

---

```

function result = controll1()
% control vertical hopper

global h_axes body leg
global dt time x y xd yd
global hip_torque leg_angle body_angle leg_angled body_angled
global leg_state foot_x foot_y leg_lengthd leg_length rest_leg_length
global control_state height_desired leg_angle_desired last_bounce_time
global last_touchdown_time last_takeoff_time max_height
    last_max_height
global speed_desired

% control_state values
init = 0;
in_air = 1;
on_ground_going_down = 2;
on_ground_going_up = 3;

hip_air_k = 1000;
hip_air_b = 40;
hip_grnd_k = 25;
hip_grnd_b = 20;

leg_length_default = 0.5;

leg_length_gain = 0.0;

rest_leg_length = leg_length_default;
hip_torque = 0;

foot_y_new = y - rest_leg_length*cos( leg_angle );
leg_length_new = sqrt( (x - foot_x)^2 + (y - foot_y)^2 );

% initialization
if control_state == init
    control_state = in_air;
    result = control_state;
    return;
end;

if control_state == in_air
    if foot_y_new < 0
        last_touchdown_time = time;
        if yd <= 0
            control_state = on_ground_going_down;
        else
            control_state = on_ground_going_up;
        end;
        result = control_state;
        return;
    end;
    kxd = 0.02;

```

---

---

```

    if bitget( floor( time ), 1 )
        % See Raibert paper on how psi is defined
        leg_angle_desired = body_angle + asin( ( xd*last_bounce_time/2 +
kxd*(xd - speed_desired ) ) / leg_length_default);
    else
        leg_angle_desired = body_angle + asin( ( xd*last_bounce_time/2 +
kxd*(xd - speed_desired ) ) / leg_length_default);
    end;
    hip_torque = hip_air_k*(leg_angle - leg_angle_desired) + ...
                hip_air_b*leg_angle;

    if ( y > max_height )
        max_height = y;
    end;
    if ( yd < 0 )
        last_max_height = max_height;
    end;
end;

if control_state == on_ground_going_down
    if leg_length_new > rest_leg_length
        control_state = in_air;
        max_height = y;
        result = control_state;
        last_takeoff_time = time;
        return;
    end;
    if yd > 0
        control_state = on_ground_going_up;
        result = control_state;
        return;
    end;
    hip_torque = hip_grnd_k*(-body_angle) - hip_grnd_b*body_angle;
end;

if control_state == on_ground_going_up
    rest_leg_length = leg_length_default + 0.0185 +
leg_length_gain*(height_desired - last_max_height); %P-controller
with bias to default value and 2nd feedforward term to correct for
Ess
    if leg_length_new > rest_leg_length
        control_state = in_air;
        max_height = y;
        result = control_state;
        last_takeoff_time = time;
        if ( last_touchdown_time > 0 )
            last_bounce_time = last_takeoff_time - last_touchdown_time;
        end;
        return;
    end;
    if yd < 0
        control_state = on_ground_going_down;
        result = control_state;
        return;
    end;
end;

```

---

---

```
    hip_torque = hip_grnd_k*(-body_angle) - hip_grnd_b*body_angled;  
end;
```

*Published with MATLAB® R2019b*