MEC ENG 193B/292B: Feedback Control of Legged Robots

Homework 2

Professor Koushil Sreenath
UC Berkeley, Department of Mechanical Engineering
September 18, 2025

Larry Hui[1]
SID: 3037729658

[1]University of California at Berkeley, College of Engineering, Department of Mechanical Engineering. Author to whom any correspondence should be addressed. email: larryhui7@berkeley.edu

1 Problems

1.1 Dynamics of Systems with Constraints

The dynamics derived in HW #1 represents a freely falling three-link robot with two motor inputs at the hip. To derive the dynamics of a walking robot, we need to add an additional constraint while formulating the dynamics, which is enforced through an external (constraint) force. For walking, we assume the stance foot of the robot to be fixed w.r.t. the ground. This is enforced through the ground reaction forces acting on the stance foot. Such a constraint can be written as,

$$p_{st}(q) \equiv c, \tag{1}$$

where $p_{st} \in \mathbb{R}^2$ is the position of the stance foot and $c \in \mathbb{R}^2$ is a constant. For instance, if the stance foot is constrained to be on the ground at the origin, then $c = [0 \quad 0]^\top$. This type of constraint (a constraint only on the configuration position variables) is called a *holonomic constraint*. Then, the external forces at the stance foot, $F_{st} \in \mathbb{R}^2$ enter the system dynamics as,

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = B(q)u + J_{st}(q)^\top F_{st}, \tag{2}$$

where $J_{st} = \frac{\partial p_{st}}{\partial q} \in \mathbb{R}^{2\times 5}$ is the *Jacobian* of the constraint (position of the stance foot here). In this problem, we will derive the expression for this external force that enforces the above holonomic constraint.

(a) Symbolically compute the expression of the position of the stance foot p_{st} as a function of the configuration variables. Compute the value of the stance foot position for the two numerical configurations (given in Problem 1 of HW #1.)

Proof. Let us assume that leg 1 is to be the stance leg since foot 1 is lower than foot 2. Since we want the stance foot position $p_{st}(q)$ to be in terms of the configuration variables, we can write the absolute leg angles in terms of the relative angles of the legs to the torso

$$q_3 = \theta_1 - q_1 \implies \theta_1 = q_1 + q_3$$

Then, we can write the position of the stance foot using the full leg 1 length as

$$p_{st}(q) = \begin{bmatrix} x \\ y \end{bmatrix} + l_{leg1} \begin{bmatrix} \sin(q_1 + q_3) \\ \cos(q_1 + q_3) \end{bmatrix}$$

Using MATLAB, for configuration 1 and 2 we have the following numerical values

$$P_{st,1}(q) = \begin{bmatrix} 0.5000 \\ -0.1340 \end{bmatrix} \mathtt{m}, \quad P_{st,2}(q) = \begin{bmatrix} -2.0143 \times 10^{-5} \\ 7.3792 \times 10^{-6} \end{bmatrix} \mathtt{m}$$

□

(b) Compute the expression of the Jacobian of the position of the stance foot J_{st} as a function of the configuration variables. Compute its value for the two numerical configurations.

Proof. Given our holonomic constraint $p_{st}(q) \equiv c$, we can write our constraint function

$$\gamma(q) \triangleq p_{st}(q) - c$$

The Jacobian of the constraint would then be only of $p_{st}(q)$ since c is constant hence

$$\begin{aligned} J_{st}(q) &= \frac{\partial \gamma(q)}{\partial q} \\ &= \frac{\partial p_{st}(q)}{\partial q} = \begin{bmatrix} \frac{\partial p_{st,x}}{\partial x} & \frac{\partial p_{st,x}}{\partial y} & \frac{\partial p_{st,x}}{\partial q_1} & \frac{\partial p_{st,x}}{\partial q_2} & \frac{\partial p_{st,x}}{\partial q_3} \\ \frac{\partial p_{st,y}}{\partial x} & \frac{\partial p_{st,y}}{\partial y} & \frac{\partial p_{st,y}}{\partial q_1} & \frac{\partial p_{st,y}}{\partial q_2} & \frac{\partial p_{st,y}}{\partial q_3} \end{bmatrix} \end{aligned}$$

