Table of Contents


```
clc
close all
clear all
```

ME 292B - HW #2

System Constants

```
syms g L_torso L_leg1 L_leg2 m_torso m_leg1 m_leg2...
    I_torso I_leg1 I_leg2 real
```

Configuration Variables (Figure 1a)

```
syms x y q1 q2 q3 dx dy dq1 dq2 dq3 d2x d2y d2q1 d2q2 d2q3 real
% Generalized Coordinates
q = [x; y; q1; q2; q3] ;
% Generalized Velocities
dq = [dx; dy; dq1; dq2; dq3] ;
% Generalized Acceleration
d2q = [d2x; d2y; d2q1; d2q2; d2q3] ;
```

Define numerical values / vectors (cases (i) and (ii))

```
qi = [0.5, 0.5*sqrt(3), 150*pi/180, 120*pi/180, 30*pi/180]';
dqi = [-0.8049, -0.4430, 0.0938, 0.9150, 0.9298]';
qii = [0.3420, 0.9397, 170*pi/180, 20*pi/180, 30*pi/180]';
dqii = [-0.1225, -0.2369, 0.5310, 0.5904, 0.6263]';

L_torso_num = 1/2; L_leg1_num = 1; L_leg2_num = 1;
m_torso_num = 10; m_leg1_num = 5; m_leg2_num = 5;
I_torso_num = 1; I_leg1_num = 1/2; I_leg2_num = 1/2;
```

```
g_num = 9.81;
```

Problem 1(a)

```
% Assume leg 1 is the stance leg, then from geometry find p_st
p_st = [x - L_leg1*sin(q3 + q1 - pi);y - L_leg1*cos(q3 + q1 - pi)];
p_st = simplify(p_st) % m

p_st_i = subs(p_st, q, qi);
p_st_i = subs(p_st_i, L_leg1, L_leg1_num);
p_st_i = vpa(p_st_i,4) % m

p_st_ii = subs(p_st, q, qii);
p_st_ii = subs(p_st_ii, L_leg1, L_leg1_num);
p_st_ii = vpa(p_st_ii,4) % m
```

```
p_st =

 x + L_leg1*sin(q1 + q3)
 y + L_leg1*cos(q1 + q3)


p_st_i =

    0.5
 -0.134


p_st_ii =

 -2.014e-5
  7.379e-6
```

Problem 1(b)

```
J_st = jacobian(p_st, q);
J_st = simplify(J_st)

J_st_i = subs(J_st, q, qi);
J_st_i = subs(J_st_i, L_leg1, L_leg1_num);
J_st_i = vpa(J_st_i,4) % m

J_st_ii = subs(J_st, q, qii);
J_st_ii = subs(J_st_ii, L_leg1, L_leg1_num);
J_st_ii = vpa(J_st_ii,4) % m
```

```
J_st =

[ 1, 0,  L_leg1*cos(q1 + q3), 0,  L_leg1*cos(q1 + q3)]
[ 0, 1, -L_leg1*sin(q1 + q3), 0, -L_leg1*sin(q1 + q3)]
```

```
J_st_i =

[ 1.0,    0, -1.0, 0, -1.0]
[   0, 1.0,    0, 0,    0]


J_st_ii =

[ 1.0,    0, -0.9397, 0, -0.9397]
[   0, 1.0,   0.342, 0,   0.342]
```

Problem 1(c)

```
% Take derivative of Jacobian with respect to time
for i = 1:size(J_st,1)
    for j = 1:size(J_st,2)
        J_st_dot(i,j) = jacobian(J_st(i,j), q) * dq;
    end
end

J_st_dot_i = subs(J_st_dot, q, qi);
J_st_dot_i = subs(J_st_dot_i, dq, dqi);
J_st_dot_i = subs(J_st_dot_i, L_leg1, L_leg1_num);
J_st_dot_i = vpa(J_st_dot_i,4) % m

J_st_dot_ii = subs(J_st_dot, q, qii);
J_st_dot_ii = subs(J_st_dot_ii, dq, dqii);
J_st_dot_ii = subs(J_st_dot_ii, L_leg1, L_leg1_num);
J_st_dot_ii = vpa(J_st_dot_ii,4) % m
```

```
J_st_dot_i =

[ 0, 0,     0, 0,     0]
[ 0, 0, 1.024, 0, 1.024]


J_st_dot_ii =

[ 0, 0, 0.3958, 0, 0.3958]
[ 0, 0,  1.088, 0,  1.088]
```

