AGH University of Science and Technology

Faculty of Mechanical Engineering and Robotics

Fundamentals of Design of Mechanisms in Mechatronic Devices

Project

Mechanical Hand

Year: II, Mechatronic Engineering with English as instruction language

Project Overview

The goal of the project was to design a mechanical hand capable of picking up objects of various shapes and sizes.

- -The hand was to be made up of four fingers with two degrees of freedom each.
- -Each finger pair has their base joint connected to one shaft which will serve as the main compression force for holding a given object.
- -The four finger tips all move independently of each other to allow them to conform to a desired shape.
- -The gripping mechanism is friction based.

The report first shows the full kineostatic analysis of one mechanical finger. All the calculations; including velocity, acceleration and force calculations, were completed using graphical methods. The results were then transformed to matrix form to allow for computer aided computations.

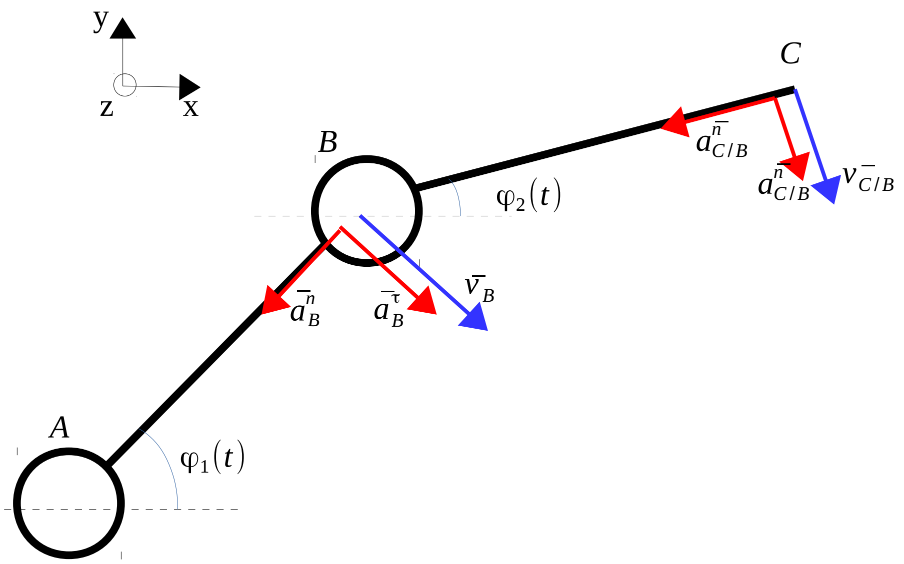
Since the four mechanical fingers are symmetrical, one analysis was sufficient to describe the entire system.

Using Matlab and the derived matrix equations, we estimated the parameters of our hand and chose the appropriate parts; motors, bearings, retaining rings. Now we had sufficient information to begin producing a 3D model using Autodesk's Inventor.

The exact specifications of the mechanical hand were calculated from the Inventor software and recalculated in Matlab. Our assumptions were sufficient and the limitations of the hand were found.

Mechanical Analysis

Velocity and Acceleration of Mechanical Finger



$$\begin{cases} V_B = \omega_1(t) l_{AB} \\ V_{C/B} = \omega_2(t) l_{BC} \end{cases} \quad \begin{cases} a_B^{\tau} = \varepsilon_1(t) l_{AB} \\ a_B^{n} = \omega_1^2(t) l_{AB} \end{cases} \quad \begin{cases} a_{C/B}^{\tau} = \varepsilon_2(t) l_{BC} \\ a_{C/B}^{n} = \omega_2^2(t) l_{BC} \end{cases}$$

Velocity:

$$\begin{cases} V_C^x = V_B^x + V_{C/B}^x \\ V_C^y = V_B^y + V_{C/B}^y \end{cases}$$

$$\begin{cases} V_C^x = V_B \sin\varphi_1(t) + V_{C/B} \sin\varphi_2(t) \\ V_C^y = V_B \cos\varphi_1(t) + V_{C/B} \cos\varphi_2(t) \end{cases}$$

$$V_c = \sqrt{(V_c^x)^2 + (V_c^y)^2} =$$

$$= \sqrt{V_B^2[\sin\varphi_1^2(t) + \cos\varphi_1^2(t)] + 2V_B V_{C/B}[\sin\varphi_1(t)\sin\varphi_2(t) + \cos\varphi_1(t)\cos\varphi_2(t)] + V_{C/B}^2[\sin\varphi_2^2(t) + \cos\varphi_2^2(t)]}$$

$$= \sqrt{V_B^2 + 2V_B V_{C/B}\cos(\varphi_1(t) - \varphi_2(t)) + V_{C/B}^2}$$

Acceleration:

$$\begin{cases} a_C^x = (a_B^\tau)^x + (a_B^n)^x + (a_{C/B}^\tau)^x + (a_{C/B}^n)^x \\ a_C^y = (a_B^\tau)^y + (a_B^n)^y + (a_{C/B}^\tau)^y + (a_{C/B}^n)^y \end{cases}$$

$$\begin{cases} a_C^x = a_B^\tau \sin\varphi_1(t) - a_B^n \cos\varphi_1(t) + a_{C/B}^\tau \sin\varphi_2(t) - a_{C/B}^n \cos\varphi_2(t) \\ a_C^y = a_B^\tau \cos\varphi_1(t) + a_B^n \sin\varphi_1(t) + a_{C/B}^\tau \cos\varphi_2(t) + a_{C/B}^n \sin\varphi_2(t) \end{cases}$$

$$a_c = \sqrt{(a_c^x)^2 + (a_c^y)^2} =$$

$$= \Big[[(a_B^\tau)^2 + (a_B^n)^2 + (a_{C/B}^\tau)^2 + (a_{C/B}^n)^2][\sin\varphi_1^2(t) + \cos\varphi_1^2(t)] + (a_B^\tau)(a_{C/B}^\tau)[\cos(\varphi_1(t) - \varphi_2(t))] +$$
$$+ (a_B^\tau)(a_{C/B}^n)[\sin(\varphi_2(t) - \varphi_1(t))] + (a_B^n)(a_{C/B}^\tau)[\sin(\varphi_1(t) - \varphi_2(t))] + (a_B^n)(a_{C/B}^n)[\cos(\varphi_1(t) - \varphi_2(t))]\Big]^{\frac{1}{2}}$$

$$= \Big[[(a_B^\tau)^2 + (a_B^n)^2 + (a_{C/B}^\tau)^2 + (a_{C/B}^n)^2] + [a_B^\tau a_{C/B}^\tau + a_B^n a_{C/B}^n][\cos(\varphi_1(t) - \varphi_2(t))] +$$
$$+ [a_B^n a_{C/B}^\tau - a_B^\tau a_{C/B}^n][\sin(\varphi_1(t) - \varphi_2(t))]\Big]^{\frac{1}{2}}$$

Conclusion

After substituting the linear velocities and accelerations with their angular variants, we obtain the following.

