Matlab Code

The code for analyzing our system is composed of several functions and allows for both static and dynamic analysis. The Code is comprised of six functions, these functions are taken advantage of by two other scripts to provide the appropriate plots.

ForceAnalysis

This function takes the matrix equations we derived for the forces and implements them in Matlab. The function returns the state vector which includes the reaction forces at the joints and the moments on said joints.

Before the matrices we have other necessary calculations.

-The friction force depends on P and the friction coefficient.

-The angle phi should be returned by the motor in the final controller, however for a static analysis we may simply provide the desired angles.

-The inertial forces are only present during motion, we are given the choice whether to include them or not.

```
function M = ForceAnalysis(G,lcm,l,n,P)
    % Friction force
    F = 0.2*P(1);       %[N]
    % Input from servo encoder
    phi = [45 45];     %[deg]

    % For static configurations B = [0,0]
    if n == 1
        B = InertialForce(m,l,lcm,phi);
    end
    if n ~= 1
        B = [0 0];
    end

    A = [1 0 -1 0 0 0;
        0 -1 0 1 0 0;
        0 0 sin(phi(1))*l(1) cos(phi(1))*l(1) -1 1;
        0 0 1 0 0 0;
        0 0 0 1 0 0;
        0 0 0 0 0 1];

    C = [-B(1)*sin(phi(1));
        -G(1) + B(1)*cos(phi(1));
        (B(1) - G(1)*cos(phi(1)))*lcm(1);
        -B(2)*sin(phi(2)) + P(3)*sin(phi(2)) - F*cos(phi(2));
        P(3)*cos(phi(2)) + G(2) - B(2)*cos(phi(2)) + F*sin(phi(2));
        P(3)*l(2) - B(2)*lcm(2) + G(2)*cos(phi(2))*lcm(2)];
    %In our calculations, we assumed the force of gravity to be the wrong
    %way, thus we changed the sign for the simulation.

    M = A\C;    %[R01x; R01y; R12x; R12y; Mr1; Mr2]

end
```

InertialForce

Here we see our equations from the kinematic analysis come to play.

The angular velocity and acceleration should all be inputs from the servo encoder. The angle also should be an input from the servo encoder, however it is provided by the function ForceAnalysis.

First we calculate the acceleration for the center of mass, before acquiring the inertial forces. This function is only called when the mechanical hand is in motion.

```
function B = InertialForce(m,l,lcm,phi)

    % Inputs from servo encoder
    omg = [1 1];
    eps = [0.1 0.1];

    acm = Acceleration(l,lcm,phi,omg,eps);

    B(1) = m(1)*acm(1)*-1;
    B(2) = m(2)*acm(2)*-1;

end
```

Acceleration/Velocity

These are the kinematic equations.

```
function v = Velocity(l, li, phi, omg)
    v = [omg(1)*li(1) sqrt( (omg(1)*l(1))^2 + (omg(2)*li(2))^2 +
    2*omg(1)*omg(2)*l(1)*li(2)*cos(phi(1)-phi(2)))];
end

function a = Acceleration(l, li, phi, omg, eps )
    a = [li(1)*sqrt(eps(1)^2 + omg(1)^4) sqrt( (eps(1)*l(1))^2 +
    (omg(1)^2*l(1))^2 + (eps(2)*li(2))^2 + (omg(2)^2*li(2))^2 +
    (eps(1)*l(1)*eps(2)*li(2)  + omg(1)^2*l(1)*omg(2)^2*li(2))*(cos(phi(1)-phi(2))) +
    (omg(1)^2*l(1)*eps(2)*li(2) + eps(1)*l(1)*omg(2)^2*li(2))*(sin(phi(1)-phi(2))) )];
end
```

PressingForce

The weight, vertical and horizontal accelerations of the manipulated object are considered here. The function returns the minimum force required to hold the object under certain conditions.

```
function P = PressingForce(m,up_a,horz_a)
    F = m*(9.81+up_a);        %[N]
    Ff = F/4;                 %Friction forces are distributed onto four fingers

    P(1) = Ff/0.2;
    P(2) = m*horz_a/2;        %Horizontal forces are distributed onto two fingers
    P(3) = sqrt(P(1)^2 + P(2)^2);     %Force generated per finger [N]
end
```