Problem 1(d)

```
% Import D,C,G,B vectors for cases (i) and (ii) from homework 1
Di = HW1_Di;
Dii = HW1_Dii;
Ci = HW1_Ci;
Cii = HW1_Cii;
```

```
Gi = HW1_Gi;
Gii = HW1_Gii;
Bi = HW1_Bi;
Bii = HW1_Bii;

% Given input
u = [0;0];

% From class notes, we have two equations and two unknowns (ddq and
 F_st)
% Two equations are,
% D(q)*d2q + C(q, dq)*dq + G(q) = B(q)*u + J_st'(q)*F_st
% J_st(q)*d2q + dJ_st(q,dq) = 0

% we can implement these equations in matrix form and solve for
 unknown
% using Ax = b, x = A\b. F_ext is displayed below for each
 configuration

% Case (i)
A = [Di,-J_st_i';J_st_i,zeros(2,2)];
b = [Bi*u-Ci*dqi-Gi;-J_st_dot_i*dqi];
xi = A \ b;

F_ext_i = vpa(xi(6:7), 4)

% Case (ii)
A = [Dii,-J_st_ii';J_st_ii,zeros(2,2)];
b = [Bii*u-Cii*dqii-Gii;-J_st_dot_ii*dqii];
xii = A \ b;

F_ext_ii = vpa(xii(6:7), 4)
```

```
F_ext_i =

 0.2878
  167.2


F_ext_ii =

 43.57
 139.6
```

Problem 2(a)

```
% Input q- and dq-
q_minus = [0.3827 0.9239 3.0107 2.2253 0.5236]';
dq_minus = [1.4782 -0.6123 1.6 -1.6 0]';

% Compute B,C,D,G from q_minus and dq_minus
```

```
% NOTE: B,C,D,G functions were imported from HW1

B_q2 = HW1_B;

C_q2 = HW1_C(L_leg1_num,L_leg2_num,L_torso_num,dq_minus(3),...
    dq_minus(4),dq_minus(5),m_leg1_num,m_leg2_num,m_torso_num,...
    q_minus(3),q_minus(4),q_minus(5));

D_q2 =
 HW1_D(I_leg1_num,I_leg2_num,I_torso_num,L_leg1_num,L_leg2_num,...
    L_torso_num,m_leg1_num,m_leg2_num,m_torso_num,q_minus(3),...
    q_minus(4),q_minus(5));

G_q2 = HW1_G(L_leg1_num,L_leg2_num,L_torso_num,g_num,...

 m_leg1_num,m_leg2_num,m_torso_num,q_minus(3),q_minus(4),q_minus(5));

% Assume leg 2 is the swing leg, then from geometry find p_sw
p_sw = [x - L_leg1*sin(q3 + q2 - pi);y - L_leg1*cos(q3 + q2 - pi)];
p_sw = simplify(p_sw) % m

% Find J_sw for given pre-impact state
J_sw = jacobian(p_sw, q);

J_sw = subs(J_sw, q, q_minus);
J_sw = subs(J_sw, dq, dq_minus);
J_sw = subs(J_sw, [L_leg1], [L_leg1_num]);
J_sw = vpa(J_sw, 4) % m

% From class notes, we have two equations and two unknowns (dq_plus
 and
% F_c). The two equations are,
% D(q_plus)*dq_plus - D(q_minus)*dq_minus = F_ext
% J_sw(q)*dq_plus = 0

% Note also that F_ext = J_sw'*F_c

% we can implement these equations in matrix form and solve for
 unknowns
% using Ax = b, x = A\b
A = [D_q2, -J_sw'; J_sw, zeros(2,2)];
b = [D_q2*dq_minus; zeros(2,1)];
x_q2 = A\b;

dq_plus = x_q2(1:5);
dq_plus = vpa(dq_plus, 4)

F_c = x_q2(6:7);
F_c = vpa(F_c, 4)


p_sw =

 x + L_leg1*sin(q2 + q3)
```

```
 y + L_leg1*cos(q2 + q3)


J_sw =

[ 1.0,    0, 0, -0.9239, -0.9239]
[    0, 1.0, 0, -0.3827, -0.3827]


dq_plus =

  0.9038
  0.3743
  -1.222
 -0.5461
   1.524


F_c =

 -11.15
  14.12
```

Problem 2(b)

```
% Finally, we can calculate F_ext from the principal of virtual work,
 this
% will be the impact impulse at the swing leg
F_ext = J_sw'*F_c;
F_ext = vpa(F_ext, 4)
```

```
F_ext =

 -11.15
  14.12
      0
  4.895
  4.895
```

Published with MATLAB® R2019a