Three Link Biped Model

March 10, 2023

The Equations of Motion of the three link biped model are derived according to the hybrid dynamics note. The following code is supposed to run at least in MATLAB 2018b:

```
close all; clear;

% In this three link model, there are three links,
% with one being the body and two being the legs.
% The three links are inter-connected at the hip,
% with the other end of each either being head or foot.
% The stance leg is labeled link 1, and the swing leg is labeled link 2.
% Refer to Figure 3.5 in the westervelt2007feedback book.

syms g % Gravitational acceleration.
syms m % Mass of the leg.
syms M_H % Mass of the hip.
syms M_T % Mass of the torso.
syms r % Length of the leg..
syms l % Length of the torso.
syms q_1 % Angle from the torso to the stance leg, clockwise.
syms q_2 % Angle from the torso to the swing leg, clockwise.
syms q_3 % Angle from the vertical up direction to the torso, clockwise.
syms dq_1 % Time derivative of q_1.
syms dq_2 % Time derivative of q_2.
syms dq_3 % Time derivative of q_3.
syms th_1 % Angle from the vertical up direction to the stance leg, clockwise.
syms th_2 % Angle from the vertical up direction to the swing leg, clockwise.
syms th_3 % Angle from the vertical up direction to the torso, clockwise.
syms dth_1 % Time derivative of th_1.
syms dth_2 % Time derivative of th_2.
syms dth_3 % Time derivative of th_3.
syms p_1x % Horizontal position of the stance foot.
syms p_1y % Vertical position of the stance foot.
syms dp_1x % Time derivative of p_1x.
syms dp_1y % Time derivative of p_1y.
syms p_2x % Horizontal position of the swing foot.
syms p_2y % Vertical position of the swing foot.
syms dp_2x % Time derivative of p_2x.
syms dp_2y % Time derivative of p_2y.
syms tpof % The constant pi: (t)hree (p)oint (o)ne (f)our.
```

```
q_s = [q_1; q_2; q_3]; % Generalized coordinates.
dq_s = [dq_1; dq_2; dq_3]; % Time derivative of generalized coordinates.

q_e = [q_s; p_1x; p_1y]; % Extended generalized coordinates.
dq_e = [dq_s; dp_1x; dp_1y]; % Time derivative of extended generalized coordinates.

th_1 = q_1 + q_3 - tpof;
th_2 = q_2 + q_3 - tpof;
th_3 = q_3;
dth_1 = dq_1 + dq_3;
dth_2 = dq_2 + dq_3;
dth_3 = dq_3;

p_1 = [p_1x; p_1y];
p_1c = p_1 + [r/2 * sin(th_1); r/2 * cos(th_1)];
p_h = p_1 + [r * sin(th_1); r * cos(th_1)];
p_2c = p_h + [r/2 * sin(-th_2); - r/2 * cos(-th_2)];
p_2 = p_h + [r * sin(-th_2); - r * cos(-th_2)];
p_t = p_h + [l * sin(th_3); l * cos(th_3)];

dp_1c = jacobian(p_1c, q_e) * dq_e;
dp_h = jacobian(p_h, q_e) * dq_e;
dp_2c = jacobian(p_2c, q_e) * dq_e;
dp_t = jacobian(p_t, q_e) * dq_e;

% KE - Kinetic Energy, PE - Potential Energy.
% 1 - stance leg, 2 - swing leg, 3 - hip, 4 - torso.
KE1 = simplify(m / 2 * (dp_1c.') * dp_1c);
KE2 = simplify(m / 2 * (dp_2c.') * dp_2c);
KE3 = simplify(M_H / 2 * (dp_h.') * dp_h);
KE4 = simplify(M_T / 2 * (dp_t.') * dp_t);
KE = simplify(KE1 + KE2 + KE3 + KE4);
PE1 = simplify(m * g * p_1c(2));
PE2 = simplify(m * g * p_2c(2));
PE3 = simplify(M_H * g * p_h(2));
PE4 = simplify(M_T * g * p_t(2));
PE = PE1 + PE2 + PE3 + PE4;

% Stance phase.
D_s = simplify(jacobian(jacobian(KE, dq_s).', dq_s));
C_s = simplify(jacobian(D_s * dq_s, q_s) - jacobian(D_s * dq_s, q_s).' / 2);
G_s = simplify(jacobian(PE, q_s).');
B_s = [diag(ones(1, length(q_s)-1)); zeros(1, length(q_s)-1)];

% Impact phase.
D_e = simplify(jacobian(jacobian(KE, dq_e).', dq_e));
E = simplify(jacobian(p_2, q_e));
J_Ff = simplify(-(E / D_e * E.') \ E * jacobian(dq_e, dq_s)); % F_f / \dot{q}_s^-
J_qedot = simplify(jacobian(dq_e, dq_s) + D_e \ E.' * J_Ff); % \dot{q}_e^+ / \dot{q}_s^-
```

```
R = [0 1 0; 1 0 0; 0 0 1];
Delta_qs = R; %Delta_{q_s}
Delta_qsdot = [R zeros(3, 2)] * J_qedot; % Delta_{\dot{q}_s}
```