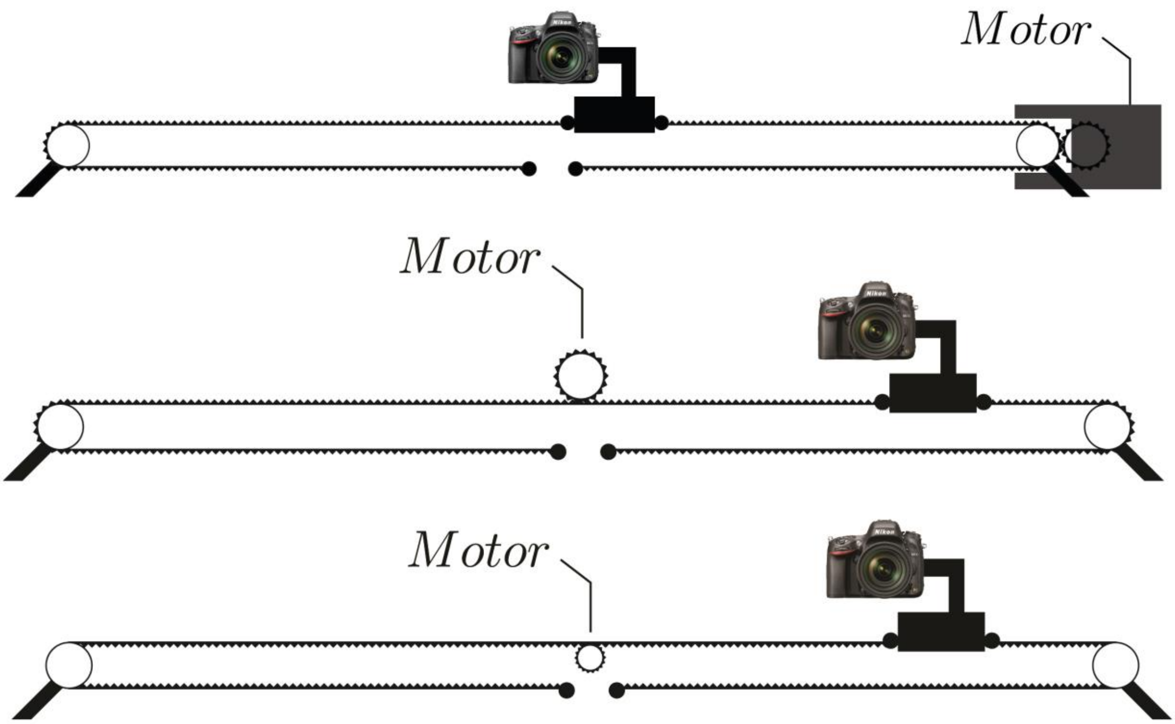


Figure 1

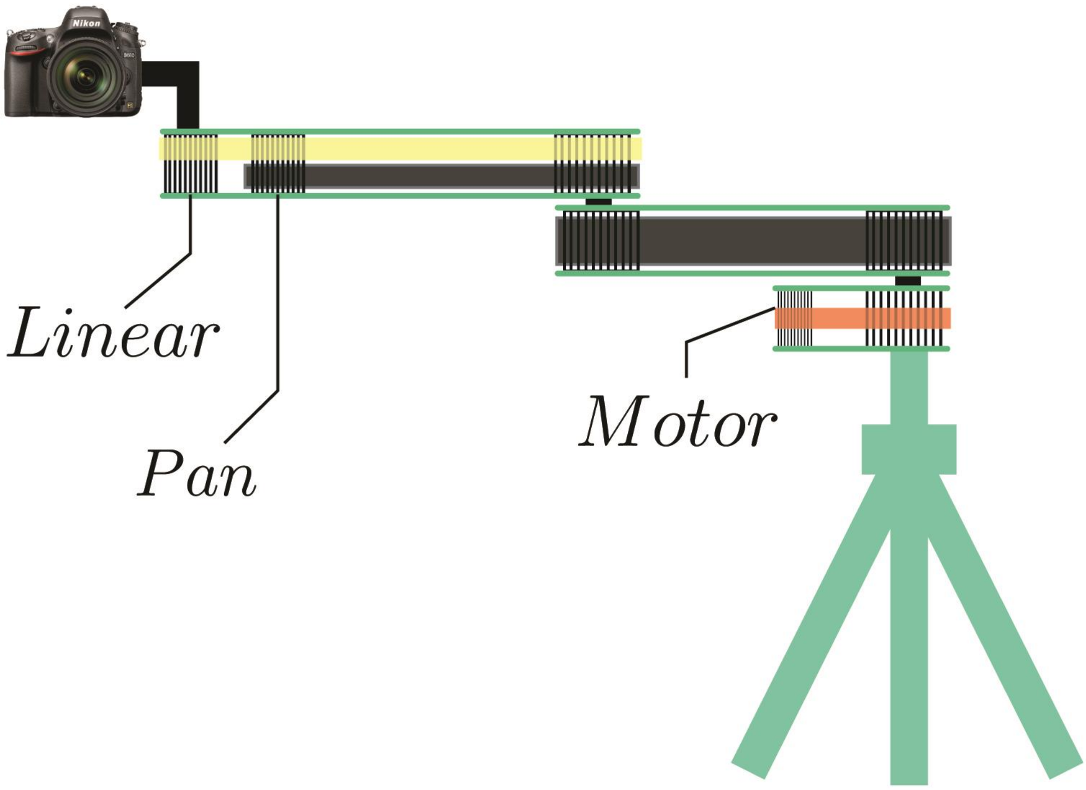


Figure 2

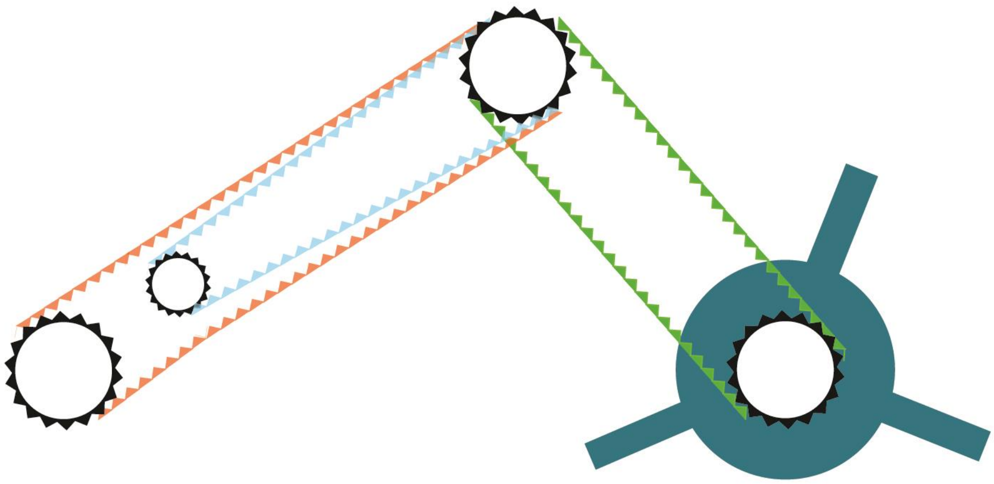


Figure 3

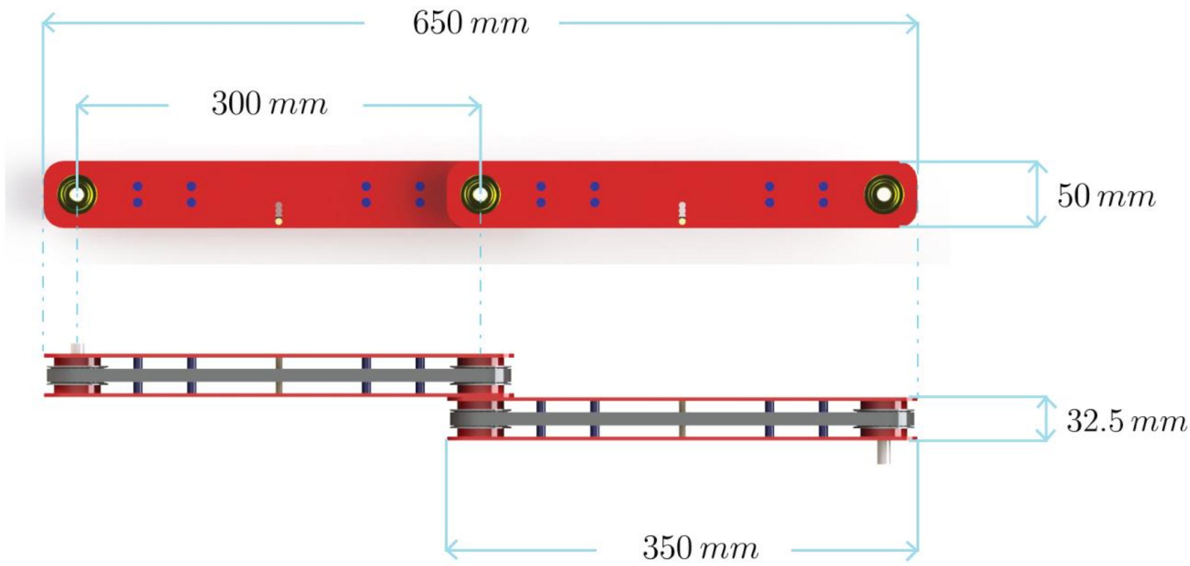


Figure 4

1. Analysis

Kinematic Analysis

Kinematic analysis applied on the mechanism that is represented in figure. m_1 and m_2 are the masses of links respectively. Links have lengths of L_1 and L_2. End points of link 1 is defined as (x_1, y_1, z_1), and (x_2, y_2, z_2) is the end point of link 2. The aim of this kinematic analysis is to obtain τ_{θ_1} and τ_{θ_2} for given inputs of m, L, $\dot{\theta}$ and $\ddot{\theta}$. f_1 and f_2 are forces needed to obtain the motion. Newton's second law of motion states that;

$$f = ma$$

In this mechanism, forces f_1 and f_2 can be defined as;

$$f_1 = \begin{bmatrix} f_{x_1} \\ f_{y_1} \\ f_{z_1} \end{bmatrix} = m_1 \begin{bmatrix} \ddot{x}_1 \\ \ddot{y}_1 \\ \ddot{z}_1 \end{bmatrix}$$

