```
function [T1, T2, T3, err_Fx_3, err_Fy_3, err_T3] = HW06_Function(theta1,theta2, ...
    thetadot1, thetadot2,thetadotdot1,thetadotdot2,GRF_hori,GRF_vert)
```

Loading constants and other necessary variables

```
k = matfile('kinematic_data.mat');
c = matfile('constants.mat');
h = c.h;
g = c.g;
I1 = c.I1;
I2 = c.I2;
I3 = c.I3;
L1 = c.L1;
L2 = c.L2;
L3 = c.L3;
m1 = c.m1;
m2 = c.m2;
m3 = c.m3;
r1 = c.r1;
r2 = c.r2;
r3 = c.r3;
a = k.a;
theta3 = k.theta3;
thetadot3 = k.thetadot3;
thetadotdot3 = k.thetadotdot3;
```

Computation of Net Joint Torque at Ankle

```
T1 = (h*GRF_hori) - (GRF_vert.*a);
```

Necessary force and dimension computations

```
Fx_2 = GRF_hori + (m1*r1.*((sin(theta1).*thetadotdot1) + ...
    (cos(theta1).*thetadot1.^2)));
Fy_2 = GRF_vert - (m1*g) - (m1*r1.*((cos(theta1).*thetadotdot1) - ...
    (sin(theta1).*thetadot1.^2)));
d1 = L1 - r1;
```

Computation of Net Joint Torque at Knee

```
T2 = T1 + (GRF_hori.*r1.*sin(theta1)) - (GRF_vert.*r1.*cos(theta1))...
    + (Fx_2*d1.*sin(theta1)) - (Fy_2*d1.*cos(theta1)) - I1.*thetadotdot1;
```

Necessary force and dimension computations

```
Fx_3 = Fx_2 + (m2*r2.*((sin(theta2).*thetadotdot2) + (cos(theta2).*thetadot2.^2))) ...
    + (m2*L1.*((sin(theta1).*thetadotdot1) + (cos(theta1).*thetadot1.^2)));
Fy_3 = Fy_2 - (m2*r2.*((cos(theta2).*thetadotdot2)- (sin(theta2).*thetadot2.^2))) ...
```

```
    - (m2*L1.*((cos(theta1).*thetadotdot1) - (sin(theta1).*thetadot1.^2)));
d2 = L2 - r2;
```

Computation of Net Joint Torque at Hip

```
T3 = T2 + (Fx_2.*r2.*sin(theta2)) - (Fy_2.*r2.*cos(theta2))...
    + (Fx_3.*d2.*sin(theta2)) - (Fy_3.*d2.*cos(theta2)) - I2.*thetadotdot2;
```

Hand of God computations

Necessary force computations

```
%Fx_3_hg = -m3*L1.*((sin(theta1).*thetadotdot1) + (cos(theta1).*thetadot1.^2)) ...
   % -m3*L2.*((sin(theta2).*thetadotdot2) + (cos(theta2).*thetadot2.^2)) ...
    %-m3*r3.*((sin(theta3).*thetadotdot3) + (cos(theta3).*thetadot3.^2));

Fx_3_hg =  -m3*r3*(sin(theta3).*thetadotdot3 + cos(theta3).*thetadot3.^2);

%Fy_3_hg = m3*g + m3*L1.*((cos(theta1).*thetadotdot1) - (sin(theta1).*thetadot1.^2)) .
    %+m3*L2.*((cos(theta2).*thetadotdot2) - (sin(theta2).*thetadot2.^2)) ...
    %+m3*r3.*((cos(theta3).*thetadotdot3) - (sin(theta3).*thetadot3.^2));

 Fy_3_hg = m3*g + m3*r3*(cos(theta3).*thetadotdot3 - sin(theta3).*thetadot3.^2);
```

"Hand of God": Computation of Net Joint Torque at Hip

```
T3_hg = (r3*Fy_3_hg.*cos(theta3)) - (r3*Fx_3_hg.*sin(theta3)) + I3.*thetadotdot3;
```

Error calculations

```
err_Fx_3 = Fx_3_hg - Fx_3;
err_Fy_3 = Fy_3_hg - Fy_3;
err_T3 = T3_hg - T3;

end
```