```
% Hopping leg with boom
% Author: Yanran Ding and Joao Ramos
% Last modified: 2020/04/02
clc
clear all
close all

global Torques
global counter
global Forces
global omegas
global Force_input


Torques = zeros(1000,2);
Forces = zeros(1000,2);
omegas = zeros(1000,2);
counter = 0;

addpath gen
addpath fcns


fprintf('------ ME446 Milestone 4 -------\n')
fprintf('Initializing ..............\n')

% --- parameters ---
p = get_params;     % Getting physical parameters of the robot
Nstep = 10;          % number of desired hops

% Initial condition
% q0 = [0; 0; pi/3.5; -pi/1.5]; %Joint angles for Knee forward
q0 = [0; 0; -pi/3; pi/2]; % Joint angles for Knee backward
dq0 = [0; 0; 0; 0];       %Joint velocities
ic = [q0; dq0];

%Ploting the robot in the initial configuration:
%plotRobot(ic,p);

% Recording
tstart = 0;
%tfinal = 2*Nstep;   %Maximum simulation time
tfinal = 2*Nstep;
tout = tstart;
Xout = ic';
Uout = [0,0];
Fout = [0,0];


Nh = 26.9;
Nk = 28.8;
Rw = 1.3;
```

```
kT = 0.0135;
kv = 0.0186;

for istep = 1:Nstep
    % aerial phase
    options = odeset('Events',@(t,X)event_touchDown(t,X,p));
    [t,X] = ode45(@(t,X)dyn_aerial(t,X,p),[tstart, tfinal],
 Xout(end,:),options);
    Fz = 1;

    p.tTD = t(end);                                 % touchdown time
    p.ptTD = fcn_p_toe(X(end,1:4),p.params);    % touchdown toe pos

    % log
    nt = length(t);
    tout = [tout; t(2:nt)];
    Xout = [Xout; X(2:nt,:)];
    [~,u,F] = dyn_aerial(t,X,p);
    Uout = [Uout;u(2:nt,:)];
    Fout = [Fout;F(2:nt,:)];
    tstart = tout(end);



    % Impact map (hard contact)
    X_prev = Xout(end,:);
    X_post = fcn_impactMap(X_prev,p);
    Xout(end,:) = X_post';


    if (istep == 8)
        time_single_hop = t(2:nt);

        X_8 = X(2:nt,:);
        U_8 = u(2:nt,:);

        theta_1 = X_8(:,1);
        theta1_dot_single_hop = X_8(:,5);
        theta_dot_hip = X_8(:,7);
        theta_dot_knee = X_8(:,8);

        voltage_hip = U_8(:,1)/(kT/Rw*Nh);
        voltage_knee = U_8(:,2)/(kT/Rw*Nk);

    end



    % stance phase
    options = odeset('Events',@(t,X)event_liftOff(t,X,p));
    [t,X] = ode45(@(t,X)dyn_stance(t,X,p),[tstart, tfinal],
 Xout(end,:), options);

    nt = length(t);
```

```
    tout = [tout; t(2:nt)];
    Xout = [Xout; X(2:nt,:)];
    [~,u,F] = dyn_stance(t,X,p);
    Uout = [Uout;u(2:nt,:)];
    Fout = [Fout;F(2:nt,:)];
    tstart = tout(end);

    fprintf('%d out of %d steps complete!\n',istep,Nstep)

    % Computing 8th hop for calculations for part 2b,d.
    % 8th hop was chosen since the system stabilizes at 8th hop.
    if (istep == 8)
        time_single_hop = [time_single_hop; t(2:nt)];

        X_8 = X(2:nt,:);

        U_8 = u(2:nt,:);

        theta_1 = [theta_1; X_8(:,1)];
        theta1_dot_single_hop = [theta1_dot_single_hop; X_8(:,5)];

        theta_dot_hip = [theta_dot_hip; X_8(:,7)];
        theta_dot_knee = [theta_dot_knee; X_8(:,8)];
        voltage_hip = [voltage_hip; U_8(:,1)/(kT/Rw*Nh)];
        voltage_knee = [voltage_knee; U_8(:,2)/(kT/Rw*Nk)];
    end
end
fprintf('Simulation Complete!\n')

% Visualing the motion
% [t,HIP] = animateRobot(tout,Xout,Uout,Fout,p);



% % Part 2
% LB = 0.5;
% time = tout;
% theta1_dot = Xout(:,5);
%
% plot_speed(time, theta1_dot);
%
% figure(3)
% plot(time_single_hop, theta1_dot_single_hop*LB);
% hold on;
% plot(time_single_hop, theta_dot_hip);
% plot(time_single_hop, theta_dot_knee);
% legend('theta_1','theta_hip','theta_knee');
%
% [avg_velocity, max_velocity] =
 compute_velocity(theta1_dot_single_hop);
% [power, avg_power] = compute_power(voltage_hip, voltage_knee,
 theta_dot_hip, theta_dot_knee);
% cost_transport = compute_cost_transport(theta_1,avg_power);
```

Published with MATLAB® R2020a