For any point from A to B (change l_{AB} accordingly):

$$V_B = \omega_1 l_{AB}$$

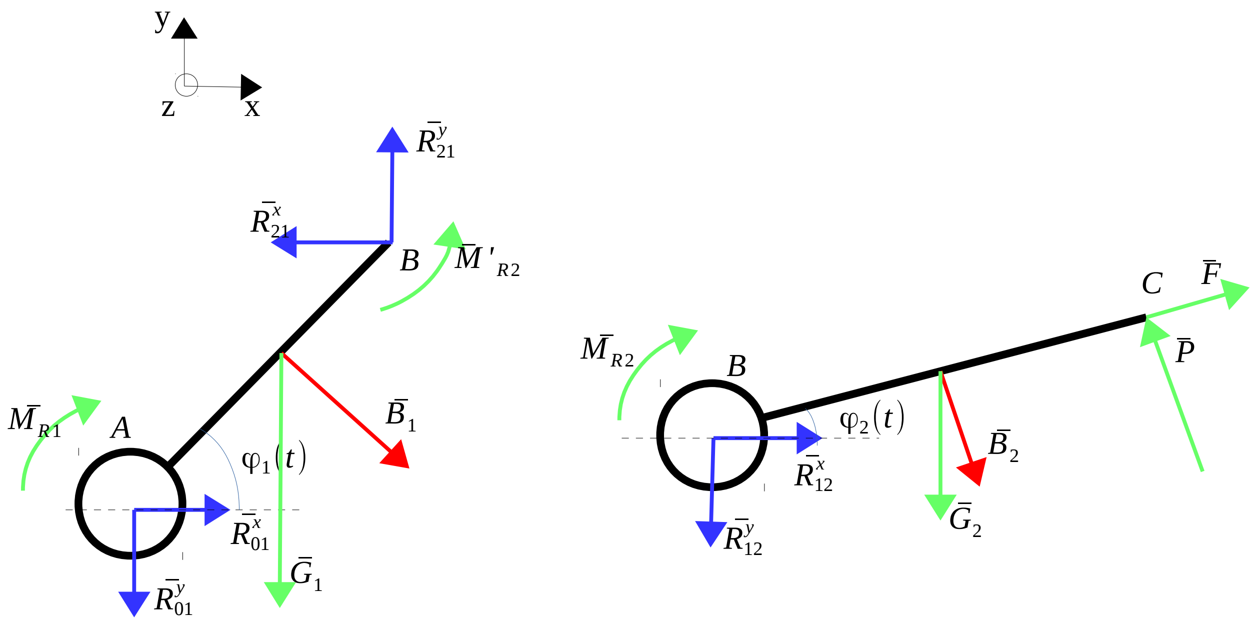
$$a_B = l_{AB}\sqrt{\varepsilon_1^2 + \omega_1^4}$$

For any point from B to C (change l_{BC} accordingly) :

$$V_c = \sqrt{(\omega_1 l_{AB})^2 + 2\omega_1 l_{AB}\omega_2 l_{BC}\cos(\varphi_1(t) - \varphi_2(t)) + (\omega_2 l_{BC})^2}$$

$$a_c = \Big[[(\varepsilon_1 l_{AB})^2 + (\omega_1^2 l_{AB})^2 + (\varepsilon_2 l_{BC})^2 + (\omega_2^2 l_{CB})^2] + [\varepsilon_1 l_{AB}\varepsilon_2 l_{BC} + \omega_1^2 l_{AB}\omega_2^2 l_{BC}][\cos(\varphi_1(t) - \varphi_2(t))] +$$
$$+ [\omega_1^2 l_{AB}\varepsilon_2 l_{BC} - \varepsilon_1 l_{AB}\omega_2^2 l_{BC}][\sin(\varphi_1(t) - \varphi_2(t))]\Big]^{\frac{1}{2}}$$

Force Analysis of Mechanical Finger



For the first element:

$$\Sigma F_x:\ R_{01}^x + B_1 \sin\varphi_1(t) - R_{21}^x = 0$$

$$\Sigma F_y:\ -R_{01}^y - G_1 - B_1 \cos\varphi_1(t) + R_{21}^y = 0$$

$$\Sigma M_{(A)}: -M_{R1} - B_1 l_{cm1} - G_1 \cos\varphi_1(t) l_{cm1} + R_{21}^x \sin\varphi_1(t) l_{AB} + R_{21}^y \sin\varphi_1(t) l_{AB} + M'_{R2} = 0$$

For the second element:

$$\Sigma F_x: R_{12}^x + B_2 \sin\varphi_2(t) - P \sin\varphi_2(t) + Fcos\,\varphi_2(t) = 0$$

$$\Sigma F_y: -R_{12}^y - G_2 - B_2 \cos\varphi_2(t) + P\cos\varphi_2(t) + F\sin\varphi_2(t) = 0$$

$$\Sigma M_{(A)}:\ -M_{R2} - B_2 l_{cm2} - G_2 \cos\varphi_2(t) l_{cm2} + P\, l_{BC} = 0$$

We have six unknowns ($R_{01}^x, R_{01}^y, R_{12}^x = R_{21}^x, R_{12}^y = R_{21}^y, M_{R1,} M_{R2} = M'_{R2}$) and six equations.

The equations need to be prepared for matrix form:

$$\begin{cases} R_{01}^x - R_{21}^x = -B_1 \sin\varphi_1(t) \\ -R_{01}^y + R_{21}^y = G_1 + B_1 \cos\varphi_1(t) \\ -M_{R1} + R_{21}^x \sin\varphi_1(t) l_{AB} + R_{21}^y \sin\varphi_1(t) l_{AB} + M_{r2} = B_1 l_{cm1} + G_1 \cos\varphi_1(t) l_{cm1} \\ R_{12}^x = -B_2 \sin\varphi_2(t) + P \sin\varphi_2(t) - F\cos\varphi_2(t) \\ R_{12}^y = P\cos\varphi_2(t) - G_2 - B_2 \cos\varphi_2(t) + F\sin\varphi_2(t) \\ M_{R2} = P\, l_{BC} - B_2 l_{cm2} - G_2 \cos\varphi_2(t) l_{cm2} \end{cases}$$

The final matrix form is as follows:

$$\begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & \sin\varphi_1(t) l_{AB} & \sin\varphi_1(t) l_{AB} & -1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{01}^x \\ R_{01}^y \\ R_{12}^x \\ R_{12}^y \\ M_{R1} \\ M_{R2} \end{bmatrix} = \begin{bmatrix} -B_1 \sin\varphi_1(t) \\ G_1 + B_1 \cos\varphi_1(t) \\ B_1 l_{cm1} + G_1 \cos\varphi_1(t) l_{cm1} \\ -B_2 \sin\varphi_2(t) + P \sin\varphi_2(t) - F \cos\varphi_2(t) \\ P \cos\varphi_2(t) - G_2 - B_2 \cos\varphi_2(t) + F \sin\varphi_2(t) \\ P l_{BC} - B_2 l_{cm2} - G_2 \cos\varphi_2(t) l_{cm2} \end{bmatrix}$$

The above system is now mathematically described and computer aided design may be implemented.