GearRatio

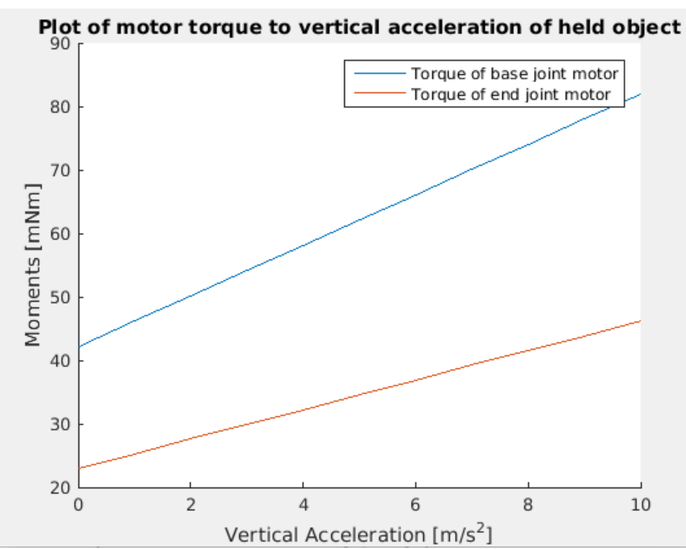
The ratio between the worm and worm gear will have a significant impact on the required torque of the motor.

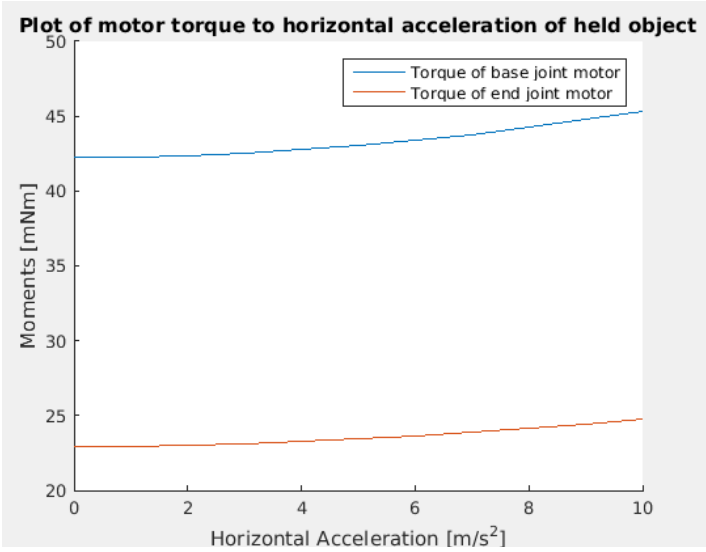
```
function Mw = GearRatio(M)
    pitch = 0.0025;
    d_gear = 0.015;
    Mw = (M * pitch)/(2*3.14*d_gear);
end
```

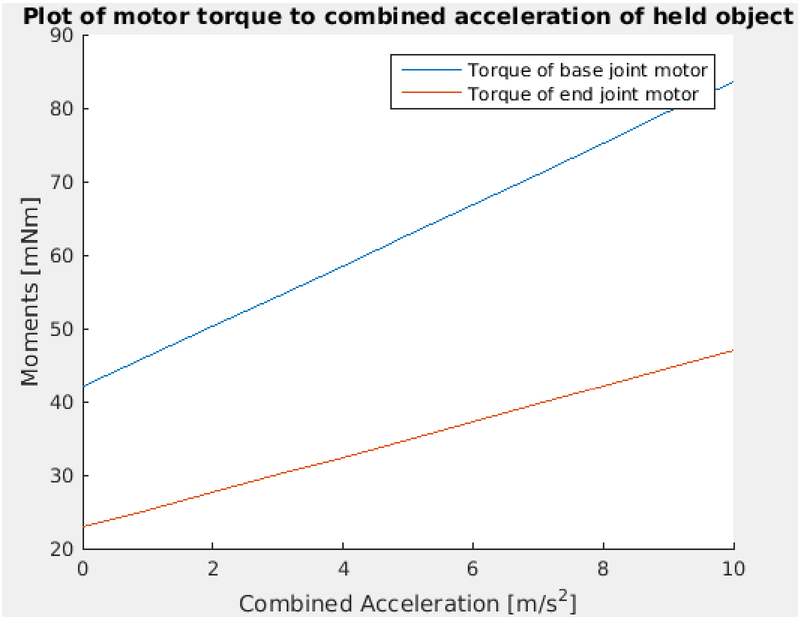
The following script provides us with the force analysis when the hand is under the greatest strain(while in motion). We consider three case; vertical, horizontal and combined accelerations.

```
% Const values for our configuration
m = [0.4 0.02];                 %[kg]
lcm = [0.05 0.06];              %[m]
l = [0.1 0.14];                 %[m]

for i = 0:10
    P = PressingForce(0.5,i,i);      % The analysis returns values
    M = ForceAnalysis(m,lcm,l,1,P); % combined for all fingers
    M1(i+1) = GearRatio(M(5));
    M2(i+1) = GearRatio(M(6));
end

figure;
hold on;
x = 0:1:10;
plot(x,M1*1000)
plot(x,M2*1000)
title('Plot of motor torque to combined acceleration of held object')
xlabel('Combined Acceleration [m/s^2]')
ylabel('Moments [mNm]')
legend('Torque of base joint motor','Torque of end joint motor')
```

