(q)$ for an n-link robot is always positive definite

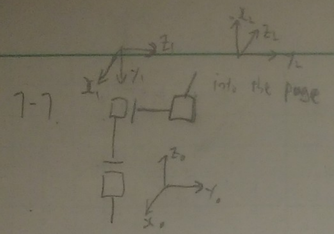
$$D(q) = \sum_{i=1}^{n}\left(m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R(q)^T J_{\omega_i}(q)\right)$$

$$K = \frac{1}{2}\dot{q}\,D(q)\,\dot{q}$$

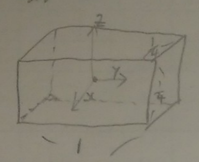
Since kinetic energy can't be negative, $D(q)$ must be positive definite

$J_{v_i}^T J_{v_i}$ is positive definite

$J_{\omega_i}(q)^T R_i(q) I_i R(q)^T J_{\omega_i}(q)$ is also positive definite

7-7. 

What is this Cartesian Manipulator's configuration?

(a) 

mass=1

equations on page 253.

$$I_{xx} = \frac{1}{12}\left(1^2 + \frac{1}{4}^2\right) = \frac{17}{192}$$

$$I_{yy} = \frac{1}{12}\left(\frac{1}{4}^2 + \frac{1}{4}^2\right) = \frac{1}{192}$$

$$I_{zz} = I_{yy} = \frac{1}{192}$$

(b) $$D(q) = \sum_{i=1}^{n}\left[m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R(q)^T J_{\omega_i}(q)\right]$$

$$J_{vc1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad J_{vc2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad J_{vc3} = \begin{bmatrix} z_0^0 & z_1^0 & z_2^0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$J_{vc1}^T J_{vc1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad J_{vc2}^T J_{vc2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad J_{vc3}^T J_{vc3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D(q) = \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_1+m_2 & 0 \\ 0 & 0 & m_1+m_2+m_3 \end{bmatrix}$$

(c) Christoffel $$C_{ijk} = \frac{1}{2}\left\{\frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} + \frac{\partial d_{ij}}{\partial q_k}\right\}$$

Since $D(q)$ is not a function of q_r, all Christoffel are zero.

(d) EOM in matrix form

$$D(q_r)\ddot{q}_r + C(q_r, \dot{q})\dot{q}_r + g(q_r) = u$$

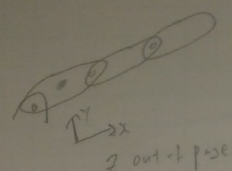
potential energy $P_i = m_i g^T r_{c,i}$

$$P = m_1 g^T r_{c1} + m_2 g^T r_{c2} + m_3 g^T r_{c3} = (m_1+m_2+m_3)\, 9.81\, (q_{r1} + d_{1,com})$$

$$\frac{\partial P}{\partial q_{r1}} = (m_1+m_2+m_3)\, 9.81 \qquad \frac{\partial P}{\partial q_{r2}} = 0 \qquad \frac{\partial P}{\partial q_{r3}} = 0$$

Thus, $$\begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_1+m_2 & 0 \\ 0 & 0 & m_1+m_2+m_3 \end{bmatrix}\begin{bmatrix} \ddot{q}_{r1} \\ \ddot{q}_{r2} \\ \ddot{q}_{r3} \end{bmatrix} + \begin{bmatrix} g(m_1+m_2+m_3) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

7-10



$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau$$

kinetic energy

$$J_{com,1} = \begin{bmatrix} z_0^0 \times (O^0_{com,1} - O^0_0) & 0 & 0 \\ z_0^0 & 0 & 0 \end{bmatrix}$$

I need $z_0^0 \quad z_1^0 \quad z_2^0$

$O^0_{com,1} \quad O^0_{com,2} \quad O^0_{com,3}$

potential energy

$O_0^0 \quad O_1^0 \quad O_2^0$

$$J_{com,3} = \begin{bmatrix} z_0^0 \times (O^0_{com,3} - O^0_0) & z_1^0 \times (O^0_{com,3} - O^0_1) & z_2^0 \times (O^0_{com,3} - O^0_2) \\ z_0^0 & z_1^0 & z_2^0 \end{bmatrix}$$

$$K = \frac{1}{2}\dot{q}^T \underbrace{\left[\sum_{i=1}^{n} m_i J_{v_i}(q)^T J_{v_i}(q) + J_{\omega_i}(q)^T R_i(q) I_i R(q)^T J_{\omega_i}(q)\right]}_{= D(q)} \dot{q}$$