and

$$f_2 = \begin{bmatrix} f_{x_2} \\ f_{y_2} \\ f_{z_2} \end{bmatrix} = m_2 \begin{bmatrix} \ddot{x}_2 \\ \ddot{y}_2 \\ \ddot{z}_2 \end{bmatrix}$$

Points presented in figure can be written as a function of L_1 , L_2, $\theta_1$and θ_2;

$$for\ link\ 1 \rightarrow \begin{cases} x_1 = L_1 \text{x} \cos\theta_1 \\ y_1 = L_1 \text{x} \sin\theta_1 \\ z_1 = 0 \end{cases}$$

$$for\ link\ 2 \rightarrow \begin{cases} x_2 = L_1 \text{x} \cos\theta_1 + L_2 \text{x} \cos\theta_2 \\ y_2 = 0 \\ z_2 = 0 \end{cases}$$

Time derivative of the positions will be

$$\dot{x}_1 = -L_1 \text{x } sin\theta_1 x\ \dot{\theta}_1$$
$$\dot{y}_1 = L_1 \text{x } cos\theta_1 x\dot{\theta}_1$$
$$\dot{z}_1 = 0$$

and

$$\dot{x}_2 = -L_1 \text{x } sin\theta_1 x\ \dot{\theta}_1 - L_2 \text{x } sin\theta_2 x\ \dot{\theta}_2$$
$$\dot{y}_2 = 0$$
$$\dot{z}_2 = 0$$