The results when both joints are at 45deg angles are as follows:







We may see that the system is under the largest strain when the system has an upward acceleration. The horizontal acceleration has a very small impact on the system, most of the torque generated is to counter the weight of the held object.

We may also notice that the required torque of the motors are well below their actual specifications. This means we can either choose a heavier object to carry or further increase the acceleration. Keeping in mind however that the torque still needs to be overcompensated in order ensure a proper friction grip on the object.

The second script below provides us with the plots that correspond to the kinematic analysis.

```
l = [0.1 0.14];             %[m]
% Starting values
initphi = [0 5];
initomg = [0 0.1];
eps = [115 115];
i = 0;
for t = 0:0.1:1
    i = i + 1;
    omg = initomg + eps * t;
    phi = initphi + omg * t + 0.5 * eps * t^2;

    omg1(i) = omg(1);
    omg2(i) = omg(2);
    phi1(i) = phi(1);
    phi2(i) = phi(2);
    % Angular to linear velocity
    v = Velocity(l,l,phi,omg);
    v1(i) = v(1);
    v2(i) = v(2);
    % Angular to linear acceleration
    a = Acceleration(l,l,phi, omg, eps);
    a1(i) = a(1);
    a2(i) = a(2);
end
t = 0:0.1:1;
figure(1); hold on; plot(t,phi1); plot(t,phi2); title('Change of angle');
xlabel('Time [s]'); ylabel('Change in angle [deg]');
legend('Point B','Point C');

figure(2); subplot(1,2,1); hold on; plot(t,omg1); plot(t,omg2); title('Angular velocity plot');
xlabel('Time [s]'); ylabel('Change in angular velocity [deg/s]');
legend('Point B','Point C');
subplot(1,2,2); hold on; plot(t,v1); plot(t,v2); title('Linear velocity plot');
xlabel('Time [s]'); ylabel('Change in linear velocity [m/s]');
legend('Point B','Point C');

figure(3); hold on; plot(t,a1); plot(t,a2); title('Linear acceleration plot');
xlabel('Time [s]'); ylabel('Change in linear acceleration [m/s^2]');
legend('Point B','Point C');
```

Before taking an actual look at the plots, we need to calculate the maximum rpm of the motor with the worm gear. This value needs to be compared with plots to ensure we are within working range of our parts.