The angular positions, velocities and accelerations need to be read from the servo encoders in real time and fed into the matrix to apply proper control. Since our design lacks sensors, the forces P need to be determined before hand, based on the manipulated objects. This makes the system less robust and limited to picking up and moving only one object, say on a production line.

The reaction forces are calculated to properly obtain the moments of the motors. These moments are the counter balance moments, or torques, that need to be created and sustained by the motors to ensure proper functioning.

Without pressure sensors on the finger tips, designing any feedback controller for the hand is not feasible. Therefore we are forced to work with an open loop system that requires very precise calibration. It is imperative for the mathematical description of the system and its parameters to be as accurate as possible.

An immediate flaw of our analysis is the lack of friction forces working between the joints. The pressing force, and therefore the motor torque, needs to be overcompensated to ensure a proper grip.

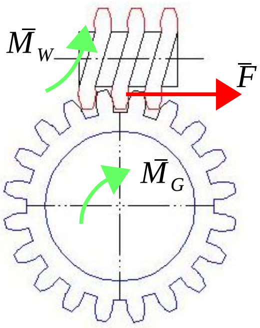
Assumptions

The friction(F) and pressing(P) forces we see above are what hold our object in place. As mentioned, these need to be predetermined depending on the weight and size of the manipulated object. Therefore, the first assumptions that need to be made are its parameters. Let us assume our hand will be picking up a cube of a weight of 0.5kg and an appropriate size that depends on the length of the fingers.

We also need to assume the motion of the hand. If the hand is to simply pick up the object and place it down somewhere else we can limit its movement to one plane of motion. The hand then will be subjected to only four accelerations; upward, downward, left and right. The analysis for both horizontal accelerations are symmetrical, therefore they can be reduced to the analysis of only one direction. While the upward acceleration will be much more demanding on the hand than the downward acceleration (total acceleration = gravity +/- relative acceleration), therefore the downward acceleration may be omitted. This leaves us with only two necessary analysis:

- -Vertical upward acceleration
- -Horizontal acceleration

The next assumption that needs to be made is the driving method. The base of the fingers will be driven in pairs with a worm gear, while the finger tips will be driven independently also by worm gears. The torque required to produce a given moment on the finger is as follows:



$$\begin{cases} P_{M_w} = 2\pi M_W \\ P_F = Fp \end{cases} \text{,gives} \quad 2\pi M_W = pF$$

$$\begin{cases} 2\pi M_W = pF \\ M_G = Fd \end{cases}$$

$$M_W = \frac{M_G p}{2\pi d} \quad \text{, where} \quad \begin{matrix} p - pitch \\ d - diameter\ of\ gear \end{matrix}$$

These calculations are only an approximation as friction was omitted.

The final assumption to be made is the friction coefficient of the finger tip grip and object.

With the provided information, we are capable of preparing the Matlab code for our simulation and get a basic idea of the required motors and parts needed for the project. To not repeat ourselves, the Matlab code is provided at the end of the report. The only difference are the system parameters, now we only have assumptions to create a working prototype, were as the provided code contains the final system parameters for an accurate final analysis.

Part Selection

For the mechanism elements which transmit torque we selected a standard steel from the Autodesk Material Library to guarantee that they will have enough strength. This includes all shafts, gears, keys and retaining rings.

The frame, bushings, cover plates and servomotor mounts were designed to be light weight but still rigid. A reasonable choice of material was 6061 aluminum alloy. As these parts are none standard, production of these parts should be easy and inexpensive. Most parts that will be made from aluminum plates are either die pressed, cut and/or bent into shape.

The bolted connections (bolts, nuts and washers) are composed of standardized elements.

For the bearings, rough calculation-based assumptions lead to the selection of SKF W 638/4 XR-2Z deep groove ball bearings. They are located in the place of contact between any bodies rotating relative to each other.

As for the motors, the end joints are powered by 3564_B brushless DC servomotor and the base joint by 4490_B brushless DC servomotor. Their large stall torques makes them adequate for the job.

The end joints function only to obtain a good positioning grip, while the larger base joints will provide the necessary torque to attain a sustainable grip. The end joints will also position themselves in order to have a mechanical advantage by acquiring such an angle that the pressing force is transferred to the joint shaft as much as possible. The worm gear also provides an additional locking mechanism.

In the table below we placed selected standardized elements

Element	**Source**
4490_B brushless DC servomotor	https://fmcc.faulhaber.com/resources/img/EN_4490_B_FMM.PDF

Brushless DC-Servomotors 190 mNm

2 Pole Technology 232 W

Series 4490 ... B

Values at 22°C and nominal voltage		4490 H	024 B	036 B	048 B	
1	Nominal voltage	U_N	24	36	48	V
2	Terminal resistance, phase-phase	R	0,22	0,44	0,7	Ω
3	Efficiency, max.	$\eta_{max.}$	87	87	87	%
4	No-load speed	n_0	9 700	10 400	10 800	min^{-1}
5	No-load current, typ. (with shaft ø 6 mm)	I_0	0,527	0,397	0,317	A
6	Stall torque	M_H	2 635	2 760	2 978	mNm
7	Friction torque, static	C_0	4,96	4,96	4,96	mNm
8	Friction torque, dynamic	C_V	$7{,}72 \cdot 10^{-4}$	$7{,}72 \cdot 10^{-4}$	$7{,}72 \cdot 10^{-4}$	mNm/min^{-1}
9	Speed constant	k_n	395	283	220	min^{-1}/V
10	Back-EMF constant	k_E	2,53	3,54	4,56	mV/min^{-1}
11	Torque constant	k_M	24,2	33,8	43,5	mNm/A
12	Current constant	k_I	0,041	0,03	0,023	A/mNm
13	Slope of n-M curve	$\Delta n/\Delta M$	3,6	3,7	3,5	min^{-1}/mNm
14	Terminal inductance, phase-phase	L	73	142	235	µH
15	Mechanical time constant	τ_m	4,9	5	4,8	ms
16	Rotor inertia	J	130	130	130	gcm^2
17	Angular acceleration	$\alpha_{max.}$	203	212	229	$\cdot 10^3 rad/s^2$
18	Thermal resistance	R_{th1} / R_{th2}	0,96 / 3,9			K/W
19	Thermal time constant	τ_{w1} / τ_{w2}	23 / 1 222			s
20	Operating temperature range:					
	– motor		-30 ... +125			°C
	– winding, max. permissible		+125			°C
21	Shaft bearings		ball bearings, preloaded			

