```
function Tau_new = quad_constraints(q,dq, Tau)
%This function uses quadratic programming to find the point in the
%feasible set that minimizes ||(Tau_new - Tau)||

%Parameters
L_H = 0.096;
L_K = 0.155;
D_K = 0.052;
L = sqrt(L_K^2+D_K^2);
mu = 0.6;
tau_Imax = 0.162;
w_NL = 645.2;
tau_stall = 0.124;
N_H = 26.9;
N_K = 28.8;

theta3 = q(3);
theta4 = q(4);
w_H = dq(3);
w_K = dq(4);
%Foot Jacobian w.r.t the hip
J_HIP =  [L_H*cos(theta3)+L*cos(theta3+theta4)
 L*cos(theta3+theta4); ...
    L_H*sin(theta3)+L*sin(theta3+theta4) L*sin(theta3+theta4)];

F_old = (J_HIP')^-1*Tau;

%Defining parameters for quadprog
H = [2 0; 0 2];
f = -2*[Tau(1) Tau(2)];

J_inverse_transpose = (J_HIP')^-1;
J_row_1 = [J_inverse_transpose(1,:) ; 0 0];
H = H + 2*J_row_1.'*J_row_1;
c = -2*F_old(1)*J_inverse_transpose(1,:);
f = f + c;

%Constraints
hip_motor_constraint = 1/N_H*[1 0; -1 0; 1 0; -1 0];
knee_motor_constraint = 1/N_K*[0 1; 0 -1; 0 1; 0 -1];

contact_constraint = [1 mu; -1 mu;0 1]*(J_HIP')^-1;

hori_force_sign = -F_old(1)*[1 0]*(J_HIP')^-1;

A = [hip_motor_constraint; knee_motor_constraint; contact_constraint];
%A = [hip_motor_constraint; knee_motor_constraint; contact_constraint;
 hori_force_sign];
%A = [hip_motor_constraint; knee_motor_constraint];
%A = [contact_constraint];

hip_limit = [tau_Imax; tau_Imax; tau_stall*(1 - N_H * w_H / w_NL); ...
```

```
                        tau_stall*(1 + N_H * w_H / w_NL)];

knee_limit = [tau_Imax; tau_Imax; tau_stall*(1 - N_K * w_K /
 w_NL); ...
                        tau_stall*(1 + N_K * w_K / w_NL)];

force_limit = [0;0;0];

hori_limit = 0;

b = [hip_limit;knee_limit;force_limit];
%b = [hip_limit;knee_limit;force_limit; hori_limit];
%b = [hip_limit;knee_limit];
%b = [force_limit];

options = optimoptions('quadprog','Display','off');
Tau_new = quadprog(H,f,A,b,[],[],[],[],[],options);

if length(Tau_new) == 0
    Tau_new = [0;0];
end


end
```

*Published with MATLAB® R2019b*