$C_{111} = \frac{1}{2}$

$\boxed{\text{See the Code}}$

and plug into the equation below

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau$$

can get from MATLAB

7-10

```
clear;
close all;

a1 = 0.4;
a2 = 0.4;
a3 = 0.4;
%   theta d a alpha
L(1) = Link([ 0      0   a1  0], 'standard');
L(2) = Link([ 0      0   a2  0], 'standard');
L(3) = Link([ 0      0   a3  0], 'standard');


threelink = SerialLink(L, 'name', 'three link', ...
         'comment', 'from Spong, Hutchinson, Vidyasagar');
threelink.base = [1 0 0 0;
                  0 0 -1 0;
                  0 1 0 0;
                  0 0 0 1];

syms q1 q2 q3
qz = [q1 q2 q3];
% threelink.plot(qz);

T_1 = threelink.A(1,qz);
T_c1 = T_1;
T_c1(1:3,4) = T_c1(1:3,4)/2;
T_c1;

temp_t2 = threelink.A(2,qz);
T_2 = T_1*temp_t2;
temp_c2 = temp_t2;
temp_c2(1:3,4) = temp_c2(1:3,4)/2;
T_c2 = T_1*temp_c2;

temp_t3 = threelink.A(3,qz);
T_3 = T_2*temp_t3;
temp_c3 = temp_t3;
temp_c3(1:3,4) = temp_c3(1:3,4)/2;
T_c3 = T_2*temp_c3;

z = [0; 0; 1];
jacobian_c1 = [cross(z,T_c1(1:3,4)) zeros(3,2); z zeros(3,2)];
jacobian_c2 = [cross(z,T_c2(1:3,4)) cross(z,T_c2(1:3,4)-T_1(1:3,4))
 zeros(3,1); z z zeros(3,1)];
jacobian_c3 = [cross(z,T_c3(1:3,4)) cross(z,T_c3(1:3,4)-T_1(1:3,4))
 cross(z,T_c3(1:3,4)-T_2(1:3,4)); z z z];

syms Ixx Iyy Izz q_dot1 q_dot2 q_dot3
q_dot = [q_dot1; q_dot2; q_dot3];

mass = 1;
```

```
K_rot = 1/2*q_dot.'*(jacobian_c1(4:6,:).'*[Ixx 0 0; 0 Iyy 0; 0 0
 Izz]*jacobian_c1(4:6,:)...
                        + jacobian_c2(4:6,:).'*[Ixx 0 0; 0 Iyy 0; 0 0
 Izz]*jacobian_c2(4:6,:)...
                        + jacobian_c3(4:6,:).'*[Ixx 0 0; 0 Iyy 0; 0 0
 Izz]*jacobian_c3(4:6,:))*q_dot;

K_trans = 1/2*q_dot.'*(mass*jacobian_c1(1:3,:).'*jacobian_c1(1:3,:) +
 mass*jacobian_c2(1:3,:).'*jacobian_c2(1:3,:)...
                          +
 mass*jacobian_c3(1:3,:).'*jacobian_c3(1:3,:))*q_dot;
K_total = K_rot+K_trans;

D = jacobian_c1(4:6,:).'*[Ixx 0 0; 0 Iyy 0; 0 0
 Izz]*jacobian_c1(4:6,:)...
                        + jacobian_c2(4:6,:).'*[Ixx 0 0; 0 Iyy 0; 0 0
 Izz]*jacobian_c2(4:6,:)...
                        + jacobian_c3(4:6,:).'*[Ixx 0 0; 0 Iyy 0; 0 0
 Izz]*jacobian_c3(4:6,:)...
                        + mass*jacobian_c1(1:3,:).'*jacobian_c1(1:3,:) +
 mass*jacobian_c2(1:3,:).'*jacobian_c2(1:3,:)...
                        + mass*jacobian_c3(1:3,:).'*jacobian_c3(1:3,:);


for k=1:3
    for i=1:3
        for j=1:3
            c(i,j,k) = 1/2 * (diff(D(k,j),qz(i)) + diff(D(k,i),qz(j))
 - diff(D(i,j),qz(k)));
        end
    end
end

gravity = [0; 9.81; 0];
P = gravity'*T_c1(1:3,4) + gravity'*T_c2(1:3,4) +
 gravity'*T_c3(1:3,4);
g1 = diff(P,q1);
g2 = diff(P,q2);
g3 = diff(P,q3);
g = [g1; g2; g3];
% Plug into the EOM equation
D
c
g
```

