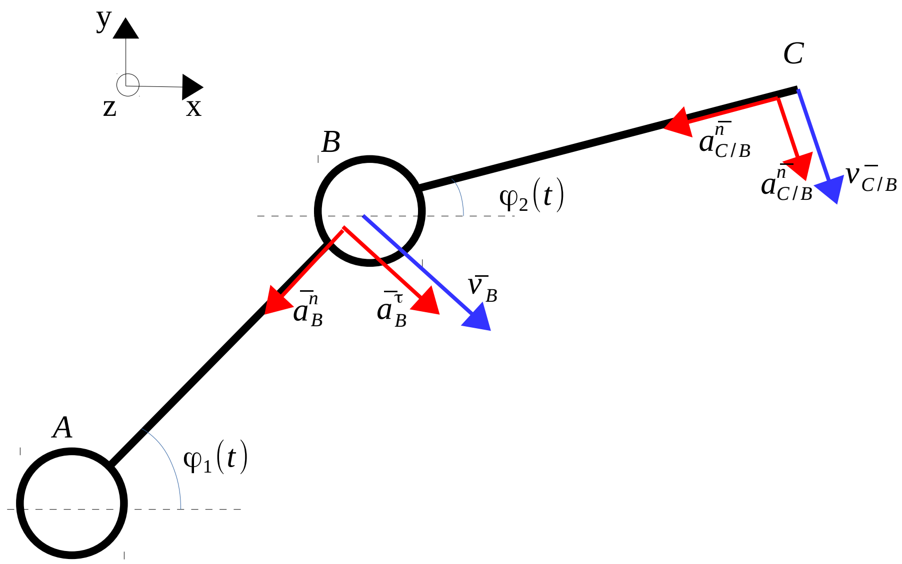
Mechanical Analysis

Velocity and Acceleration of Mechanical Finger



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

Velocity:

$$\begin{cases} V^x_C = V^x_B + V^x_{C/B} \\ V^y_C = V^y_B + V^y_{C/B} \end{cases}$$

$$\begin{cases} V^x_C = V_B \sin\varphi_1(t) + V_{C/B} \sin\varphi_2(t) \\ V^y_C = V_B \cos\varphi_1(t) + V_{C/B} \cos\varphi_2(t) \end{cases}$$

$$V_c = \sqrt{(V^x_c)^2 + (V^y_c)^2} =$$

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

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

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):

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

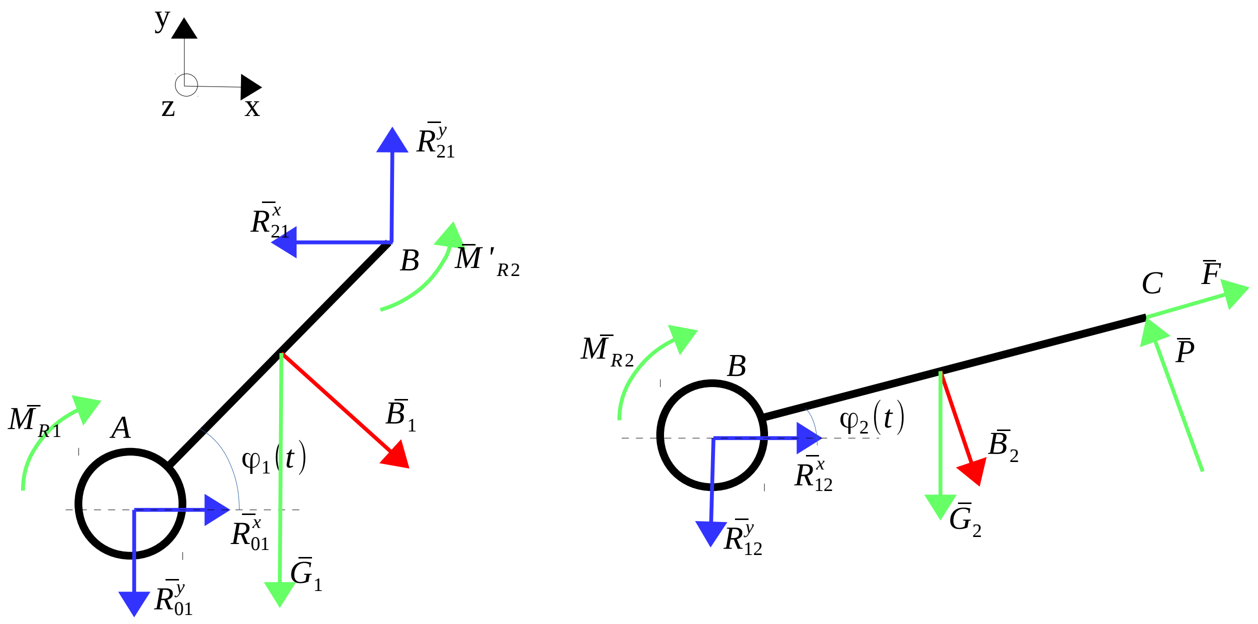
$$a_B = l_{AB}\sqrt{\varepsilon_1^2 + \omega_1^4}$$

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:

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

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

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

For the second element:

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

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

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

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

The equations need to be prepared for matrix form:

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

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.

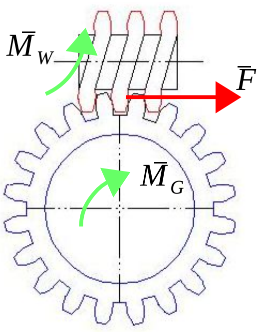
Assumptions

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:



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

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

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

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.