Then, we can take partial derivatives

$$\frac{\partial p_{st,x}}{\partial x} = 1, \quad \frac{\partial p_{st,x}}{\partial y} = 0, \quad \frac{\partial p_{st,x}}{\partial q_1} = l_{\text{leg}}\cos(q_1+q_3), \quad \frac{\partial p_{st,x}}{\partial q_2} = 0, \quad \frac{\partial p_{st,x}}{\partial q_3} = l_{\text{leg}}\cos(q_1+q_3)$$

$$\frac{\partial p_{st,y}}{\partial x} = 0, \quad \frac{\partial p_{st,y}}{\partial y} = 1, \quad \frac{\partial p_{st,y}}{\partial q_1} = -l_{\text{leg}}\sin(q_1+q_3), \quad \frac{\partial p_{st,y}}{\partial q_2} = 0, \quad \frac{\partial p_{st,y}}{\partial q_3} = -l_{\text{leg}}\sin(q_1+q_3)$$

Assembling the Jacobian, we have

$$J_{st}(q) = \frac{\partial p_{st}(q)}{\partial q} = \begin{bmatrix} 1 & 0 & l_{leg}\cos(q_1+q_3) & 0 & l_{leg}\cos(q_1+q_3) \\ 0 & 1 & -l_{leg}\sin(q_1+q_3) & 0 & -l_{leg}\sin(q_1+q_3) \end{bmatrix}$$

Using MATLAB, for configurations 1 and 2, we have the following numerical results

$$J_{st,1}(q) = \begin{bmatrix} 1 & 0 & -1 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad J_{st,2}(q) = \begin{bmatrix} 1 & 0 & -0.9397 & 0 & -0.9397 \\ 0 & 1 & 0.3420 & 0 & 0.3420 \end{bmatrix}$$

□

(c) Compute the expression of the time derivative of the Jacobian of the stance foot $\dot{J}_{st}$ as a function of the configuration variables and velocities. Report its value for the given two configurations and velocities.

Proof. Taking the derivative of of the Jacobian above, we note that the only terms dependent on time are the configuration variables q_1 and q_3, so by the chain rule, we have the following

$$\dot{J}_{st}(q,\dot{q}) = \begin{bmatrix} 0 & 0 & -l_{leg}\sin(q_1+q_3)(\dot{q}_1+\dot{q}_3) & 0 & -l_{leg}\sin(q_1+q_3)(\dot{q}_1+\dot{q}_3) \\ 0 & 0 & -l_{leg}\cos(q_1+q_3)(\dot{q}_1+\dot{q}_3) & 0 & -l_{leg}\cos(q_1+q_3)(\dot{q}_1+\dot{q}_3) \end{bmatrix}$$

Using MATLAB, for configurations 1 and 2, we have the following numerical results

$$\dot{J}_{st,1}(q,\dot{q}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0236 & 0 & 1.0236 \end{bmatrix}, \quad \dot{J}_{st,2}(q,\dot{q}) = \begin{bmatrix} 0 & 0 & 0.3958 & 0 & 0.3958 \\ 0 & 0 & 1.0875 & 0 & 1.0875 \end{bmatrix}$$

□

(d) Compute the external force required to enforce the constraint (1) for the given two configurations and velocities. Assume the motor inputs to be 0, i.e., $u = [0 \quad 0]^\top$.

Proof. The constrained dynamics are given by

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(q)u + J^\top(q)F_{st}$$

From the constraint equation for the holonomic constraint, we have that

$$p_{st}(q) = c \implies J_{st}(q)\dot{q} = 0 \implies \frac{\partial}{\partial q}(J(q)\dot{q})\dot{q} + J(q)\ddot{q} = 0$$

Since we assume the motor inputs to be $u = 0$, we have in matrix form

$$\begin{bmatrix} D(q) & -J(q)^\top \\ J(q) & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ F_{st} \end{bmatrix} = \begin{bmatrix} -C(q,\dot{q})\dot{q} - G(\dot{q}) \\ -\dot{J}(q,\dot{q})\dot{q} \end{bmatrix}$$

Using MATLAB and our previous code in homework 1 coupled with Lagrangian Dynamics, we get the following numerical values for configuration 1 and 2 are

$$F_{st1} = \begin{bmatrix} 0.2878 \\ 167.1578 \end{bmatrix} \texttt{N}, \quad F_{st2} = \begin{bmatrix} 43.5687 \\ 139.5922 \end{bmatrix} \texttt{N}$$

□

1.2 Impact Map

Derive the expression for the function that computes (a) the post-impact velocity given the pre-impact velocity and (b) the impact impulse at the swing leg. Assume a rigid plastic impact (i.e. coefficient of restitution is zero) and that the swing food does not slip or bounce on impact.