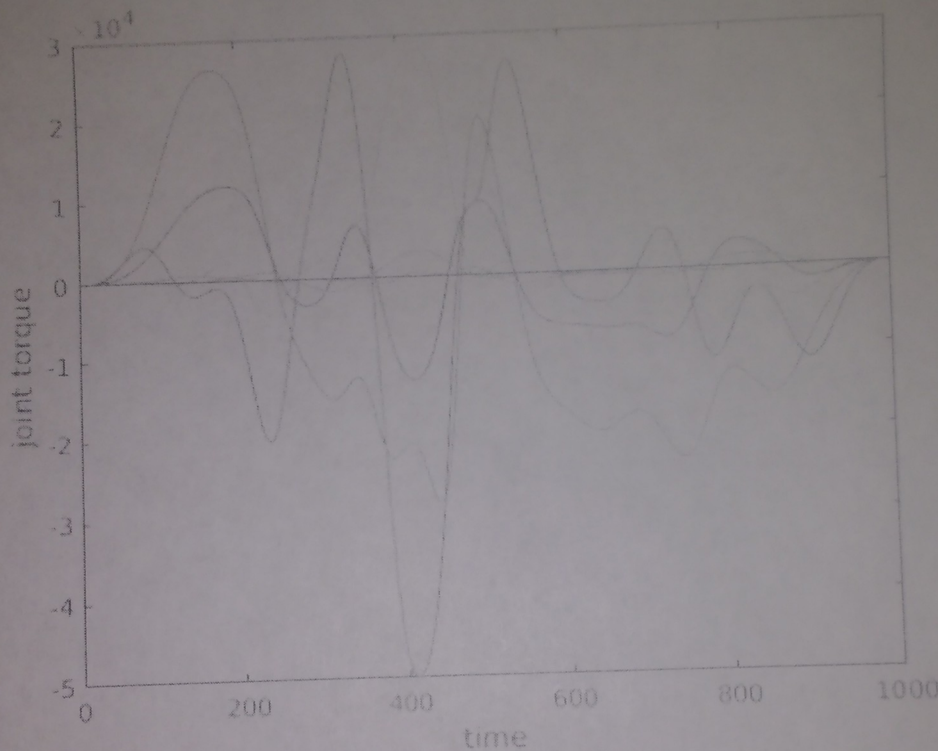
```
D =

[

                       3*Izz + ((2*cos(q1))/5 + (cos(q1)*cos(q2))/5
```

2-(b)

```
clear; close all;
storage = load('desired_accel.mat');
joint_angles = storage.q;
time = storage.t;

%calling the file as a function that returns a left and right
[left, right] = mdl_baxter('sim');

q_d0 = zeros(1,7);
q_dd0 = zeros(1,7);
q_d = zeros(1001,7);
q_dd = zeros(1001,7);
time_step = 0.01;
for i=2:length(time)-1
    q_d(i,:) = joint_angles(i+1,:)-joint_angles(i,:)/time_step;
    q_dd(i,:) = joint_angles(i
+1,:)-2*joint_angles(i,:)+joint_angles(i-1,:)/time_step^2;
end

joint_tau = zeros(7,length(time));
for i=1:length(time)
    M = left.inertia(joint_angles(i,:));
    C = left.coriolis(joint_angles(i,:),q_d(i,:));
    G = left.gravload(joint_angles(i,:));
    joint_tau(:,i) = M*q_dd(i,:)'+C*q_d(i,:)'+G';
end
```

Loaded Baxter Model in Simulation Mode (urdf-data)

```
plot(joint_tau');
xlabel('time');
ylabel('joint torque');
xlim([0 1001]);
```



Published with MATLAB® R2016a