And then the directional accelerations of the given system is;

$$\ddot{x}_1 = -L_1 \text{x } cos\theta_1 x\ \dot{\theta}^2{}_1 - L_1 \text{x } sin\theta_1 x\ \ddot{\theta}_1$$
$$\ddot{y}_1 = L_1 \text{x } sin\theta_1 x\dot{\theta}^2{}_1 - L_1 \text{x } cos\theta_1 x\ \ddot{\theta}_1$$
$$\ddot{z}_1 = 0$$

and

$$\ddot{x}_2 = -L_1 \text{x } sin\theta_1 x\ \ddot{\theta}_1 - L_1 \text{x } cos\theta_1 x\ \dot{\theta}^2{}_1 - L_2 \text{x } sin\theta_2 x\ \ddot{\theta}_2 - L_2 \text{x } cos\theta_2 x\ \dot{\theta}^2{}_2$$
$$\ddot{y}_2 = 0$$
$$\ddot{z}_2 = 0$$

These accelerations are substituted into Newton's second law of motion

$$f_1 = \begin{bmatrix} f_{x_1} \\ f_{y_1} \\ f_{z_1} \end{bmatrix} = m_1 \begin{bmatrix} -L_1 \text{x } cos\theta_1 x\ \dot{\theta}^2{}_1 - L_1 \text{x } sin\theta_1 x\ \ddot{\theta}_1 \\ L_1 \text{x } sin\theta_1 x\dot{\theta}^2{}_1 - L_1 \text{x } cos\theta_1 x\ \ddot{\theta}_1 \\ 0 \end{bmatrix}$$

$$f_2 = \begin{bmatrix} f_{x_2} \\ f_{y_2} \\ f_{z_2} \end{bmatrix} = m_2 \begin{bmatrix} -L_1 \text{x } sin\theta_1 x\ \ddot{\theta}_1 - L_1 \text{x } cos\theta_1 x\ \dot{\theta}^2{}_1 - L_2 \text{x } sin\theta_2 x\ \ddot{\theta}_2 - L_2 \text{x } cos\theta_2 x\ \dot{\theta}^2{}_2 \\ 0 \\ 0 \end{bmatrix}$$

The torques needed to obtain the motion can be found by torque matrix that is obtained by using free-body-diagrams

$$\tau_{\theta_i} = \begin{bmatrix} \tau_{\theta_1} \\ \tau_{\theta_2} \end{bmatrix} = \begin{bmatrix} \tau_{\theta_2} - f_1 x L_1 x sin\theta_1 + f_1 x L_1 x cos\theta_1 \\ -f_2 x L_2 x sin\theta_2 - f_2 x L_2 x cos\theta_2 \end{bmatrix}$$

And, so that velocity squared is

$$v^2{}_1 = \dot{x}^2{}_1 + \dot{y}^2{}_1$$
$$v^2{}_2 = \dot{x}^2{}_2 + \dot{y}^2{}_2$$

so directional velocity of each link is

$$v_1 = \sqrt{(-L_1 \text{x } sin\theta_1 x\ \dot{\theta}_1)^2 + (L_1 \text{x } cos\theta_1 x\dot{\theta}_1)^2}$$
$$v_2 = \sqrt{(-L_1 \text{x } sin\theta_1 x\ \dot{\theta}_1 - L_2 \text{x } sin\theta_2 x\ \dot{\theta}_2)^2}$$

And angular velocity can be found by using directional velocity and length of links;

$$v_1 = \omega_1 x L_1$$
$$v_2 = \omega_2 x L_1$$

By using these one can obtain

$$\omega_1 = \frac{v_1}{L_1}$$

and

$$\omega_2 = \frac{v_2}{L_2}$$

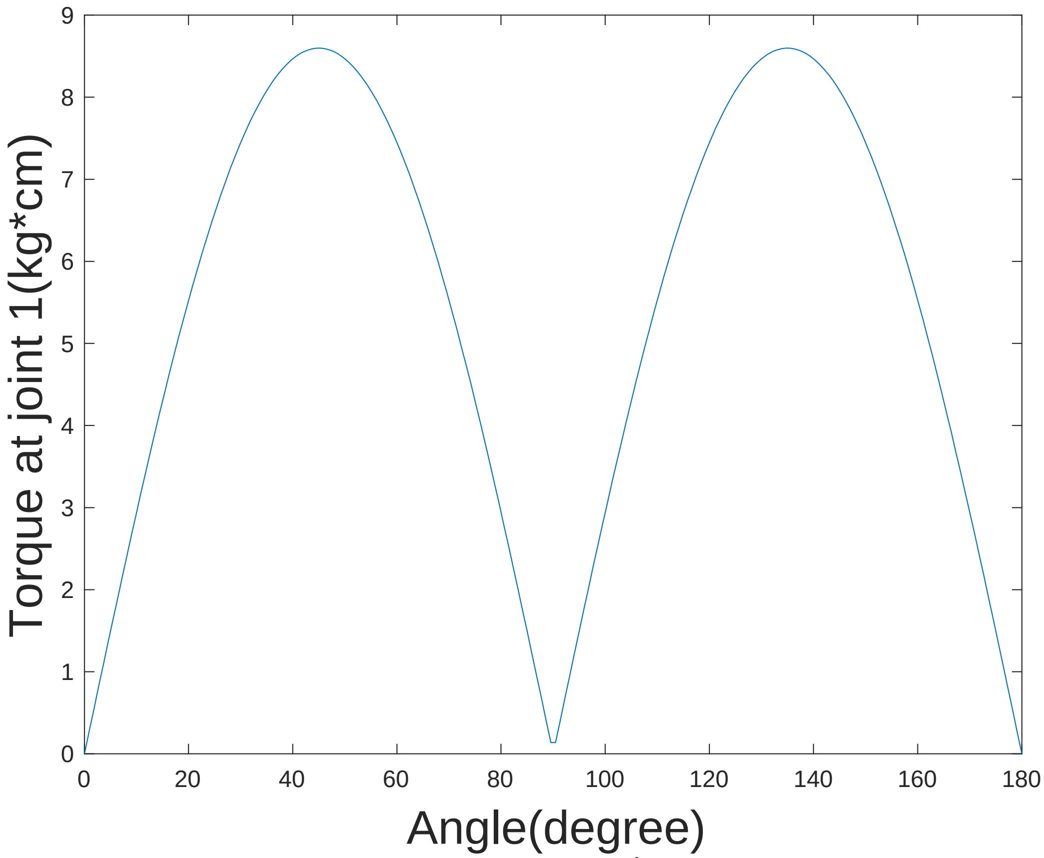
Power needed to run this system will be;

