José Corona

HomeTask5

Task1

My program is the file with name “Main.m”.

Point B is a revolute joint, with variable q1 (“phi”). Point L is a prismatic joint, with variable q2 (“s”)

q2 goes from “B” to ”S”. And B is the center of mass of the first link.

```
%origins and centers
O0 = [0 0 0]'
Oc1 = [0 0 0]'  %center of mass in B
O1 = [(l1)*cos(q1) (l1)*sin(q1) 0]'
Oc2 = [(l1+q2)*cos(q1) (l1+q2)*sin(q1) 0 ]'  %q2 0 to L
O2 = [((2*q2)+l1)*cos(q1) ((2*q2)+l1)*sin(q1) 0 ]'
```

```
%Axes rotation
Z0 = [0 0 1]'
Z1 = [cos(q1) sin(q1) 0]'
Zer = [0 0 0]';
```

```
%Jacobbians
Jv1 = [cross(Z0,(Oc1-O0)), Zer]
Jv2 = [cross(Z0,(Oc2-O0)), Z1]
Jw1 = [Z0, Zer]
Jw2 = [Z0, Zer]
```

We could try to calculate K

$$\mathcal{K} = \frac{1}{2}mv^Tv + \frac{1}{2}\omega^T\mathcal{I}\omega.$$

```
Vc1=Jv1(1:3,1)*dq1 + Jv1(1:3,2)*dq2;
Vc2=Jv2(1:3,1)*dq1 +  Jv2(1:3,2)*dq2;
Wc1=Jw1(1:3,1)*dq1 +  Jw1(1:3,2)*dq2;
Wc2=Jw2(1:3,1)*dq1 +  Jw2(1:3,2)*dq2;

K=0.5*m1*Vc1'*Vc1 + 0.5*m1*Vc2'*Vc2 + 0.5*Wc1'*I1*Wc1 + 0.5*Wc2'*I1*Wc2;
```

```
K =

(m1*(dq2*sin(q1) + dq1*cos(q1)*(l1 + q2))^2)/2 + (m1*(dq2*cos(q1) - dq1*sin(q1)*(l1 + q2))^2)/2 + I1*dq1^2
```

And matrix D is:

```
%D
D1 = m1*Jv1'*Jv1 + Jw1'*R1*I1*R1'*Jw1;
D2 = m2*Jv2'*Jv2 + Jw2'*R2*I2*R2'*Jw2;
D=D1+D2;
D=simplify(D)
```

```
D =

[ m2*l1^2 + 2*m2*l1*q2 + m2*q2^2 + I1 + I2,   0]
[                                        0, m2]
```

And P

```
%P
P1 = m1*g*0*sin(q1);
P2 = m2*g*((l1+q2)*sin(q1));
P = P1+P2
```

```
P =

g*m2*sin(q1)*(l1 + q2)
```

So the torques are

$$\frac{d}{dt}\left[\frac{\partial(\mathcal{K}-\mathcal{P})}{\partial \dot{q}}\right]-\frac{\partial(\mathcal{K}-\mathcal{P})}{\partial q}=\tau$$

```
tor =

ddq1*(m2*l1^2 + 2*m2*l1*q2 + m2*q2^2 + I1 + I2) + g*m2*cos(q1)*(l1 + q2) + 2*dq1*dq2*m2*(l1 + q2)
                                                   - m2*(l1 + q2)*dq1^2 + ddq2*m2 + g*m2*sin(q1)
```

Task2

$$M(q) * \ddot{q} + C(q,\dot{q}) * \dot{q} + G(q) = \tau\,(t)$$

```
D =

[ m2*l1^2 + 2*m2*l1*q2 + m2*q2^2 + I1 + I2,  0]
[                                        0, m2]
```

```
G =

 g*m2*cos(q1)*(l1 + q2)
           g*m2*sin(q1)


C =

[  dq2*m2*(l1 + q2), dq1*m2*(l1 + q2)]
[ -dq1*m2*(l1 + q2),                0]
```

```
tor =

ddq1*(m2*l1^2 + 2*m2*l1*q2 + m2*q2^2 + I1 + I2) + g*m2*cos(q1)*(l1 + q2) + 2*dq1*dq2*m2*(l1 + q2)
                                                   - m2*(l1 + q2)*dq1^2 + ddq2*m2 + g*m2*sin(q1)
```