Element	Source
3564_B brushless DC servomotor	https://fmcc.faulhaber.com/resources/img/EN_3564_B_FMM.PDF

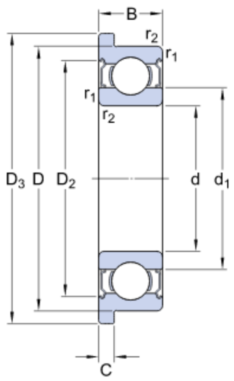
Brushless DC-Servomotors — 66 mNm

2 Pole Technology — 126 W

Series 3564 ... B

Values at 22°C and nominal voltage		3564 K		012 B	024 B	036 B	048 B	
1	Nominal voltage	U_N		12	24	36	48	V
2	Terminal resistance, phase-phase	R		0,56	1,1	2,61	4,1	Ω
3	Efficiency, max.	$\eta_{max.}$		82	83	83	83	%
4	No-load speed	n_0		8 300	11 500	11 600	12 800	min^{-1}
5	No-load current, typ. (with shaft ø 4 mm)	I_0		0,198	0,166	0,112	0,099	A
6	Stall torque	M_H		293	432	408	418	mNm
7	Friction torque, static	C_0		1,2	1,2	1,2	1,2	mNm
8	Friction torque, dynamic	C_v		$1,8 \cdot 10^{-4}$	$1,8 \cdot 10^{-4}$	$1,8 \cdot 10^{-4}$	$1,8 \cdot 10^{-4}$	mNm/min^{-1}
9	Speed constant	k_n		696	481	323	266	min^{-1}/V
10	Back-EMF constant	k_E		1,44	2,08	3,1	3,75	mV/min^{-1}
11	Torque constant	k_M		13,7	19,9	29,6	35,8	mNm/A
12	Current constant	k_I		0,073	0,05	0,034	0,028	A/mNm
13	Slope of n-M curve	$\Delta n/\Delta M$		28	27	28	31	min^{-1}/mNm
14	Terminal inductance, phase-phase	L		90	190	410	640	µH
15	Mechanical time constant	τ_m		10,4	9,7	10,4	11,1	ms
16	Rotor inertia	J		34,9	34,9	34,9	34,9	gcm^2
17	Angular acceleration	$\alpha_{max.}$		84	124	117	120	$\cdot 10^3 rad/s^2$
18	Thermal resistance	R_{th1} / R_{th2}	1,6 / 6,2					K/W
19	Thermal time constant	τ_{w1} / τ_{w2}	15,4 / 820					s
20	Operating temperature range:							
	– motor		-30 ... +125					°C
	– winding, max. permissible		+125					°C
21	Shaft bearings		ball bearings, preloaded					

Element	Source
SKF W 638/4 XR-2Z deep grove ball bearing	http://www.skf.com/group/products/bearings-units-housings/ball-bearings/deep-groove-ball-bearings/deep-groove-ball-bearings/index.html?designation=W%20638/4%20XR-2Z



d		4	mm
D		10	mm
B		4	mm
d_1	≈	5.9	mm
D_2	≈	8.8	mm
D_3		11.6	mm
C		0.8	mm
$r_{1,2}$	min.	0.15	mm