$$P_1 = \tau_{\theta_1} x \omega_1$$

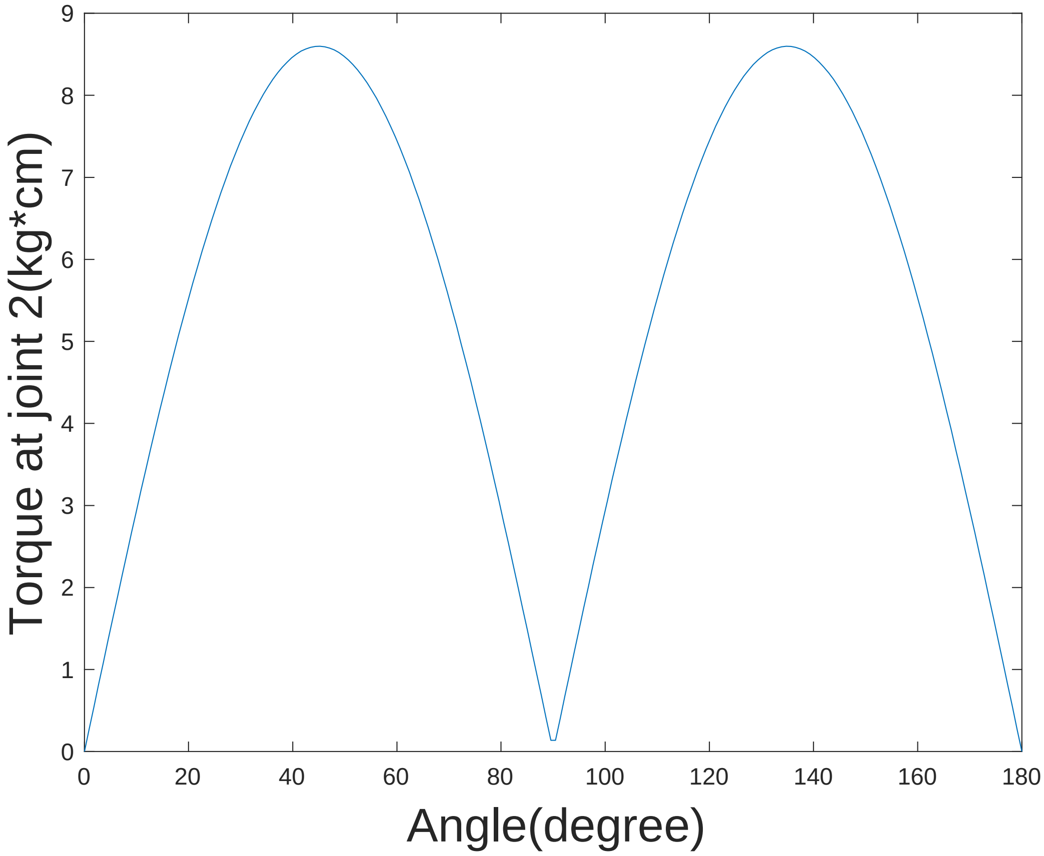
$$P_2 = \tau_{\theta_2} x \omega_2$$

The numerical values of torque at each joint can be obtained with this analysis for given inputs of mass, length of links, velocity of the system, and acceleration of the system. CAD simulation will provide information of mass of each link, then by defining minimum time required to complete the linear movement, maximum torque needed will be computed.

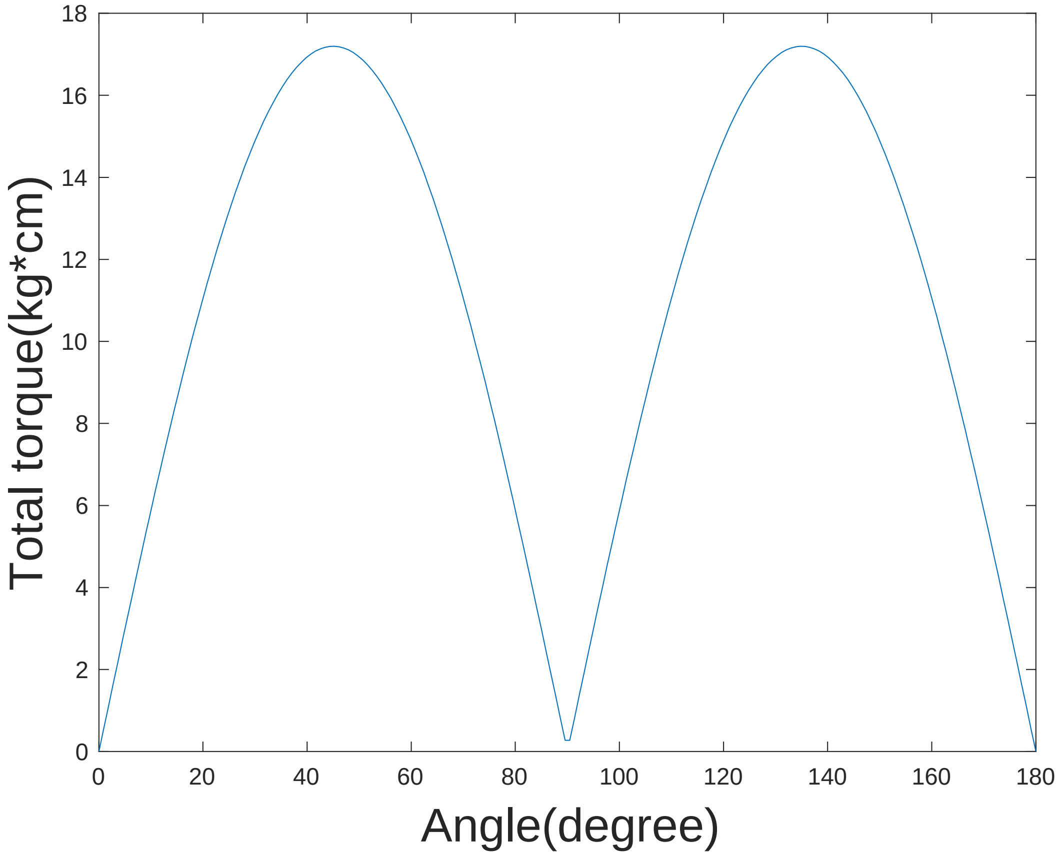
The mass of the link 1 is selected as 0.8 kg, and mass of link 2 with camera is selected as 3.8 kg. Then torque/position of θ_1 graph is given in the **FİGURE** with units of kg*cm.