Task3

First we substitute the values

```
%subs
D(q1,q2) = subs(D,{m2, I1, I2, l1},{2 1 2 0.2});
C(q1,q2,dq1,dq2) = subs(C*dq ,{m2, I1, I2, l1},{2 1 2 0.2});
G(q1,q2) = subs(G ,{m2, I1, I2, l1,g},{2 1 2 0.2 9.8});
```

Select initial condition.   In this case the position is down

```
%initial
q1_0 = -pi/2; % position
q2_0 = 0;
dq1_0 = 0; %velocity
dq2_0 = 0;
ddq1p_0 =0; %acceleration
ddq2p_0 =0;
dt=0.01;
```

Select a torque to apply.

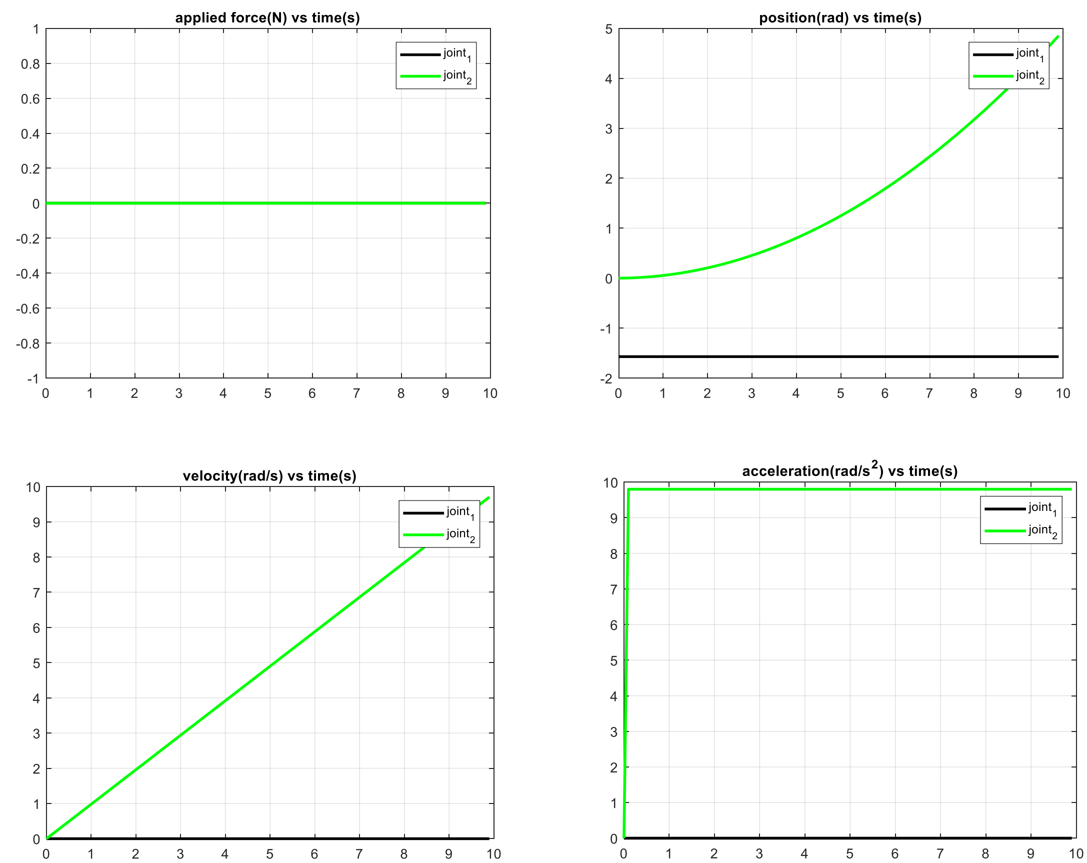
```
%force funtion to applied
u1p_0 = 0;
u2p_0 = 0;
for i = 1:n
    u1p(i)=u1p_0;
    u2p(i)=u2p_0;
end
```