Report the values of the post-impact velocities and the impact impulse given the pre-impact state

$$(q^-, \dot{q}^-) = \left(\begin{bmatrix} 0.3827m \\ 0.9239m \\ 3.0107rad \\ 2.2253rad \\ 0.5236rad \end{bmatrix}, \begin{bmatrix} 1.4782m/s \\ -0.6123m/s \\ 1.6rad/s \\ -1.6rad/s \\ 0rad/s \end{bmatrix} \right)$$

Proof. By the conservation of generalized momentum, we can compute the impact dynamics with leg 1 as the stance foot and leg 2 hitting the ground as

$$D(q^+)\dot{q}^+ - D(q^-)\dot{q}^- = J_2(q)^\top F_G, \quad J_2(q^+)\dot{q}^+ = 0$$

By substituting the rearranged momentum balance into the constraint and assuming that the impact is infinitesimal with no change in configuration, we can treat $D(q^+) = D(q^-) = D(q)$

$$D(q)(\dot{q}^+ - \dot{q}^-) = J_2(q)^\top F_G \implies \dot{q}^+ = \dot{q}^- + D(q)^{-1} J_2(q)^\top F_G \implies J_2(q)\dot{q}^- + J_2(q)D(q)^{-1}J_2(q)^\top F_G = 0$$

Then solving for the ground impulse, we have

$$F_G = -(J_2(q)D(q)^{-1}J_2(q)^\top)^{-1} J_2(q)\dot{q}^-$$

Substituting this back into our expression for post-impact velocity $\dot{q}^+$ we get

$$\dot{q}^+ = \dot{q}^- - D(q)^{-1}J_2(q)^\top (J_2(q)D(q)^{-1}J_2(q)^\top)^{-1} J_2(q)\dot{q}^-$$

Or in matrix form we have

$$\Delta_2 : \begin{bmatrix} D(q^+) & -J_2(q)^\top \\ J_2(q^+) & 0 \end{bmatrix} \begin{bmatrix} \dot{q}^+ \\ F_G \end{bmatrix} = \begin{bmatrix} D(q^-)\dot{q}^- \\ 0 \end{bmatrix}$$

Substituting in our pre-impact state, we have the following

$$\dot{q}^+ = \begin{bmatrix} 0.9038m/s \\ 0.3743m/s \\ -1.2219rad/s \\ -0.5461rad/s \\ 1.5243rad/s \end{bmatrix}, \quad F_G = \begin{bmatrix} -11.1464 \\ 14.1196 \end{bmatrix} \mathtt{N{\cdot}s}$$