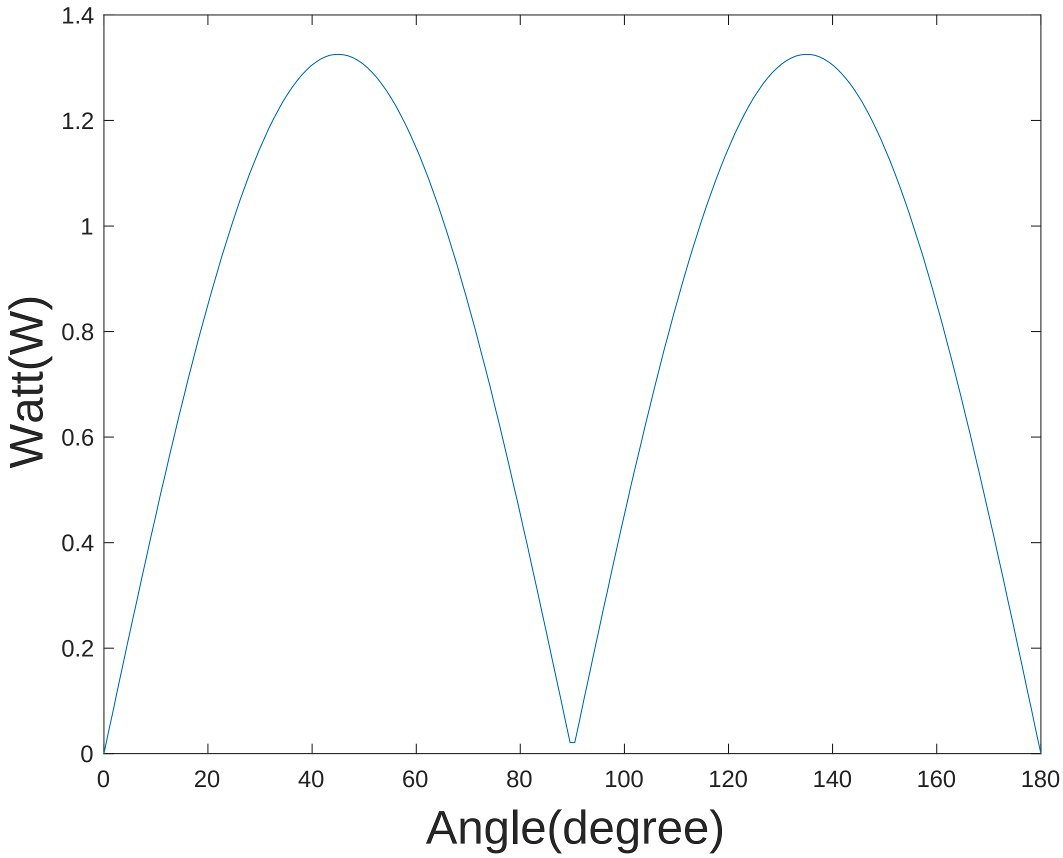


And then torque/position of θ_2 graph is shown in the **FİGURE**



Since the all torque will be provided by one motor, we can select required torque as given in the **FİGURE.** Total torque needed to run the system is $\tau = \tau_{\theta_1} + \tau_{\theta_2}$


Total torque(kg*cm)
18
16
14
12
10
8
6
4
2
0
0 20 40 60 80 100 120 140 160 180
Angle(degree)


Watt(W)
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0 20 40 60 80 100 120 140 160 180
Angle(degree)

Belt-Pulley System

Maximum tension T_{max} at the belt can be calculated with the formula of

$$T_{max} = \frac{2.5\ \tau_{max}}{Pd}$$

where Pd is the pitch diameter.

$$\tau_{max} = 1.6876Nm$$
$$\tau(lbfin) = 8.85x\ \tau(Nm)$$
$$T_{max} = \frac{37.33815}{Pd}$$
$$T_{peak} = T_{max}xService\ factor$$
$$\tau_{design} = \tau_{max}xService\ factor$$

Service factor is chosen as 1.5 as recommended value

SPD/SI A 6A53-021DF0608 was selected as pulley which has Pd value of 0.79 in. Which is 0.020066m.Then peak tension force is

$$T_{peak} = 79.88\ lbf$$

And design torque is

$$\tau_{design} = 2.53Nm$$

3 mm GT2 seems a reliable at small number of revolution per minute according to the **FİGURE appendix**