Solve the inverse dynamics to get the positions, velocities and accelerations functions of the joints

```
%Appling a force to the joints
for i = 1:n
     U = [u1p(i);u2p(i)];
     q1p(i)=q1_0;
     q2p(i)=q2_0;
     dq1p(i)=dq1_0;
     dq2p(i)=dq2_0;
     ddq1p(i)=ddq1p_0;
     ddq2p(i)=ddq2p_0;
     %inverse dinamics
     ddq = inv(D(q1_0, q2_0))*(U-C(q1_0, q2_0,dq1_0,dq2_0)-G(q1_0,q2_0));
     ddq1p_0 = ddq(1);
     ddq2p_0 = ddq(2);
     %small aceleration
     dq1_0=dq1p(i) + double(ddq(1)*dt);
     dq2_0=dq2p(i) + double(ddq(2)*dt);
     q1_0 = q1p(i) + dq1_0*dt;
     q2_0 = q2p(i) + dq2_0*dt;
 end
```

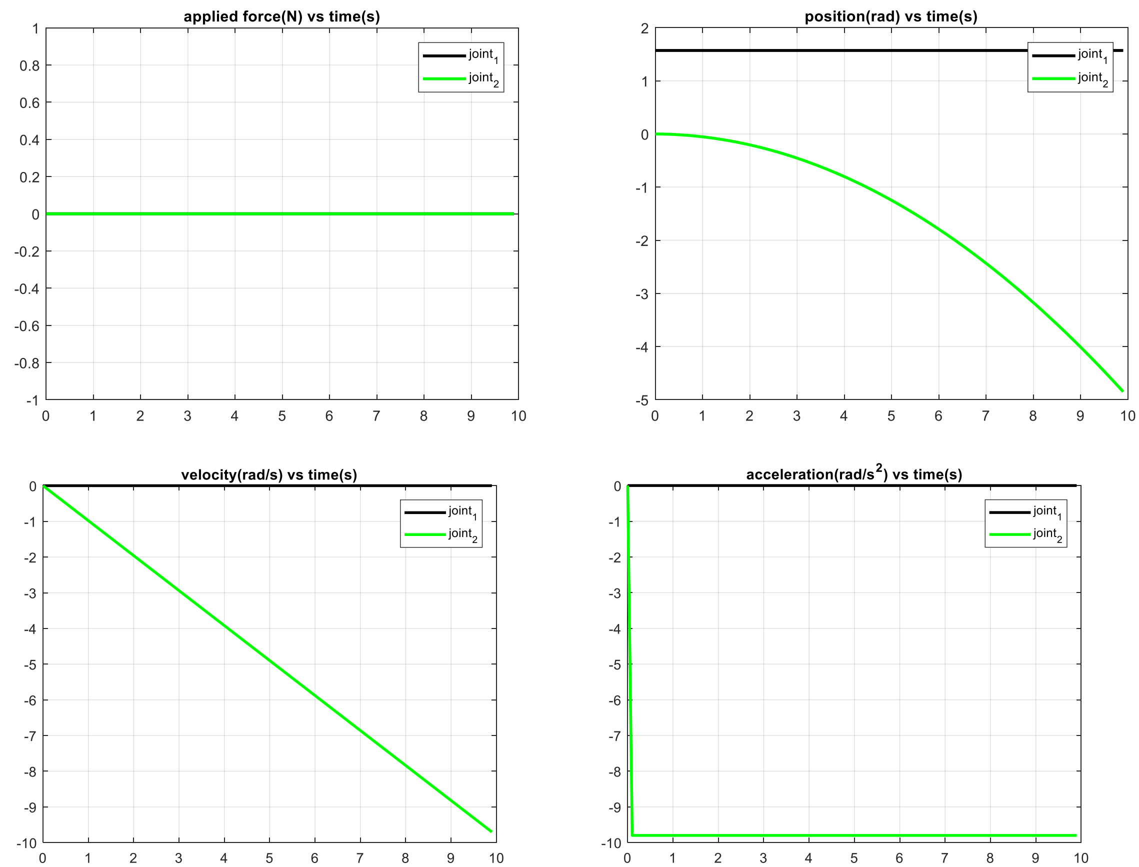
So appling no force, and in down position we get this graphic. Since is no restriction in q2, not define q2 max, the mass pointing down falls in the plane vertically; the acceleration of q2 is positive since the axis of this link is pointing down.

```
%initial conditions
q1_0 = -pi/2; % position
q2_0 = 0;
dq1_0 = 0; %velocity
dq2_0 = 0;
ddq1p_0 =0; %acceleration
ddq2p_0 =0;
dt=0.01;     %step in seconds
n=100;       %total steps

%force funtion to applied
u1p_0 = 0;
u2p_0 = 0;
for i = 1:n
   u1p(i)=u1p_0;
   u2p(i)=u2p_0;
end
```



So if we put the robot in “up” position, q1=pi/2 , the acceleration of q2 is negative since the axis of this link is pointing up.