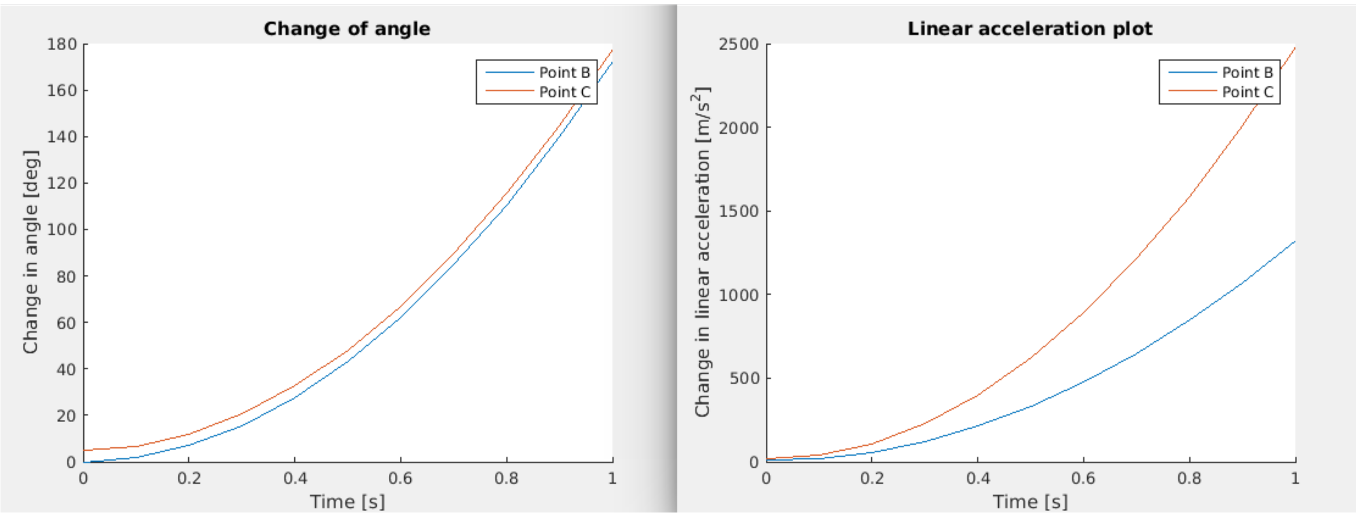
$$\omega_{out} = \omega_{motor} / n \quad \text{, where n is the number of teeth on the gear}$$

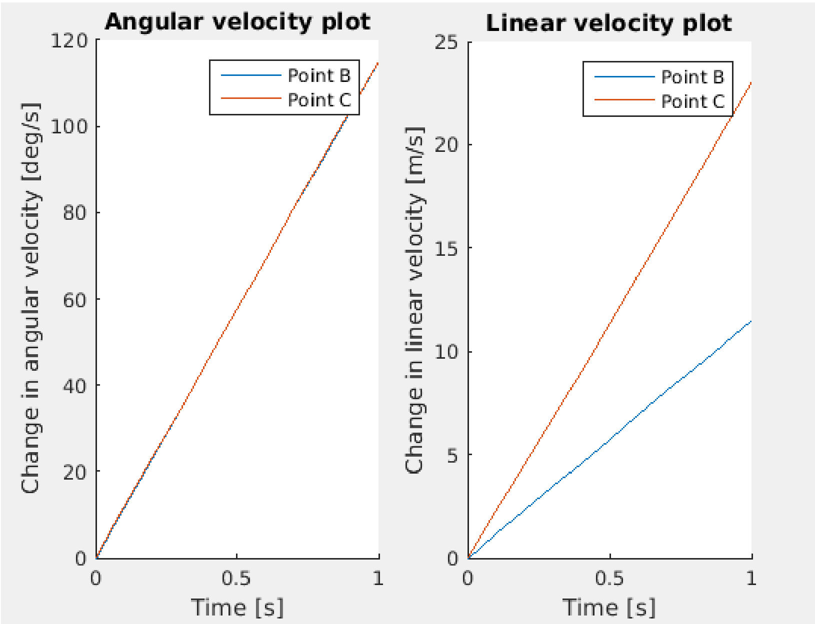
This gives us the following:

$$\omega_{out\,(base\,joint)} = \omega_{motor\,max\,base} / n = \frac{9700\,rpm}{20} = 485[rpm] = 2910[deg/s]$$

$$\omega_{out\,(end\,joint)} = \omega_{motor\,max\,end} / n = \frac{8300\,rpm}{20} = 415[rpm] = 2490[deg/s]$$

Now we may now take a look at our plots.





Points B and C are as they were in the kinematic analysis in the beginning of the report. All of the above plots are for the same system with the same initial conditions in the same time period of 1s.

The first plot shows how the angles of each joint changes over time. The working range of each joint is 180 degrees, which is completed in 1 second in this case. One of the angles is offset by 5 degrees to better see both of them on the plot.

The second plot demonstrates the linear accelerations of the end points of the joints.

The final two plots are the angular and linear velocity plots. Since we chose the same initial conditions for both joints and the same angular accelerations, their angular velocities are identical. We may also notice that the maximum 120 deg/s is very small compared to our motors maximum speed that was calculated above. This excess in angular velocity is never the less necessary as it may be significantly lowered under different loads.

According to the provided documentation, the angular accelerations are also well in range.

These analysis prove that our motors are well equipped to handle the performance required. If this was an iterative process, further inspection for smaller motors would be a good idea as the current ones may be unnecessarily over compensated.