To choose a belt design power will be used

$$Pd = PtxService\ factor$$

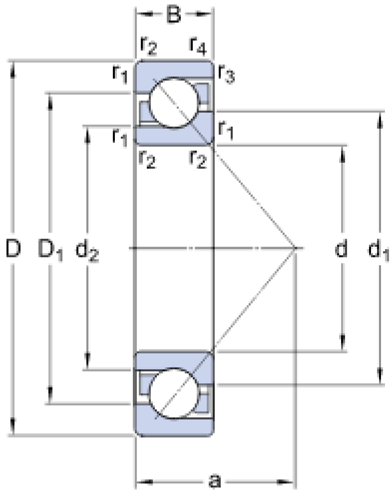
so the design power will be

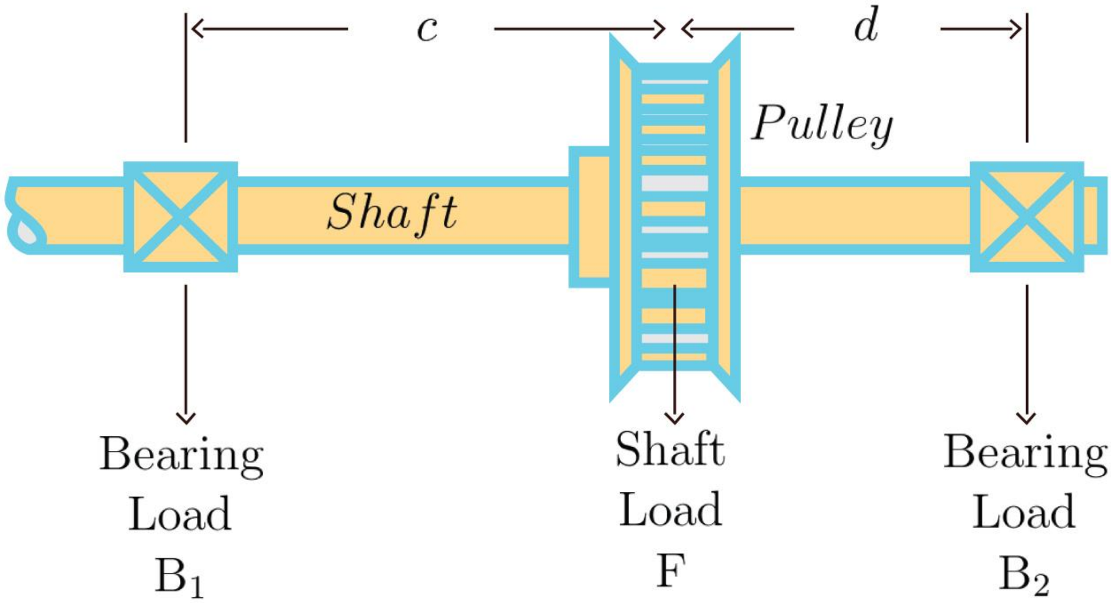
$$Pd = 0.00198375kW$$

Belt will be thickness of 6 mm and material of Nylon Covered, Fiberglass Reinforced, Neoprene. This belt's breaking strength is 158 N per 1 mm belt width. For 6 mm thickness, breaking strength will be $\sim 950N$ and that is higher than $T_{peak} = 79.88\ lbf \sim 350N$

Bearing selection

Bearings are the machine parts that can be used for reducing friction between moving parts, thus in this design 8 SKF 7200 BEP was used to get rid of energy losses. SKF 7200 has the dimensions of d 10 mm D 30 mm B 9 mm d 1 18.3 mm d 2 14.57 mm D 1 22.85 mm a 13 mm r 1,2 min. 0.6 mm r 3,4 min. 0.3 mm which are represented in **FİGURE**





$$B_1 = \frac{Fd}{(c+d)}$$
$$B_2 = \frac{Fc}{(c+d)}$$
$$c = d$$
$$B_1 = B_2 = \frac{F}{2}$$

Loads on the bearings are calculated by the peak tension on the pulley and relative positions of bearings with respect to pulley. In the design, components are aligned on a shaft as two bearings and one pulley between them with equal distance. Therefore, loads are distributed between bearings equally as shown in the equations above. According to calculated loads bearings are selected for the mechanism that withstand the required forces.

Motor selection

Required torque is $2.53Nm{\sim}25kgcm$ which is given as τ_{design} in previous parts. JX Servo PDI-HV5932MG was selected because it can provide up to 32.3 kgcm with 8.4 V. 6V operating speed can provide up to 25.2 kgcm stall torque.

Battery selection

The wing slider is expected to operate for an hour approximately for convenience. The selected motor JX Servo PDI-HV5932MG, is able to generate 2 Nm torque by using **2480** mA at 6.6 V at locked position (stall current).
The BeagleBone Rev C, is using 460 mA at 5 V,
Battery Life = Battery Capacity in Milli amps per hour / Load Current in Mill amps * 0.70
=>Battery Capacity in Milli amps per hour = Battery Life * Load Current in Mill amps / 0.70
1*(2480+460)/0.75
Battery Capacity **(mAh)** = Device Consumption **(mA) * 0.75**
2480+460=3920 mAh