$\square$

2 Code Appendix

```
% rel. coords
syms x y q1 q2 q3 xdot ydot q1dot q2dot q3dot real

% generalized coords
q = [x y q1 q2 q3];

% generalized velocity
dq = [xdot ydot q1dot q2dot q3dot];

% constants
m_torso = 10;  % kg
m_leg = 5;     % kg
I_torso = 1;   % kg-m^2
I_leg = 0.5;   % kg-m^2
l_torso = 0.5; % m
l_leg = 1;     % m
g = 9.81;      % m/s^2

% Configuration 1
conf1 = [0.5, sqrt(3)/2, deg2rad(150), deg2rad(120), deg2rad(30)];

% Configuration 2
conf2 = [0.3420, 0.9397, deg2rad(170), deg2rad(20), deg2rad(30)];


%% Q1 PART A
p_stleg1 = [x; y] + (l_leg) * [sin(q1 + q3); cos(q1 + q3)];

Pst1 = double(subs(p_stleg1, q, conf1));
Pst2 = double(subs(p_stleg1, q, conf2));

%% Q1 PART B
J_st = jacobian(p_stleg1, q);

Jst1 = double(subs(J_st, q, conf1));
Jst2 = double(subs(J_st, q, conf2));

%% Q1 PART C
Jst_dot = sym(zeros(size(J_st)));

for i = 1:length(q)
    Jst_dot = Jst_dot + diff(J_st, q(i)) * dq(i);
end

qdot1 = [-0.8049, -0.4430, 0.0938, 0.9150, 0.9298];
qdot2 = [-0.1225, -0.2369, 0.5310, 0.5904, 0.6263];

Jstdot1 = double(subs(Jst_dot, [q, dq], [conf1, qdot1]));
Jstdot2 = double(subs(Jst_dot, [q, dq], [conf2, qdot2]));

%% Q1 PART D
p_leg1 = [x; y] + (l_leg/2) * [sin(q1 + q3); cos(q1 + q3)];
p_leg2 = [x; y] + (l_leg/2) * [sin(q2 + q3); cos(q2 + q3)];
p_torso = [x; y] + (l_torso/2) * [sin(q3); cos(q3)];

P = [p_leg1 p_leg2 p_torso];

P1 = double(subs(P, q, conf1));
P2 = double(subs(P, q, conf2));

qdot1 = [-0.8049, -0.4430, 0.0938, 0.9150, 0.9298];
qdot2 = [-0.1225, -0.2369, 0.5310, 0.5904, 0.6263];

dP_leg1 = simplify(jacobian(p_leg1, q) * dq');
dP_leg2 = simplify(jacobian(p_leg2, q) * dq');
```

```
dP_torso = simplify(jacobian(p_torso, q) * dq');

dP = [dP_leg1 dP_leg2 dP_torso];

dP1 = double(subs(dP, [q, dq], [conf1, qdot1]));
dP2 = double(subs(dP, [q, dq], [conf2, qdot2]));

T_leg1 = 0.5 * m_leg * (dP_leg1' * dP_leg1) + 0.5 * I_leg * (q3dot + q1dot)^2;
T_leg2 = 0.5 * m_leg * (dP_leg2' * dP_leg2) + 0.5 * I_leg * (q3dot + q2dot)^2;
T_torso =  0.5 * m_torso * (dP_torso' * dP_torso) + 0.5 * I_torso * (q3dot)^2;

T = T_leg1 + T_leg2 + T_torso;

T1 = double(subs(T, [q, dq], [conf1, qdot1]));
T2 = double(subs(T, [q, dq], [conf2, qdot2]));

e2 = [0; 1];

U_leg1 = m_leg * g * (p_leg1' * e2);
U_leg2 = m_leg * g * (p_leg2' * e2);
U_torso = m_torso * g * (p_torso' * e2);

U = simplify(U_leg1 + U_leg2 + U_torso);

U1 = double(subs(U, [q, dq], [conf1, qdot1]));
U2 = double(subs(U, [q, dq], [conf2, qdot2]));

q_act = [q1; q2];

[D, C, G, B] = LagrangianDynamics(T, U, q', dq', q_act);

D1 = double(subs(D, q', conf1'));
D2 = double(subs(D, q', conf2'));

C1 = double(subs(C, [q'; dq'], [conf1'; qdot1']));
C2 = double(subs(C, [q'; dq'], [conf2'; qdot2']));

G1 = double(subs(G, q', conf1.'));
G2 = double(subs(G, q', conf2.'));

B1 = double(subs(B, q', conf1.'));
B2 = double(subs(B, q', conf2.'));

% constrained dynamics

Cqdot1 = C1 * qdot1';
Cqdot2 = C2 * qdot2';

P1 = [D1, -Jst1'; Jst1, zeros(2,2)];
Q1 = [-Cqdot1 - G1; -Jstdot1 * qdot1'];

P2 = [D2, -Jst2'; Jst2, zeros(2,2)];
Q2 = [-Cqdot2 - G2; -Jstdot2 * qdot2'];

F_st1 = P1\Q1;
F_st2 = P2\Q2;

qddot1 = F_st1(1:5);
Fst1 = F_st1(6:7);

qddot2 = F_st2(1:5);
Fst2 = F_st2(6:7);

%% QUESTION 2

q_pre = [0.3827, 0.9239, 3.0107, 2.2253, 0.5236];
qdot_pre = [1.4782, -0.6123, 1.6, -1.6, 0];
```

```
p_swing = [x; y] + l_leg * [sin(q2 + q3); cos(q2 + q3)];

J_2 = double(subs(jacobian(p_swing, q), q, q_pre));
D_2 = double(subs(D, q', q_pre'));

P = [D_2, -J_2'; J_2, zeros(2,2)];
Q = [D_2 * qdot_pre'; zeros(2,1)];

sol = P\Q;

qdot_plus = sol(1:5);
F_delta = sol(6:7);
```