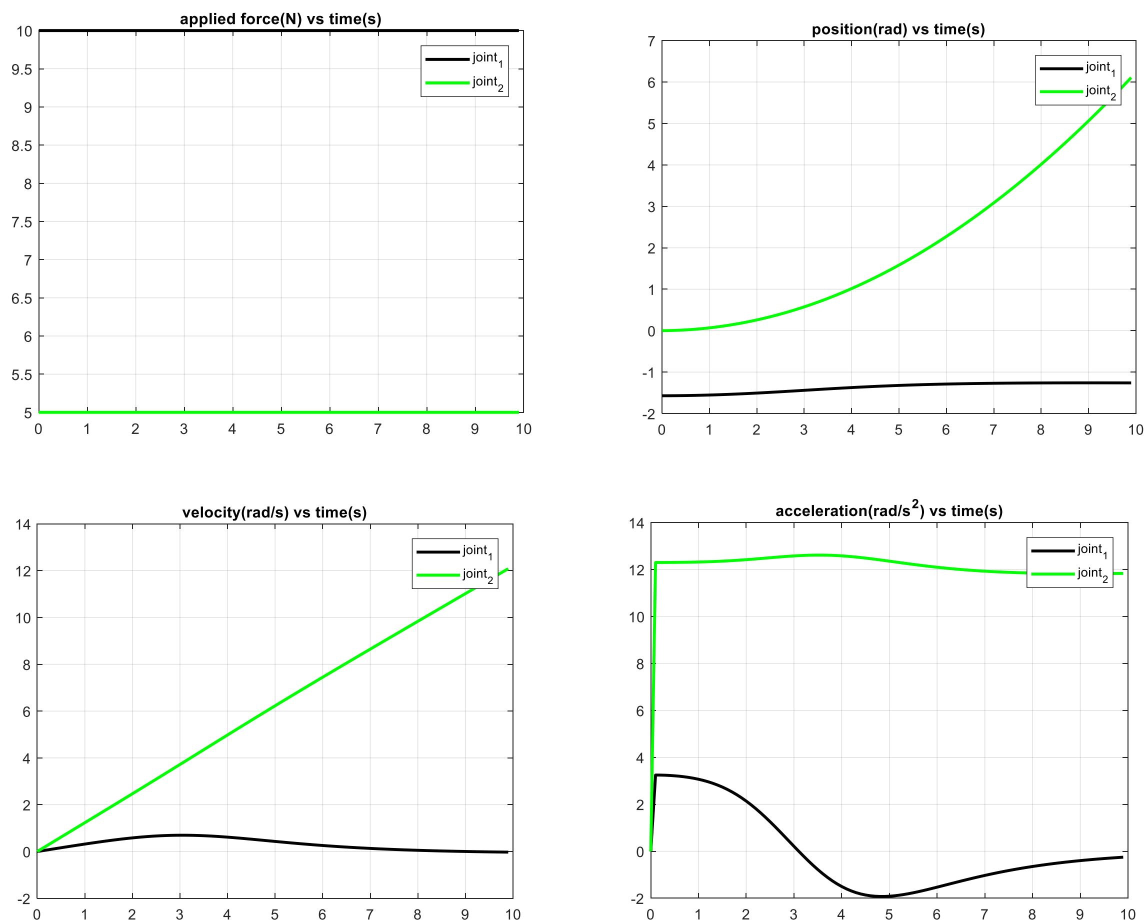
```
%initial conditions
q1_0 = pi/2; % position
q2_0 = 0;
dq1_0 = 0; %velocity
dq2_0 = 0;
ddq1p_0 =0; %acceleration
ddq2p_0 =0;
dt=0.01;     %step in seconds
n=100;       %total steps

%force funtion to applied
u1p_0 = 0;
u2p_0 = 0;
for i = 1:n
    u1p(i)=u1p_0;
    u2p(i)=u2p_0;
end
```



Appling a constant force in q1 and q2, with initial down position.