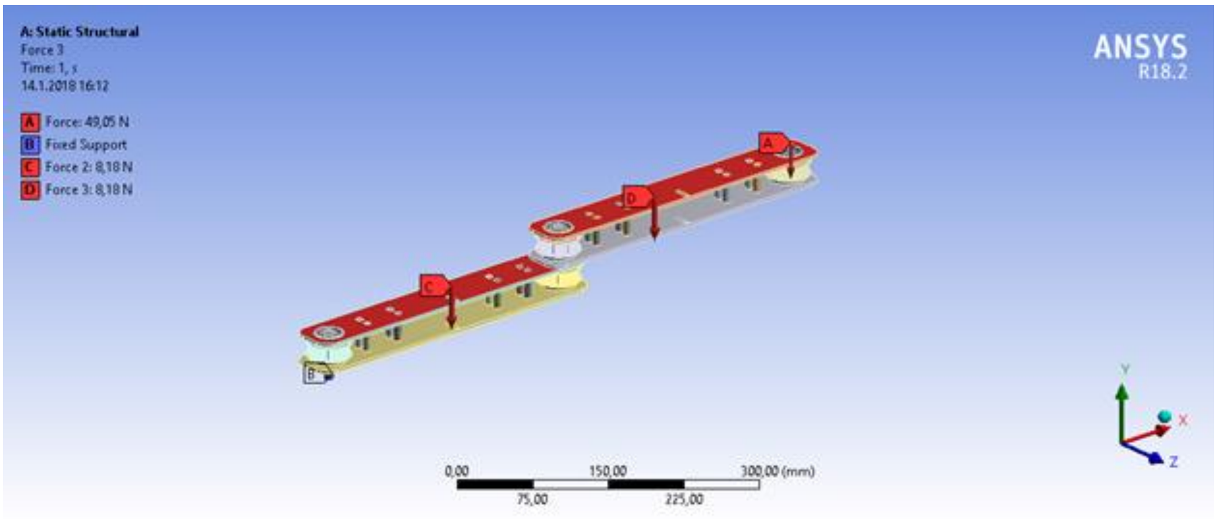
By considering the market products, 4000 mAh Li-Po battery would be sufficient to operate the mechanism for an hour without stopping. As it will not operate continuously under the normal conditions, the slider can work more than an hour with the designed specifications.

**The factor of 0.75 makes allowances for external factors which can affect battery life.*

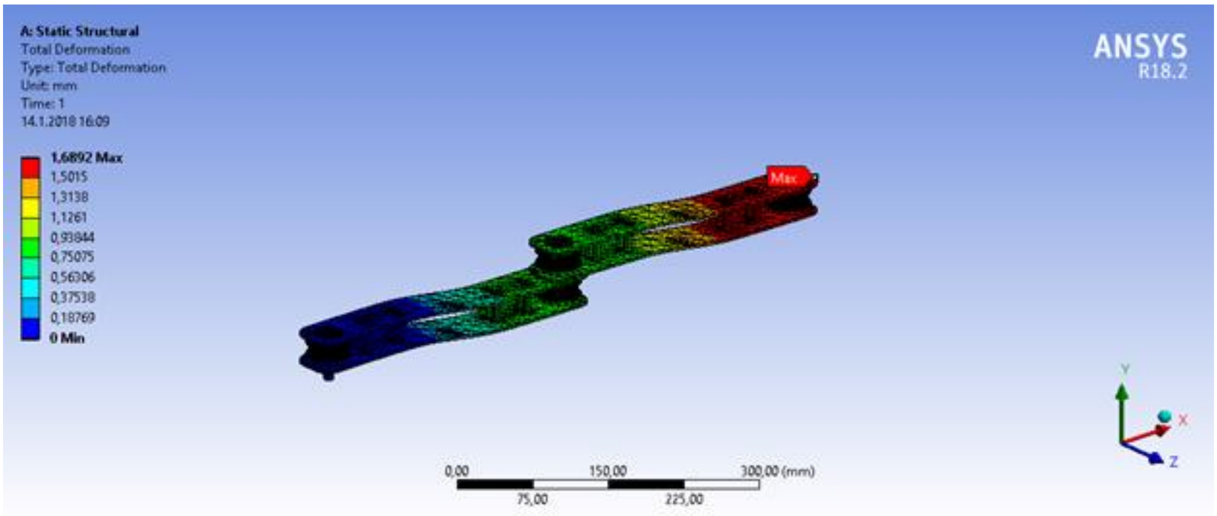
Material selection

Materials used in the wing slider are selected with considering the lightweight construction of the mechanism, required torque and force capacity of the parts and minimizing the deflection of the camera within the motion path. For the links, pulleys are made of aluminum alloy and for the bearings and supporters between links stainless steel is used.

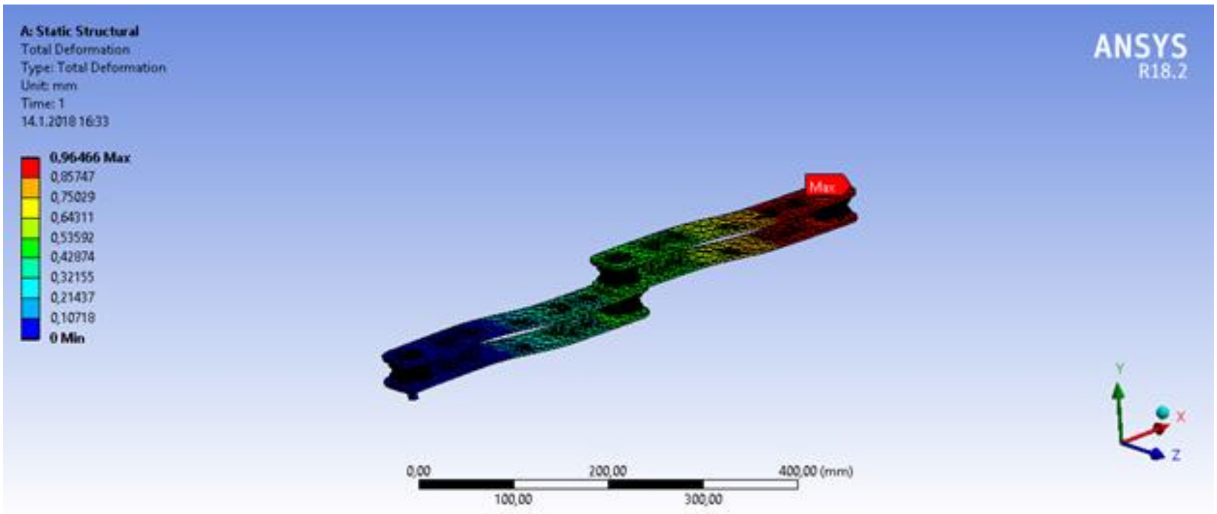
FEM Analysis



Structural analysis of the slider done by using ANSYS. Materials are defined as aluminum for the links and pulleys, stainless steel for bearings and the supporter pins between the links. To obtain maximum deformation and stresses, mechanism is fully extended. By considering weight of the camera as 5 kg, 49.05 N force is applied from the attachment of camera to the mechanism. Weight of the camera is calculated as 1,6681 kg with the defined materials and forces are applied from the center of the each link.