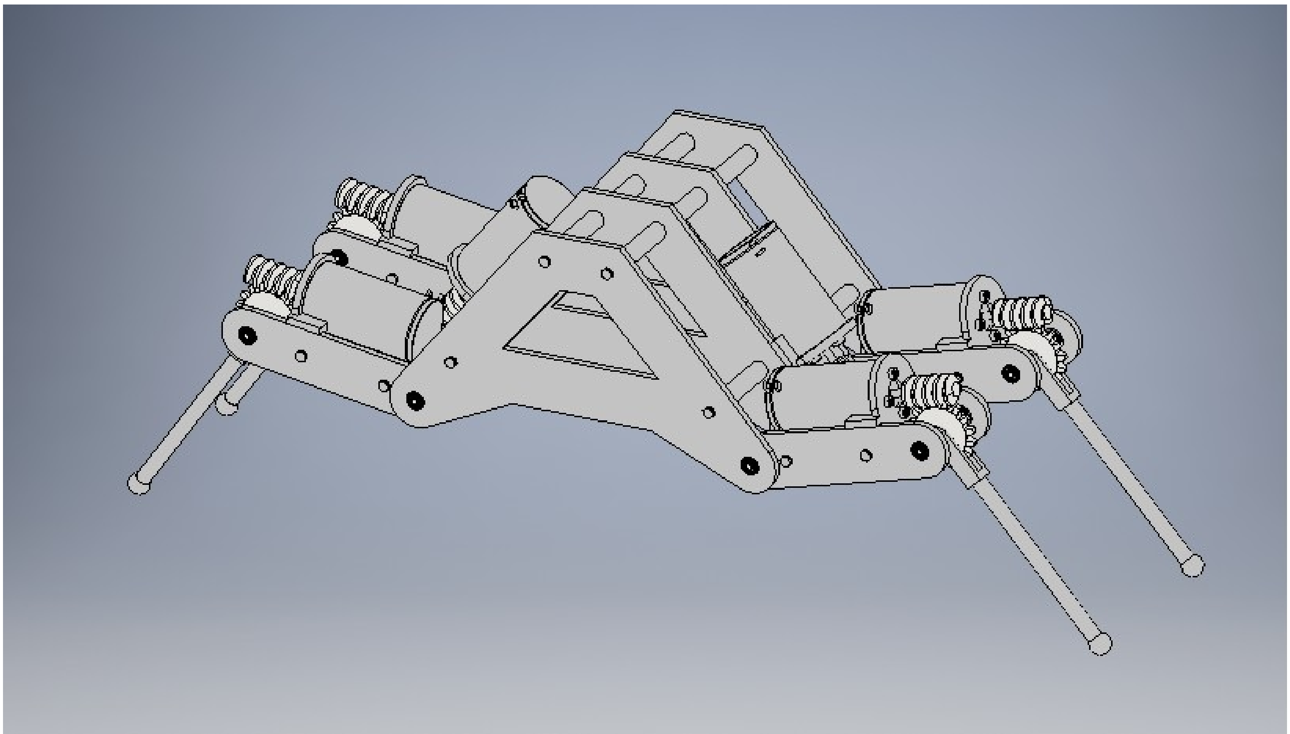
CAD Model

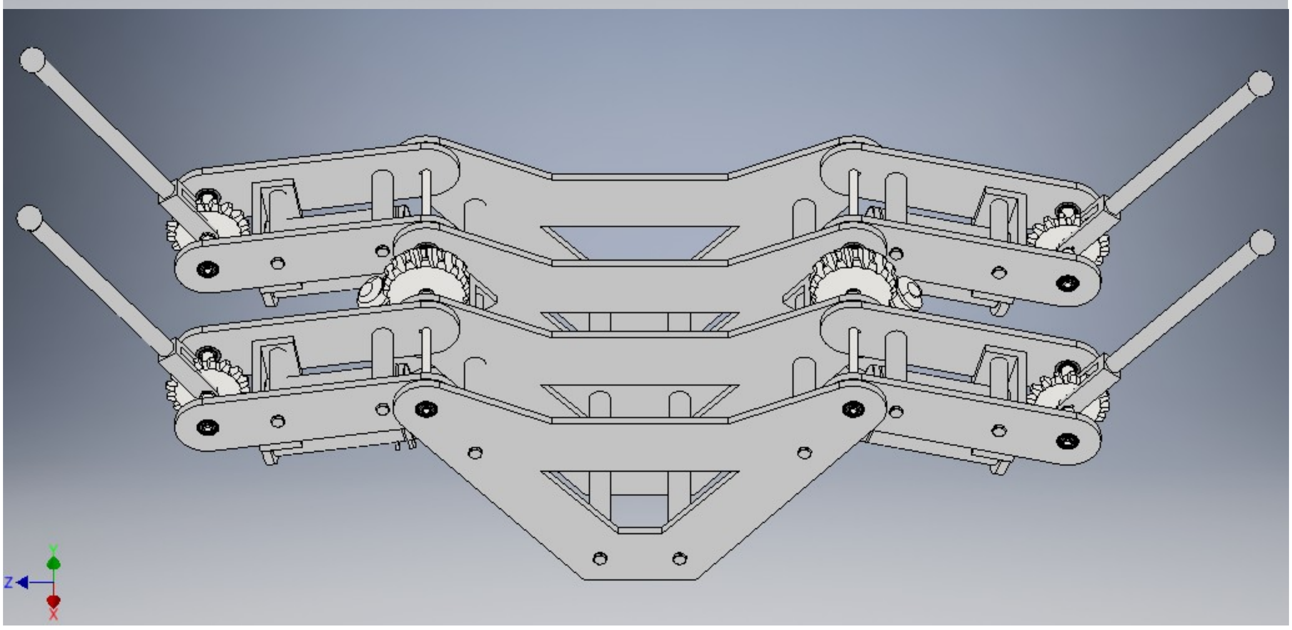
After putting all the previous elements together, we acquire such a model.

The plates were designed to be simple to manufacture and easy to be put together. Bushings and long M4 bolts tighten the aluminum plates and hold them in place.

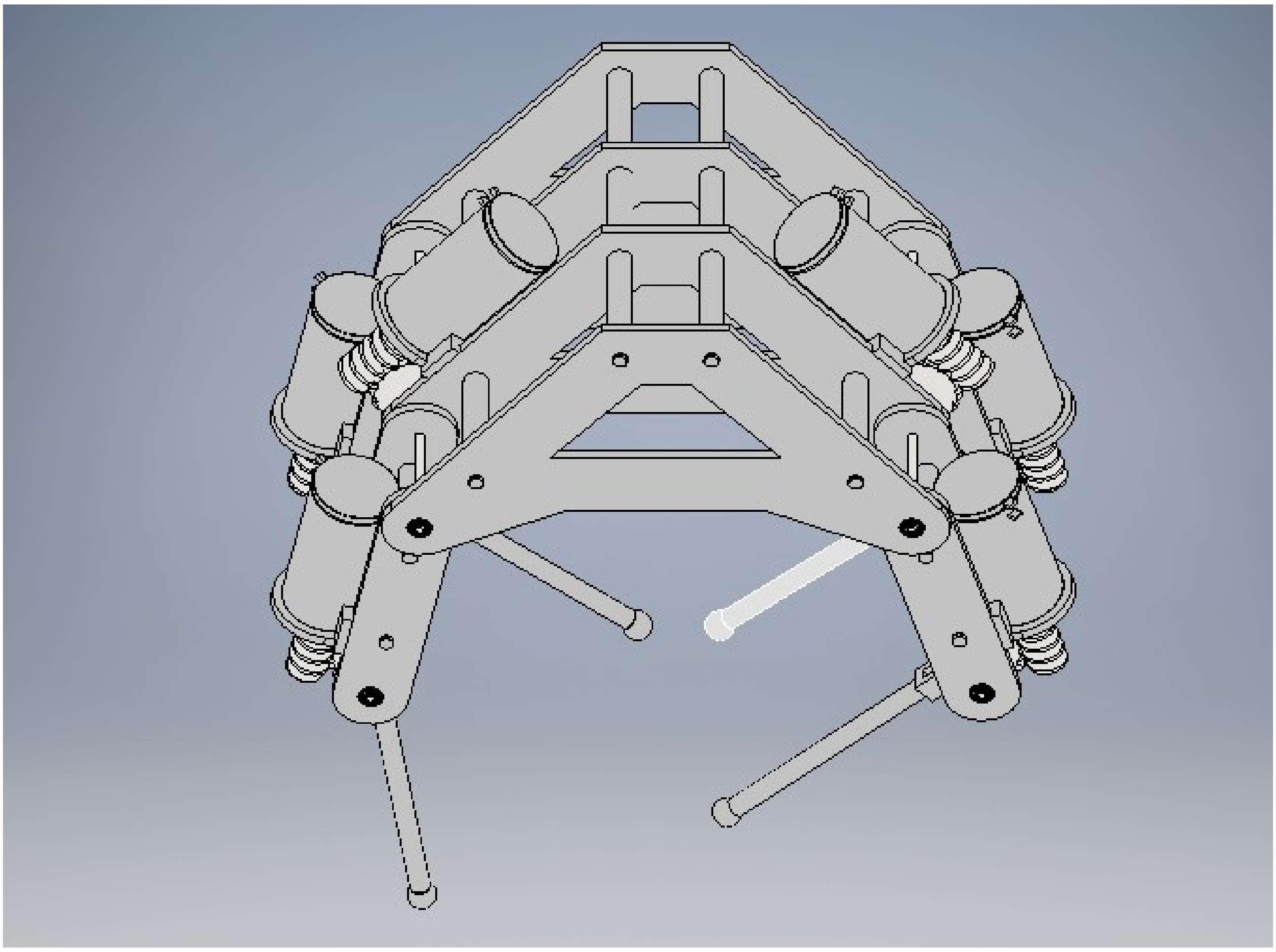
The entire design weighs approximately 3kg; including motors, shafts and gears. The design is also easily scalable thanks to its simple design.