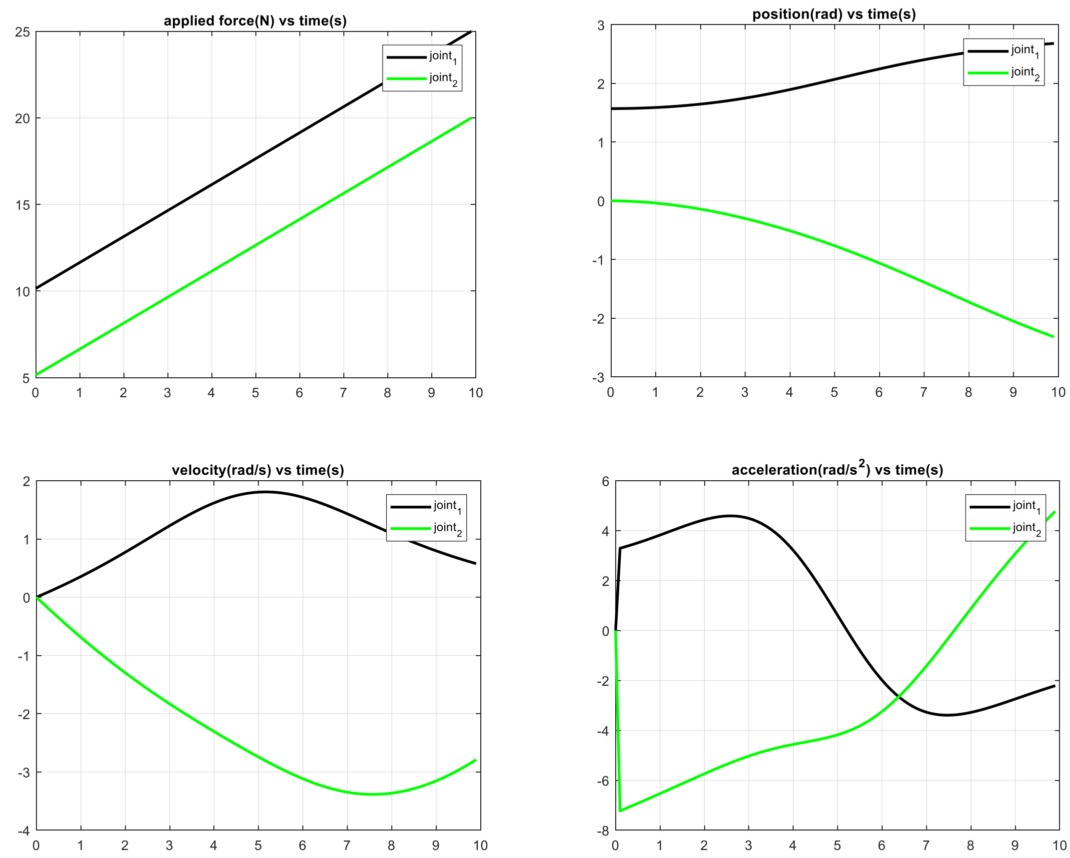
```
%force funtion to applied
u1p_0 = 10;
u2p_0 = 5;
for i = 1:n
    u1p(i)=u1p_0;
    u2p(i)=u2p_0;
end
```

```
%initial conditions
q1_0 = -pi/2; % position
q2_0 = 0;
dq1_0 = 0; %velocity
dq2_0 = 0;
ddq1p_0 =0; %acceleration
ddq2p_0 =0;
dt=0.01;     %step in seconds
n=100;       %total steps
```



Appling a constant increment force in q1 and q2, with initial up position.

```
%initial conditions
q1_0 = pi/2; % position
q2_0 = 0;
dq1_0 = 0; %velocity
dq2_0 = 0;
ddq1p_0 =0; %acceleration
ddq2p_0 =0;
dt=0.01;     %step in seconds
n=100;       %total steps

%force funtion to applied
u1p_0 = 10;
u2p_0 = 5;
for i = 1:n
    u1p(i)=u1p_0+(i*0.15);
    u2p(i)=u2p_0+(i*0.15);
end
```



Link:

https://github.com/Jose-R-Corona/Hometask5