Results reveal that 1,69 mm total displacement occurs at the end of the top link. In order to decrease displacement, link thicknesses were increased from 3 mm to 4 mm. By changing the initial design 0,96 mm total displacement achieved at the end of the fully extended mechanism.



2. CAD Design

CAD design is done with SolidWorks. The dimensions were selected with respected to belt-pulley




3. Control

A mathematical algorithm was constructed on MATLAB to obtain required values of torques at each link. The MATLAB code is presented in the appendix. This MATLAB code will use the data obtained by image processing to do the control.

4. Image processing

Framework Selection

Although there are plenty of others, MATLAB and OpenCV are two most common frameworks/software used for image processing. OpenCV is way too faster than the MATLAB in terms of image processing. MATLAB is built upon Java and Java is built upon C. Therefore, its path through low level language is complicated than the Python, which is OpenCV built upon. At the end, while MATLAB lets you to process 4-5 frames per second, OpenCV can process 30 frames per second, which is more than enough for real time image processing. On top of that, OpenCV is specifically made for image processing, and does have full support from the community and can run even on 64 MB of rams, the only issue is the memory leaks, that can be handled easily.

Board Selection

BeagleBone Black Rev C is a popular embedded computer for internet of things (IoT) applications. It has 1 GHz processor, and 512 MB RAM, which can handle low FPS image processing with any lag behind. Furthermore, it has 4 GB on-board embedded Multi-Media Controller, that is enough for average image processing applications. However, the biggest advantage of BeagleBone is its expansion abilities with 65 digital and 7 analog-to-digital controllers. Therefore, BeagleBone Black Rev C is selected for image processor board.

Development of a Prototype Object Detection Application

To further test the capabilities of OpenCV and also our competency to develop an image processing / machine learning capable IoT project, a small-medium sized code has been written. The packages imutils, numpy, argparse, time and OpenCV are used for the development of the project. As pre-trained model is needed for the software, MobileNet SSD pre-trained model with convolutional neural network, therefore machine learning integrated model has been used thanks to its fast and reliable real-time multiple detection abilities.

Output

The software has been tested on using a single body object per frame, to determine its ability to detect object in real time. The figure below tells the program is able to identify the object as bottle with confidence level of 99.64%, and tells the user to move the camera right to center the object's right and left boundaries into center of the frames.

Figure 5 - OpenCV output of realtime video

As in another example, the software is able to identify the train, and commands user to move left the camera, and when it is centered tell to stay in its current position.

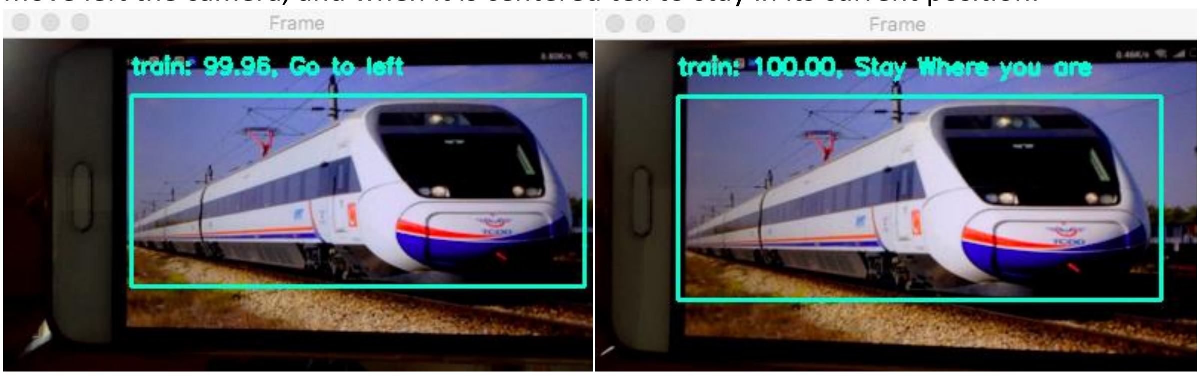


Figure 6 - OpenCV output of object placement

After the first successful attempts, the software has been tested on multiple objects in frames. From the figures below, it can be seen that it is able to identify the objects correctly, however, the confidence level is either low or the object are missing to be identified with small errors. The main reason is the illumination for the specific examples. Therefore, we can conclude that, to get successful and reliable results from OpenCV, we need to operate under the well-illuminated conditions.

Figure 7 - OpenCV analysis on multiple objects

5. Conclusion

To conclude up, a wing slider design for a slider has been chosen by the company. Furthermore, instead of mechanical panning, it has been decided to use image processing unit on top of camera, to put an extra value to the slider by adding not existing product specifications in the market.

6. References

http://www.skf.com/group/products/bearings-units-housings/ball-bearings/angular-contact-ball-bearings/single-row-angular-contact-ball-bearings/single-row/index.html?designation=7200%20BEP

https://www.banggood.com/JX-Servo-PDI-HV5932MG-30KG-Large-Torque-360-High-Voltage-Digital-Servo-p-1074871.html?cur_warehouse=CN

https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-battery-life

http://www.chd.hk/UploadFiles/Att/201701191006295.pdf

https://www.learnopencv.com/embedded-computer-vision-which-device-should-you-choose/

7. Appendix

Figures