Open Grip:





The four finger tips are meant to move independently to provide a versatile grip, while the two base joint pairs provide the holding torque.

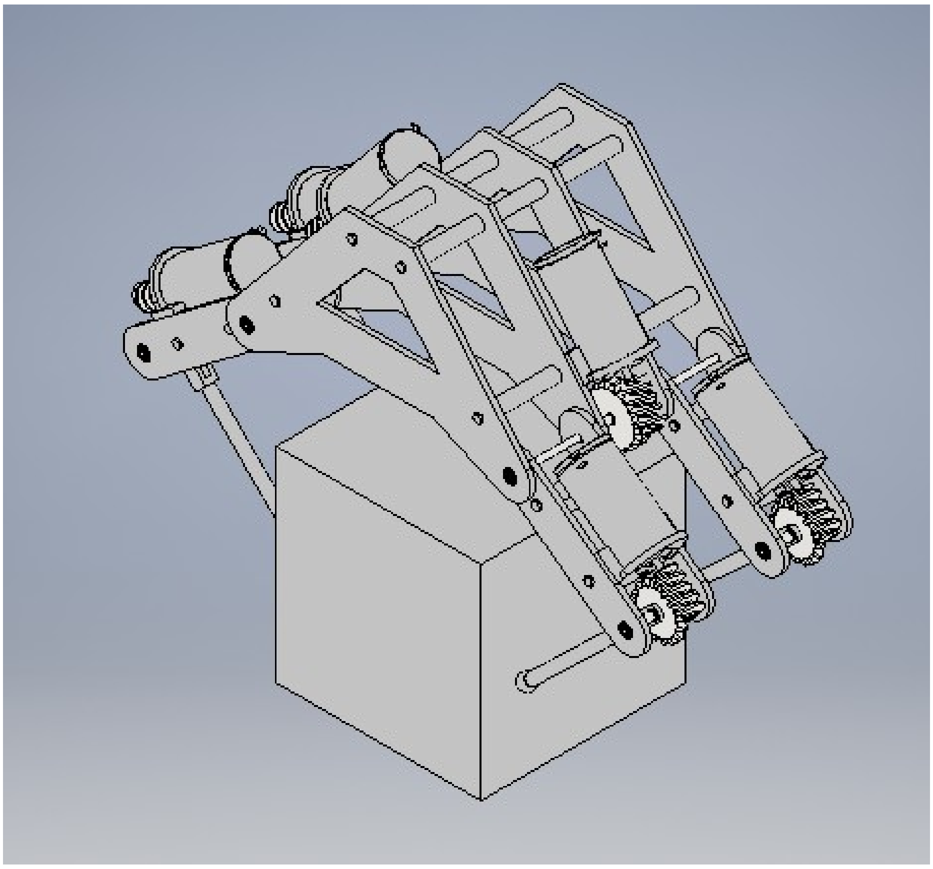


With the prototype model of the hand created, we may take the exact parameter specifications of our system (lengths, weight, center of masses) and use them to create updated. This will provide us with enough knowledge about our design to figure out its limitations.

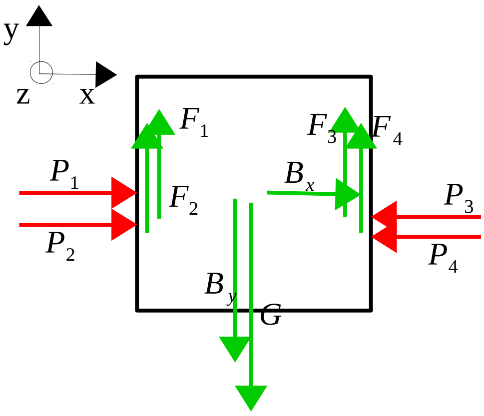
The simulations will also give us crucial information that could help us create a better design with the next iteration.

Next we inspect a static situation when the hand has already gripped a desired object.

The first task of the hand is to manipulate cubes of a weight of around 0.5kg.



It is possible that the gripping points will not align perfectly during gripping. The following conditions must be satisfied. The forces P and F are the same forces as before in the force analysis.



Forces:

$$\begin{cases} F_n = P_n \cdot 0.2 \\ F_1 = F_2 = F_3 = F_4 \end{cases}, \quad \begin{cases} F_1 + F_2 + F_3 + F_4 = 4 \cdot F_1 = G + B_y \\ P_1 + P_2 - P_3 - P_4 = B_x \end{cases}$$

Moments:

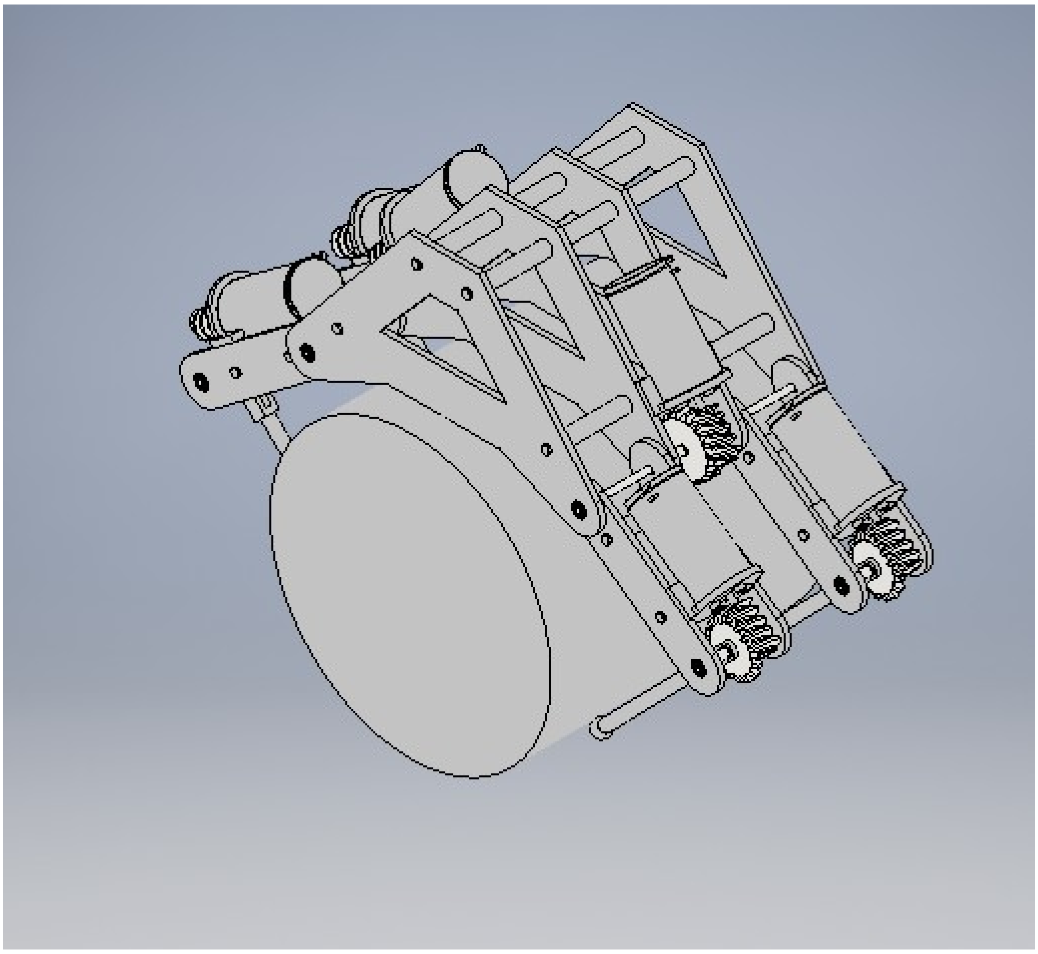
$$\begin{cases} P_1 \cdot x_1 + P_2 \cdot x_2 - P_3 \cdot x_3 - P_4 \cdot x_4 = 0 \\ F_1 \cdot z_1 + F_2 \cdot z_2 - F_3 \cdot z_3 - F_4 \cdot z_4 = 0 \end{cases}$$ x,z – center point distances

As long as the above conditions are satisfied, the cube will remain in a stable position. The formulas for moments must hold true for all three planes, therefore there should be three sets of equations for each plane. For static cases, the inertial force may be omitted and set to 0.

There are two ways of enforcing the stated conditions. The first is to implement pressure sensors and create a closed loop system with a compensator for the potential offsets. This adds the potential of making the hand robust with a proper control, however is much more difficult and expensive to implement.

The second is ensuring the accuracy and precision of the grip by careful calibration. If the hand is consistent and mechanically stable, such calibration should be possible on the condition the manipulated objects are found in the same position after every cycle. This will require an external source properly preparing the environment for our device.

Below shows that our design may be multi-functional if configured correctly.



The calculations for the cylinder may be carried out in a similar manner as the cube. However, unless the cylinder is large enough to be pressed into the palm, which will lock it into place and grip offsets will be more forgiving, the calculations may be more complex. A variation in the position of the grip changes the angle of the pressing force on the circular surface and the pressing force has to be more precise. This makes the need for precision in the calibration phase even more important and complex.

As the complexity of the handled objects increases, an open loop system becomes less practical as calibration expenses go up. At a certain point a feed back system makes more sense.

Before moving on to the next calculations, we took advantage of some Inventor tools to find values for the locations of center of masses and masses of individual joints. These include the shafts, bearings, gears and motors.

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

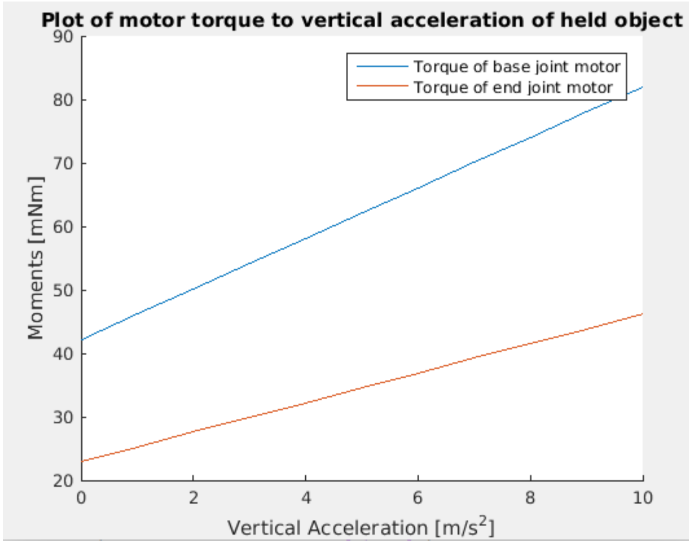
The ratio between the worm and worm gear will have a significant impact on the required torque of the motor.

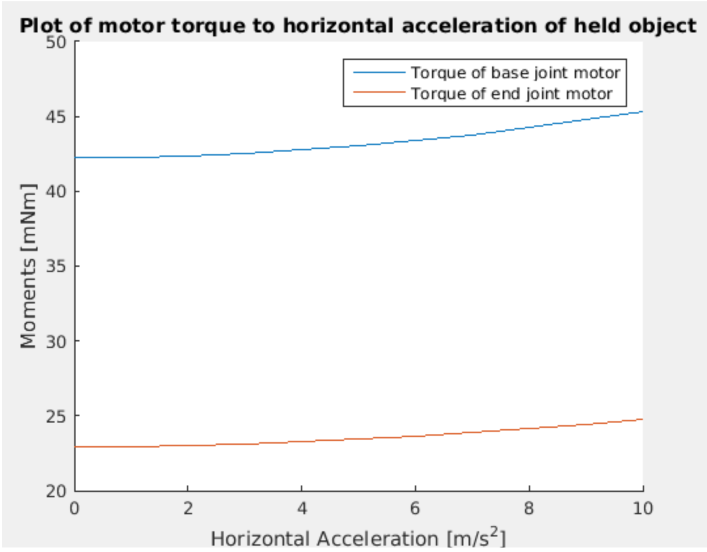
```
function Mw = GearRatio(M)
    pitch = 0.0025;
    d_gear = 0.015;
    Mw = (M * pitch)/(2*3.14*d_gear);
end
```

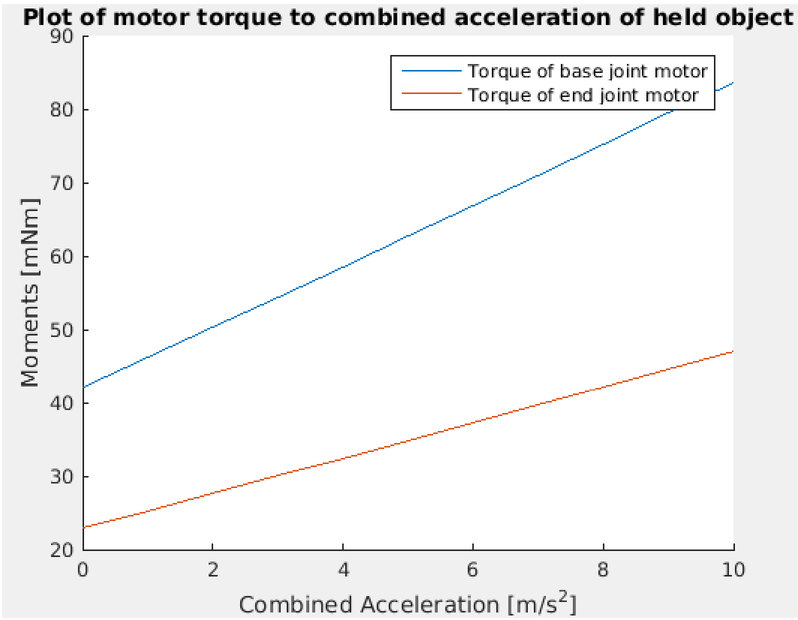
The following script provides us with the force analysis when the hand is under the greatest strain(while in motion). We consider three case; vertical, horizontal and combined accelerations.

```
% 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.

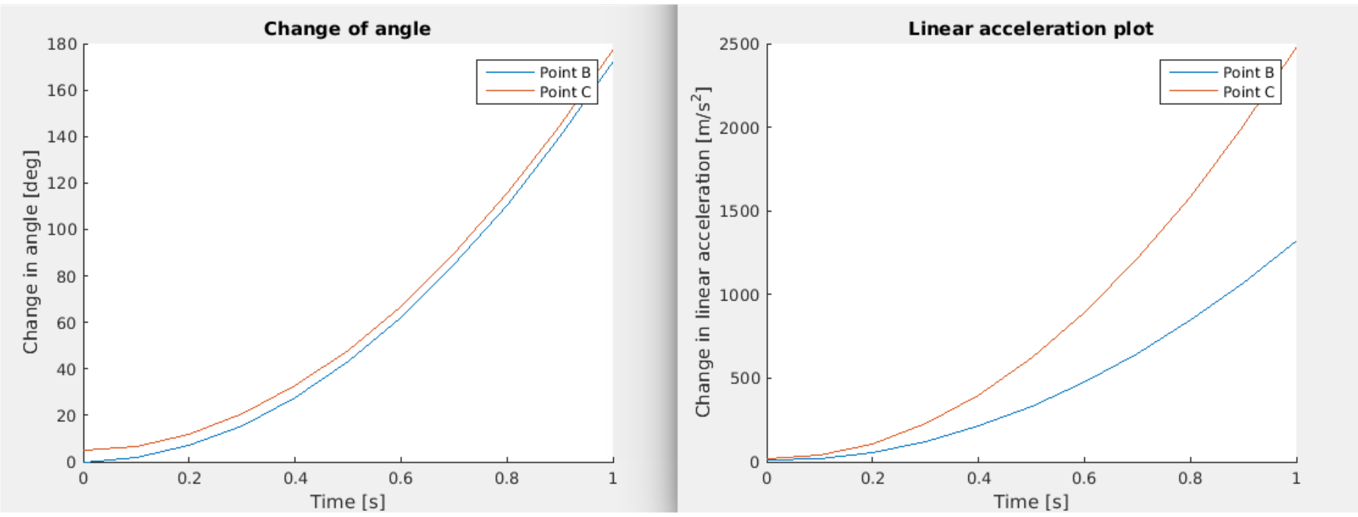
$\omega_{out} = \omega_{motor} / n$, where n is the number of teeth on the gear

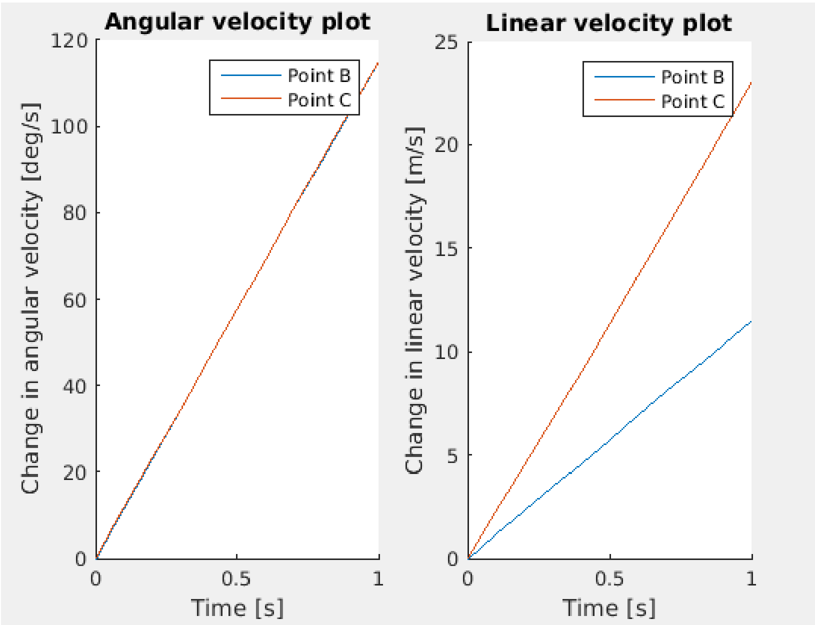
This gives us the following:

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

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

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.

Conclusions

The focal point of the project was the mechanical analysis, including both the dynamics (kinetic and kinematic) and static analysis.

The kinetic analysis is essential as the current state (position, velocity, acceleration) is very important to know to later carry out a full kinematic analysis. The kinematic analysis allows us to derive the appropriate formulas which can eventually be converted into state-space form. With a mathematical description of the plant, controller design is possible as well as simulations of motion and forces.

The next part consists of the static analysis of the hand while it is gripping an object. This analysis gives us the information needed to acquire the necessary parts and decide on the minimum torque required to hold the object still. Depending on the object being held, the forces may act at different angles and an analysis of the object should also be carried out to ensure it is in equilibrium ie. two fingers aren’t creating a couple of moments causing rotation.

Our equations were derived by hand, while all simulations were carried out in a multi-functional Matlab script.

The next part of our report consisted of the design of the machine prototype based on rough approximations created in the Matlab script. The 3D model demonstrates several of its layouts. The controller and electronics were omitted, however the design may easily incorporate cable management and external mounts to the bushings.

The final simulations show that the motors are capable of lifting much larger objects and/or may move with larger accelerations. For our case, it is well within its operating range and even smaller motors may be considered to make the hand lighter.

All files and documentation may be found on github:
https://github.com/drdurczok/MechanicalHand